Universidad Politécnica de Madrid
Facultad de Informática

INFORMATION RECONCILIATION METHODS IN SECRET KEY DISTRIBUTION

PhD Dissertation

DAVID ELKOUSS CORONAS

Ingeniero de Telecomunicaciones (ETSIT UPM)
Diplôme d'Ingénieur (Télécom ParisTech)

2011

Departamento de Matemática Aplicada
Facultad de Informática

INFORMATION RECONCILIATION METHODS IN SECRET KEY DISTRIBUTION

DAVID ELKOUSS CORONAS

Ingeniero de Telecomunicaciones (ETSIT UPM)

Diplôme d'Ingénieur (Télécom ParisTech)

Supervisors:

VICENTE MARTÍN AYUSO
Ph.D. in Physics

JESÚS GARCÍA LÓPEZ DE LACALLE
Ph.D. in Mathematics

2011

David Elkouss Coronas: *Information Reconciliation Methods in Secret Key Distribution,* PhD Dissertation

SUPERVISORS:
Vicente Martín Ayuso
Jesús García López de Lacalle

LOCATION:
Madrid

Tribunal nombrado por el Magnífico y Excelentísimo Sr. Rector de la Universidad Politécnica de Madrid,

Presidente: .....

Vocal: .....

Vocal: .....

Vocal: .....

Secretario: .....

Realizado el acto de lectura y defensa de la Tesis Doctoral en Madrid, a ..... de ..... de 20..... .

El tribunal acuerca entregar la calificación de .....

El Presidente Los Vocales

El Secretario

「無限の彼方へさあ行くぞ!」

―トイ・ストーリー

ABSTRACT

We consider in this thesis the problem of information reconciliation in the context of secret key distillation between two legitimate parties.

In some scenarios of interest this problem can be advantageously solved with low density parity check (LDPC) codes optimized for the binary symmetric channel. In particular, we demonstrate that our method leads to a significant efficiency improvement, with respect to earlier interactive reconciliation methods. We propose a protocol based on LDPC codes that can be adapted to changes in the communication channel extending the original source. The efficiency of our protocol is only limited by the quality of the code and, while transmitting more information than needed to reconcile Alice's and Bob's sequences, it does not reveal any more information on the original source than an ad-hoc code would have revealed.

Keywords: information theoretic security, secret key distribution, private communications, quantum key distribution, information theory, quantum information theory, coding theory, error correcting code, low density parity check codes, rate adaptation.

RESUMEN

En esta tesis estudiamos el problema de la reconciliación de información en el contexto de la destilación de secreto entre dos partes.

En algunos escenarios de interés, códigos de baja densidad de ecuaciones de paridad (LDPC) adaptados al canal binario simétrico ofrecen una buena solución al problema estudiado. Demostramos que nuestro método mejora significativamente la eficiencia de la reconciliación. Proponemos un protocolo basado en códigos LDPC que puede ser adaptado a cambios en el canal de comunicaciones mediante una extensión de la fuente original. La eficiencia de nuestro protocolo está limitada exclusivamente por el código utilizado y no revela información adicional sobre la fuente original que la que un código con la tasa de información adaptada habría revelado.

Palabras clave: seguridad informacional, distribución de claves secretas, distribución cuántica de claves, teoría de la información, teoría cuántica de la información , teoría de códigos, códigos de baja densidad de ecuaciones de paridad, adaptación de la tasa de información.

PUBLICATIONS

Some of the ideas described hereafter have been published in the following articles:

JOURNAL PAPERS

1. David Elkouss, Jesús Martínez-Mateo, and Vicente Martin. Efficient reconciliation with rate adaptive codes in quantum key distribution. *Quantum Information and Computation*, 11(3&4):0226–0238, 2011.

2. Jesús Martínez-Mateo, David Elkouss, and Vicente Martin. Improved construction of irregular progressive edge-growth tanner graphs. *IEEE Communications Letters*, 14:1155 – 1157, 2010.

3. David Elkouss, Jesús Martínez-Mateo, and Vicente Martin. Untainted Puncturing for Irregular Low-Density Parity-Check Codes. *IEEE Wireless Communications Letters*, 1(6):585–588, 2012.

4. Jesús Martínez-Mateo, David Elkouss, and Vicente Martin. Blind Reconciliation. *Quantum Information and Computation*, 12(9&10), 0791–0812, 2012.

PUBLISHED PROCEEDINGS FROM PEER REVIEWED INTERNATIONAL CONFERENCES

1. Jesús Martínez-Mateo, David Elkouss, and Vicente Martin. Interactive reconciliation with low-density parity-check codes. In *IEEE International Symposium on Turbo Codes & Iterative Information Processing*, pages 280–284, Brest, France, Jul. 2010.

2. D. Lancho, J. Martinez, D. Elkouss, M. Soto, and V. Martin. QKD in Standard Optical Telecommunications Networks. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 36:142–149, 2010.

3. David Elkouss, Jesús Martínez, and Vicente Martín. Secure rate-adaptive reconciliation. In *IEEE International Symposium on Information Theory and Applications*, pages 179–184, Taichung, Taiwan, Oct. 2010.

4. David Elkouss, Jesús Martínez, Daniel Lancho, and Vicente Martín. Rate Compatible Protocol for Information Reconciliation: An application to QKD. In *IEEE Information Theory Workshop*, pages 145–149, Cairo, Egypt, Jan. 2010.

5. David Elkouss, Anthony Leverrier, Romain Alleaume, and Joseph J. Boutros. Efficient reconciliation protocol for discrete-variable quantum key distribution. In *IEEE International Symposium on Information Theory*, pages 1879–1883, Seoul, South Korea, Jul. 2009.

INTELLECTUAL PROPERTY

1. David Elkouss, Jesús Martínez, Daniel Lancho, and Vicente Martín. Método de reconciliación de información para qkd mediante el uso de códigos ldpc adaptando la tasa de información. *Patent pending*, Jan. 2010.

2. David Elkouss, Jesús Martínez, and Vicente Martín. Quantum key distribution - cascade. *Registered software*, Oct. 2009.

PUBLISHED PROCEEDINGS FROM NATIONAL CONFERENCES

1. Jesús Martínez-Mateo, David Elkouss, Alex Ciurana, Daniel Espino, Vicente Martín. Reconciliación de errores mínimamente interactiva en distribución cuántica de claves In *XXXIII Bienal de Física*, in press, Santander, Spain, Sep. 2011.

2. David Elkouss, Jesus Martinez, Daniel Lancho and Vicente Martin. Protocolos de reconciliación de información Óptimos y no interactivos para distribución cuántica de claves In *XXXII Bienal de Física*, pages 550–551, Ciudad Real, Spain, Sep. 2009.

The following articles were also produced during the thesis but are not related to the content described here:

1. Alex Ciurana, Nino Walenta, Jesús Martínez-Mateo, David Elkouss, Mercedes Soto, Vicente Martín. Distribución cuántica de claves en redes de acceso WDM-PON. In *XXXIII Bienal de Física*, in press, Santander, Spain, Sep. 2011.

2. Daniel Lancho, Jesus Martinez, David Elkouss, David Menendez, Mercedes Soto and Vicente Martin. El Prototipo de Red de Distribución Cuántica de claves UPM-TID. In *XXXII Bienal de Física*, pages 548–549, Ciudad Real, Spain, Sep. 2009.

3. David Elkouss and Jesus Garcia-Lopez. Realimentación en protocolos de distribución cuántica de llave. In *XXXII Bienal de Física*, pages 533–534, Ciudad Real, Spain, Sep. 2009.

4. David Elkouss and Jesus Garcia-Lopez. Protocolos de distribución cuántica de claves. In *Jornadas de Matemática Discreta y Algorítmica*, pages 1–8, Lleida, Spain, Jul. 2008.

5. Ana Yun, David Elkouss, Elisa Callejo, Lei Liang, Linghang Fan and Zhili Sun. Multicast Architecture for IPv6 over DVB-RCS Satellite Networks. In *IP Networking over Next-Generation Satellite Systems*, pages 233–250, Springer New York, 2008.

ACKNOWLEDGMENTS

A doctoral dissertation is never a completely individual and independent work. It is true that most of the research is done by the student himself but aside from the existing state of the art —we stand on the shoulders of giants— the ideas, feedback and support from peers and friends are also an invaluable stimulus for progress.

I was introduced to research while a *double diplôme* student in Télécom ParisTech by Romain Allèaume. He kindly allowed me to do some oriented work already in information reconciliation for Quantum Key Distribution. As early as 2005 he felt that error correcting codes should improve the results of Cascade! Later, as a PhD student, he invited me to spend one academic year with his group. I thank also Anthony Leverrier and Joseph Boutros. Anthony was of extraordinary help, always ready to share his knowledge in information theory, and Joseph generously dedicated some of his limited time in Paris to help me.

This thesis has been under the supervision of Jesús García and Vicente Martín. I am sincerely thankful to both of them. Jesús directed my first line of research on authentication. Jesús was of vital importance my first year, he took the time to teach me the rudiments of the quantum world and has been a source of research topics and interesting conversation. I visited Vicente when I came back from Paris. I knew he was interested on information reconciliation and asked him to join his group. Thereafter he lead most of my research efforts. He provided some guidelines and required some hard results every now and then; but otherwise he granted me freedom in pursuing my research goals, even if they were not completely aligned with those of the group.

A special line is dedicated to Jesús Martínez. He has worked on a very similar topic for his PhD. Very passionate about research, it has been an extraordinary pleasure to have him as a colleague. Nonetheless, his strong work ethic has been an example and a motivation.

I would also like to thank my lab mates all through these years. At Paris: Simon Delamare, Alpha Diallo, Marco Nicoletti, Hany Kamal and Roberto Guerra (who introduced me to the nespresso addiction) where always keen for a *pause café* and some conversation. At UPM: Daniel Lancho and our new PhD student Alex Ciurana have been a joy to work and discuss with.

My colleagues and superiors in my parallel endeavors did understand my schedule constraints and even put some extra work whenever I had a congress or seminar and had to take absence. Thanks to Elisa Callejo and Ana Yun at Thales Alenia Space España, to Pedro

Asúa, Álvaro Cantero, Víctor Corraliza, Miguel Guerrero and Luis Miguel Izquierdo at CECO and Mizuko Uchida at the Embajada de España in Tokyo.

This work has been partially supported in chronological order by: project SEQURE (ANR-07-SESU-011), funded by the Agence Nationale de la Recherche; project PROSPIQ (ANR-06-NANO-041-05), funded by the Agence Nationale de la Recherche; project CENIT SEGURA@ (P0710051085), funded by the Center for the Development of Industrial Technology and project QUITEMAD[1] (P2009/ESP-1594), funded by Comunidad Autónoma de Madrid.

The author gratefully acknowledges the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid [2] (CeSViMa).

1 http://www.quitemad.es
2 http://www.cesvima.upm.es

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Part I

LDPC CODES DESIGN

1 INTRODUCTION

The key to perfect secrecy [...] is to modify Shannon's model such that the enemy cannot receive precisely the same information as the legitimate receiver.

— Ueli M. Maurer [82]

1.1 MOTIVATION

Claude Shannon published his seminal "A mathematical theory of communications" [107] in 1948 after eight years of intermittent work [40]. The paper meant the birth of communications and coding theory. Shannon did not only establish the frame under which communications systems could be studied and compared, he also proved their fundamental limits, i.e. the limiting rates for data compression and reliable transmission through noisy channels. This second result is specially surprising because at the time of the publication there was no certainty that reliable transmission with a positive rate was even possible [71].

A year later, in 1949, Shannon's "Communication theory of secrecy systems" [108] came to light. In words of Robert Gallager "Shannon's cryptography work can be viewed as changing cryptography from an art to a science" [40]. Shannon successfully applied the tools that he had developed in [107] to the problem of transmitting confidential messages through public channels. His main conclusion is that a message from a set of messages sent through a public channel can be obfuscated into a cypher-text with the help of a secret key in such a way that the number of possible originating messages is the whole set of messages, that is, the cypher-text leaks no information to a possible eavesdropper. The condition for this to happen is that the number of secret keys is equal or greater than the number of messages. This condition only applies to eavesdroppers with unbounded resources, if we limit the storage or computing capability of the eavesdropper secret communications are possible without fulfilling the condition. It is evident that computing power resources that today might be considered as out of reach might become available in the near future. There is an implicit risk in assuming that an eavesdropper is limited in any way beyond the fundamental limits that physics impose her, therefore the interest in establishing the scenarios in which some kind of security can be achieved without any assumption is self-evident.

The distribution of secret keys or Secret Key Distribution (SKD) is a problem closely related to confidential communications. Two parties

sharing a secret key can communicate privately through a channel in the conditions discussed in the previous paragraph. We can then study the problem of secret key sharing as a way to achieve confidential communications, though shared secret keys have other uses such as message authentication [126, 114]. The main idea is that two distant parties can agree in a secret key if they have access to a shared source of randomness [5]. The randomness source can take many incarnations, e.g. in the form of a source received from a trusted party or in the form of a noisy channel [5, 82]. It should be stressed that these mathematical models can have a real, i.e. physical correspondence. One such a model is a physical fiber carrying single photons randomly polarized in one of two non-orthogonal basis [7].

In most of the SKD scenarios the legitimate parties obtain instances of correlated sources which means that they obtain similar but not identical strings. It is then assumed that there is an authentic though otherwise public channel available to all parties —including the eavesdropper—. The legitimate parties can exchange additional information through this channel in order to reconcile their strings. They can do so revealing some information about them, for instance the parities of carefully chosen positions. This process is known as information reconciliation [14]. It is not hard to see that the information exchanged through the public channel reduces the uncertainty that the eavesdropper has on the strings of the legitimate parties. A second step known as privacy amplification is then needed [11]. In the privacy amplification step the legitimate parties agree on a secret but shorter key of which the eavesdropper has a negligible amount of information.

Paraphrasing the famous "Experimental quantum cryptography" [10] of Bennett et al. every bit used for information reconciliation has to be sacrificed "in the altar" of privacy. The motivation of this thesis is to study the information reconciliation process and develop efficient, though practical, protocols that allow to optimize the distillation process. We regard optimization from a broader perspective; that is, we aim not only to reduce the messages exchanged during information reconciliation but also to take into account the efficient use of physical resources.

1.2 CONTRIBUTIONS

We consider in this thesis the problem of information reconciliation in the context of secret key agreement between two legitimate parties: Alice and Bob. We discuss in Chap. 3 the design and optimization of LDPC codes and design specific codes for the Binary Symmetric Channel (BSC) over a wide range of rates with thresholds close to the channel capacity. In Chap. 5 we show that LDPC codes optimized for

the BSC can efficiently be used for information reconciliation in some SKD scenarios.

We understand by efficient that a protocol is close to the theoretical limits. We shall see that real SKD scenarios are time variant, the randomness sources differ over time and an efficient method for a specific kind of source might be useless for another. Thus, a good protocol should also tackle this behavior and offer a high efficiency for a wide range of scenarios. We introduce in Chap. 6 an adaptive protocol based in punctured and shortened LDPC codes. The efficiency of the reconciliation is only limited by the quality of the code and, while transmitting more information than needed to reconcile Alice's and Bob's sequences, we prove that it does not reduce any more the uncertainty on the original source than an ad-hoc code would have done.

Puncturing is a well-known coding technique used for constructing rate-compatible families of codes. In Chap. 3 we consider the problem of puncturing LDPC codes and propose a new algorithm for intentional puncturing, where an order within the set of puncturable symbols is defined. This algorithm is shown to improve on the performance of previous proposals.

1.3 STRUCTURE OF THE THESIS

The thesis has been divided in two main parts: LDPC codes design and optimization of information reconciliation.

Chap. 2 and Chap. 3 compose the first part.

In Chap. 2 we introduce basic information theoretic ideas. We begin the chapter deriving the Shannon entropy function from a set of requirements and show its relation with data compression of individual and joint sources. In the second part of the chapter, we consider channel coding and prove the capacity of some families of communications channels that are used later in the thesis.

The objective of Chap. 3 is to introduce coding theory and describe some specific topics related with LDPC codes. After reviewing the basic concepts of linear error correcting codes we describe LDPC codes. The next topic is puncturing techniques for LDPC codes. We conclude the chapter showing that linear codes can be used in the problem of source coding with side information.

In the second part we discuss secret key distillation in Chap. 4 and several information reconciliation methods in Chap. 5. We propose a rate adaptive information reconciliation protocol in Chap. 6.

In Chap. 4 we first compare the computational and information theoretic security paradigms, and then formally define SKD and study the capacity of some of the better known models.

In Chap. 5, we compare several practical information reconciliation protocols. The objective is to show that, although there are several

ad-hoc protocols proposed for the task, error correction codes are an ideal solution from the efficiency point of view. In order to compare the different reconciliation methods we concentrate on reconciliation methods for correlated discrete random variables even if the ideas presented here can be easily extrapolated to other scenarios.

Although linear codes are a good solution for the reconciliation problem, since they can be tailored to a given error rate, their efficiency degrades when it is not known beforehand. This is the case in Quantum Key Distribution (QKD), where the error rate is an a priori unknown that is estimated for every exchange. We introduce in Chap. 6 a rate adaptive protocol. This protocol adapts pre-built codes in real time while maintaining an efficiency close to the optimal value.

We close the text in Chap. 7 with a quick review of the results and a discussion about possible future work.

This thesis is meant to be, to a great extent, self-contained. Even though some chapters of this thesis review known results, the ideas are described in a linear fashion and the results are proved whenever possible. We hope that this effort does not hinder the readability of the text but, on the contrary, clarifies the discussion and even becomes a useful reference.

2 INFORMATION THEORY

It may be no exaggeration to say that man's progress in peace, and security in war, depend more on fruitful applications of information theory than on physical demonstrations [...]

— Fortune (Magazine) [1]

In this chapter we review several basic information theoretic ideas. The chapter follows the standard texts [19, 6, 45, 39, 90, 100, 71, 133] focusing only on the concepts relevant for this thesis. We begin the chapter introducing the Shannon entropy function and show its relation with data compression of individual and joint sources. In the second part of the chapter, we consider channel coding and prove the capacity of some families of communications channels that are used later in the thesis.

2.1 PRELIMINARIES AND NOTATION

The collection of all possible outcomes s in an experiment is called the sample space $\mathcal{S}$. We limit our interest to experiments with a finite number of outcomes. Any subset of the sample space is called an event. Let a and b be two events in $\mathcal{S}$, we define $a \cup b$ and $a \cap b$ as the union and intersection of a and b. $a \cup b$ is the event that contains all outcomes belonging to a, to b and to both. $a \cap b$ is the event that contains all outcomes belonging to both a and b. Two events are disjoint if their intersection is null.

We can define a function $p : \mathcal{S} \to [0, 1]$ that associates every outcome $s \in \mathcal{S}$ with $p(s)$. The extension of p to any $\mathcal{A} \subseteq \mathcal{S}$ is straightforward:

$$p(\mathcal{A}) = \sum_{a \in \mathcal{A}} p(a) \tag{2.1}$$

We say that p is a probability distribution if $\forall s \in \mathcal{S}, p(s) \geqslant 0$ and $p(\mathcal{S}) = 1$. Following Gallager's notation in [39] we call an ensemble $\mathbf{U}$ the tuple of a sample space $\mathcal{S}$ together with a probability distribution p defined on $\mathcal{S}$.

We call a discrete random variable $\mathbf{X}$ over alphabet $\mathcal{X}$ a mapping $\mathbf{X} : \mathcal{X} \to \mathcal{S}$ such that:

$$p_{\mathbf{X}}(x) = \sum_{s:\mathbf{X}(s)=x} p(s) \tag{2.2}$$

We will write $p(x)$ for $p_X(x)$. That is, we will drop the subscript that identifies the ensemble or the random variable whenever there is no possible confusion.

Let us consider a second experiment with two outcomes x and y. The joint sample space of the experiment is the direct product of the sample space associated with the individual outcomes: $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. We can associate, as well, a probability distribution function to map all tuples (x, y) to $[0, 1]$. The probability of an event in the joint experiment is equally defined as the sum of the probability of the individual outcomes. In particular we can define for every $x \in \mathcal{X}$ the probability of $p(x)$ as the sum of $p(x, y)$ for all $y \in \mathcal{Y}$:

$$p(x) = \sum_y p(x, y) \tag{2.3}$$

and equivalently $p(y)$:

$$p(y) = \sum_x p(x, y) \tag{2.4}$$

Let a and b be two events with non zero probability. We call $p(a|b)$ the conditional probability of a given that b occurs. If we repeat the experiment many times it is easy to see that $p(a|b)$ is given by the ratio of $p(a \cap b)$ and $p(b)$. a and b are said to be independent if $p(a \cap b) = p(a)p(b)$. It follows that if and only if a and b are independent $p(a|b) = p(a)$.

We define the variational distance δ between two ensembles $\mathbf{X}$ and $\mathbf{Y}$ defined on the same alphabet $\mathcal{A}$ as:

$$\delta(\mathbf{X}, \mathbf{Y}) = \frac{1}{2} \sum_{a \in \mathcal{A}} |p_X(a) - p_Y(a)| \tag{2.5}$$

δ is a proper metric for ensembles defined on the same alphabet. It is easy to show that it verifies:

$$\delta(\mathbf{X}, \mathbf{Y}) = 0 \iff \mathbf{X} = \mathbf{Y} \tag{2.6}$$

$$\delta(\mathbf{X}, \mathbf{Y}) = \delta(\mathbf{Y}, \mathbf{X}) \tag{2.7}$$

$$\delta(\mathbf{X}, \mathbf{Y}) \leqslant \delta(\mathbf{X}, \mathbf{Z}) + \delta(\mathbf{Z}, \mathbf{Y}) \tag{2.8}$$

Given the joint distribution P_{SZ} we define the distance from uniform by:

$$d(\mathbf{S}|\mathbf{Z}) = \frac{1}{2}\delta(P_{SZ}, P_U \times P_Z) \tag{2.9}$$

where P_U is the uniform distribution.

Random variables and ensembles are denoted with boldface capital letters **A**, **B**, **C**... taking values in sets with calligraphic font $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$... while the elements in a set are denoted with lower case letters a, b, c...

We denote the set of natural, integer, real and complex numbers with the American Mathematical Society (AMS) blackboard bold alphabet letters $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$ respectively.

Arrays and vectors are denoted with boldface lower case letters **a**, **b**, **c**. We write the element of an array in position n_0 as $\mathbf{a}[n_0]$, $\mathbf{b}[n_0]$, $\mathbf{c}[n_0]$... and we denote the subarray expanding from the element in position n_1 to the element in position n_2 as $\mathbf{a}[n_1, n_2]$, $\mathbf{b}[n_1, n_2]$, $\mathbf{c}[n_1, n_2]$... We denote the length of an array or vector as $|\cdot|$, for example $|\mathbf{a}| = n$.

2.2 SOURCE CODING

2.2.1 *A Measure of Information*

We proceed to introduce a measure of the information that the occurrence of an event x in a sample space $\mathcal{X}$ provides to an observer. This measure is related to certainty about the events. If an observer is completely certain that an event is about to happen, the observation that the event indeed happens provides the observer with no additional information, whereas observing an unlikely event yields new information. More formally let **X** be an ensemble. Below we list some intuitive properties that an information measure should possess.

- The occurrence of two independent events should yield the same information that the occurrence of the single events would provide an observer. If we let h be an information measuring function

$$p(a \cap b) = p(a)p(b) \Rightarrow h(a \cap b) = h(a) + h(b) \tag{2.10}$$

  and more generally the information that n independent identical events provide:

$$h(a^n) = nh(a) \tag{2.11}$$

- The measure should be non-negative, that is, an event gives either none or some information, but it can not give negative information:

$$h(a) \geqslant 0 \tag{2.12}$$

- Less probable events provide more information than more probable events. For example, if we think of a coin and a die, an

outcome of the die is more informative than an outcome of the coin:

$$p(a) < p(b) \Rightarrow h(a) > h(b) \tag{2.13}$$

- h should be a continuous function.

We now informally derive a family of functions complying with these basic properties following Shannon's original paper [107]. Several authors have shown that this family is the only one complying with these or related sets of requirements. For a complete discussion on axiomatic derivations of entropy and information please refer to [4, 3, 21, 36]. The following derivation allows us to gain some intuition in the appropriateness of the information measure. However, as the axioms have no inherent validity, this approach "lends a certain plausibility" to the information definitions, "the real justification" of these definitions "resides in their implications" [107].

Let an event with probability $1/r$ be independently repeated m times, we can always define an event with probability $1/t$ independently repeated n times such that r, m, t and n verify:

$$r^m \leqslant t^n < r^{m+1} \tag{2.14}$$

which applying logarithms and operating becomes:

$$\frac{m}{n} \leqslant \frac{\log t}{\log r} < \frac{m+1}{n} \tag{2.15}$$

Given Eq. 2.14 and Eq. 2.13, we can write the following relation between the information that r^m, t^n and r^{m+1} yield:

$$h(r^m) \leqslant h(t^n) < h(r^{m+1}) \tag{2.16}$$

and applying Eq. 2.11:

$$mh(r) \leqslant nh(t) < (m+1)h(r) \tag{2.17}$$

Finally we obtain the form of the information measure by combining Eq. 2.15 with Eq. 2.17 and taking into account that n can take arbitrarily large values:

$$h(t) = \lambda \log t \tag{2.18}$$

with $\lambda < 0$ for the measure to be positive. Choosing different values of λ allows us to measure information with different units.

2.2.2 *Entropy*

Let $\mathbf{X}^n$ be an ensemble that represents a source with n outcomes $x_1, x_2, ..., x_n$. Every outcome x_i is independently and identically distributed by the ensemble $\mathbf{X}$. Then, the probability of an event in the joint sample space is given by:

$$p(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} p(x_i) \tag{2.19}$$

Note that we will use the same notation for an ensemble $\mathbf{X}$ and for the Independent Identically Distributed (iid) source that it spans.

Definition 1. *The average information a symbol in X yields is called the entropy of a source:*

$$H(\mathbf{X}) = -\sum_{x} p(x) \log p(x) \tag{2.20}$$

where we take the convention that $0 \log 0 = 0$, i.e. adding a zero-probability event to a source does not affect its entropy.

The definition that we have just provided reads as the average or mean information that the individual symbols in $\mathbf{X}$ yield; we can then naturally identify the entropy of $\mathbf{X}$ with the expected value of the random variable $-\log p(\mathbf{X})$.

$$H(\mathbf{X}) = -\sum_{x} p(x) \log p(x) = E(-\log p(\mathbf{X})) \tag{2.21}$$

We prove some basic properties of entropy that we will use through this thesis.

Lemma 1. *The entropy is non-negative.*

$$H(\mathbf{X}) \geqslant 0$$

Proof.

$$0 \leqslant p(x) \leqslant 1 \Rightarrow -\log p(x) \geqslant 0 \Rightarrow H(X) \geqslant 0 \tag{2.22}$$

□

Lemma 2. *The distribution that maximizes entropy for any alphabet is the uniform distribution.*

$$H(p_1, ..., p_n) \leqslant \log n$$

Proof.

$$\begin{aligned} H(p_1, ..., p_n) - \log n &= \sum_{i=1}^{n} p_i \log \frac{1}{p_i} - \sum_{i=1}^{n} \frac{1}{n} \log n \\ &= \sum_{i=1}^{n} p_i \log \frac{1}{p_i} - \log n \sum_{i=1}^{n} \frac{1}{n} \\ &= \sum_{i=1}^{n} p_i \log \frac{1}{p_i} - \log n \sum_{i=1}^{n} p_i \\ &= \sum_{i=1}^{n} p_i \log \frac{1}{p_i} - \sum_{i=1}^{n} p_i \log n \\ &= \sum_{i=1}^{n} p_i \log \frac{1}{np_i} \\ &\leqslant \log \sum_{i=1}^{n} \frac{1}{n} = 0 \end{aligned} \tag{2.23}$$

where the second equality follows from the fact that a probability distribution adds up to one and the last inequality holds from log being a concave function and applying Jensen's inequality. □

2.2.3 *Conditional Entropy, Joint Entropy and Mutual Information*

The conditional entropy of a source $\mathbf{X}$ given a second source $\mathbf{Y}$ can be regarded as the average uncertainty that the events in $\mathbf{X}$ provide given that we know the outcomes of another possibly correlated variable Y. Following the reasoning in Sec. 2.2.1, we begin by defining the conditional information of one event a given a second event b:

$$h(a|b) = -\log p(a|b) \tag{2.24}$$

where the conditional information allows us to define the entropy of a source given one event:

$$H(\mathbf{X}|y) = \sum_{x} p(x|y) h(x|y) \tag{2.25}$$

where at the left hand of the equation, we write $H(\mathbf{X}|y)$ as a proxy for $H(\mathbf{X}|\mathbf{Y} = y)$.

The entropy of one source given another is just the weighed average of $H(\mathbf{X}|y)$ for all y.

$$H(\mathbf{X}|\mathbf{Y}) = \sum_{y} p(y) \sum_{x} p(x|y) h(x|y) = \sum_{y} p(y) H(\mathbf{X}|y) \tag{2.26}$$

We prove some basic properties of the conditional entropy.

Lemma 3. *The conditional entropy is non-negative.*

$$H(\mathbf{X}|\mathbf{Y}) \geqslant 0$$

Proof. $H(\mathbf{X}|\mathbf{Y})$ is a sum of entropies, which are positive by Lem. 1, weighed by the probabilities of each event which are also positive. □

Lemma 4. *The entropy of the random variable* $\mathbf{X}$ *given any random variable* $\mathbf{Y}$ *is not greater than the entropy of* $\mathbf{X}$*.*

$$H(\mathbf{X}|\mathbf{Y}) \leqslant H(\mathbf{X})$$

Proof.

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) - H(\mathbf{X}) &= \sum_y p(y) \sum_x p(x|y) \log \frac{1}{p(x|y)} - \sum_x p(x) \log \frac{1}{p(x)} \\ &= \sum_y \sum_x p(x,y) \log \frac{1}{p(x|y)} + \sum_{x,y} p(x,y) \log p(x) \\ &= \sum_{x,y} p(x,y) \log \frac{p(x)}{p(x|y)} \\ &= \sum_{x,y} p(x,y) \log \frac{p(x)p(y)}{p(x,y)} \\ &\leqslant \log \sum_{x,y} p(x)p(y) = 0 \end{aligned} \tag{2.27}$$

□

Lemma 5. *Given random variables* $\mathbf{X}$ *and* $\mathbf{Y}$ *if* $\mathbf{X} = f(\mathbf{Y})$*:*

$$H(\mathbf{X}|\mathbf{Y}) = 0$$

Proof. If $\mathbf{X} = f(\mathbf{Y})$, then given $\mathbf{Y}$ we know $\mathbf{X}$ with absolute certainty, in other words, given $\mathbf{Y}$ there is just one possible outcome.

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) &= \sum_y p(y) H(\mathbf{X}|y) \\ &= 0 \end{aligned} \tag{2.28}$$

□

Definition 2. *Given two discrete random variables* $\mathbf{X}$ *and* $\mathbf{Y}$ *taking values in sets* $\mathcal{X}$ *and* $\mathcal{Y}$ *with joint probability* $p(x,y)$ *we define the joint entropy as:*

$$H(\mathbf{XY}) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{2.29}$$

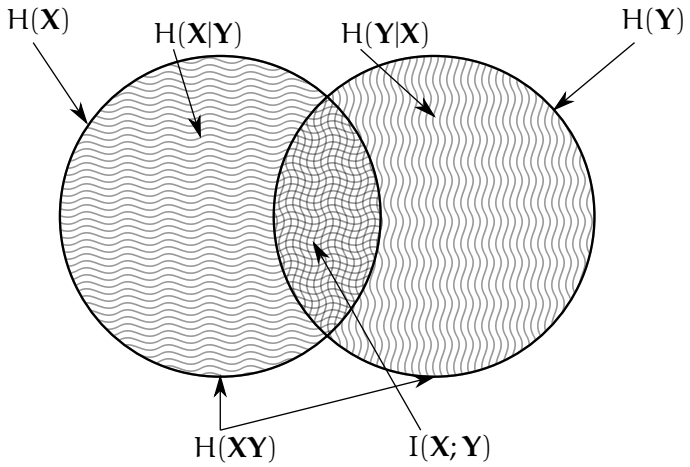

Figure 2.1: Graphical representation of the information measures.

This definition does not introduce a new concept: we can derive a random variable $\mathbf{Z}$ taking values in set $\mathcal{X} \times \mathcal{Y}$ with probability $p(\mathbf{Z} = (x, y)) = p(x, y)$. It is evident that $H(\mathbf{Z}) = H(\mathbf{XY})$ and, the non negativity and maximization by the uniform distribution of $H(\mathbf{XY})$ directly follow.

The joint and conditional entropy definitions can be also naturally extended to multiple variables.

Let $\mathbf{X}$ and $\mathbf{Y}$ be two discrete random variables. The mutual information $I(\mathbf{X};\mathbf{Y})$ is a measure of the information shared between the two variables $\mathbf{X}$ and $\mathbf{Y}$. Fig. 2.1 shows the relationship between the four measures that we have defined: entropy, joint entropy, conditional entropy and mutual information.

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) \\ &= H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) \\ &= I(\mathbf{Y};\mathbf{X}) \end{aligned} \tag{2.30}$$

2.2.4 *Other Entropy Measures*

The entropy as defined by Eq. 2.20 is but one useful measure associated with a random variable or ensemble $\mathbf{X}$. In particular it gives the average uncertainty associated with the outcomes of an experiment. We introduce two related measures; collision entropy, min-entropy and max-entropy [99]. The collision entropy measures the likelihood of two independent outcomes of the same experiment taking the same value:

$$H_2(\mathbf{X}) = -\log \sum_x p_X^2(x) \tag{2.31}$$

Min-entropy is defined as the negative logarithm of the maximum value that $p_{\mathbf{X}}$ takes:

$$H_\infty(\mathbf{X}) = -\log \max_x p_{\mathbf{X}}(x) \tag{2.32}$$

Max-entropy is defined as the logarithm of the support set of $p_{\mathbf{X}}$:

$$H_0(\mathbf{X}) = \log |x : p_{\mathbf{X}}(x) > 0| \tag{2.33}$$

Generally $H_\infty(\mathbf{X}) \leqslant H_2(\mathbf{X}) \leqslant H(\mathbf{X}) \leqslant H_0(\mathbf{X})$, the equality standing if the outcomes in $\mathbf{X}$ are given by a uniform distribution. We further define the conditional collision entropy, min-entropy and max-entropy as:

$$H_2(\mathbf{X}|\mathbf{Y}) = \sum_y p_{\mathbf{Y}}(y) H_2(\mathbf{X}|y) \tag{2.34}$$

$$H_\infty(\mathbf{X}|\mathbf{Y}) = \min_y H_\infty(\mathbf{X}|y) \tag{2.35}$$

$$H_0(\mathbf{X}|\mathbf{Y}) = \max_y H_0(\mathbf{X}|y) \tag{2.36}$$

These quantities are generalized by their smooth versions, the smooth collision entropy, smooth min-entropy and smooth max-entropy, though we are only interested in the second and the third [96, 97]. By smoothing we understand that for a certain $\varepsilon \geqslant 0$ they are respectively minimized and maximized over all events Ω such that $p(\Omega) \geqslant 1 - \varepsilon$.

$$H_\infty^\varepsilon(\mathbf{X}|\mathbf{Y}) = \max_\Omega \min_y \min_x (-\log p_{\mathbf{X}\Omega|\mathbf{Y}}(x|y)) \tag{2.37}$$

$$H_0^\varepsilon(\mathbf{X}|\mathbf{Y}) = \min_\Omega \max_y \log \left|x : p_{\mathbf{X}\Omega|\mathbf{Y}}(x|y) > 0\right| \tag{2.38}$$

where $p_{\mathbf{X}\Omega|\mathbf{Y}}(x|y)$ is the probability that the event Ω takes place and given a specific y $\mathbf{X}$ takes the value x. The min (max) smooth entropy can be identically defined as the maximization (minimization) for all the distributions with variational distance smaller than ε [16, 96].

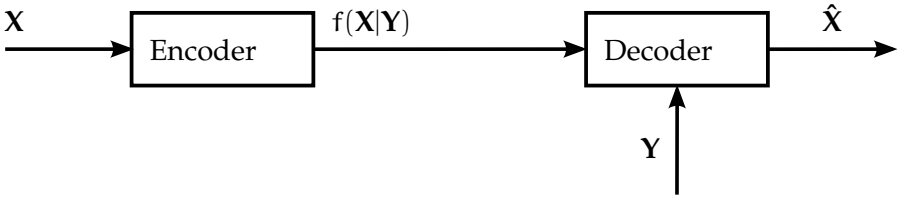

Figure 2.2: Source coding with side information.

2.2.5 *Source Coding with Side Information*

Assume that two distant parties, Alice and Bob, have access two sources $\mathbf{X}$ and $\mathbf{Y}$ and they want to communicate them to a third party Eve. Then, the minimum rate needed to encode both sources independently is $R \geqslant H(\mathbf{X}) + H(\mathbf{Y})$ is needed. This seems a very intuitive result over which not much can be improved, however, in their seminal paper, Slepian and Wolf [113] demonstrated that it is possible to jointly encode both sources at a rate of $R \geqslant H(\mathbf{XY})$. This holds even if $\mathbf{X}$ and $\mathbf{Y}$ are encoded separately.

Now consider a particular case of the scenario described above. Concretely, assume that Bob and Eve are the same party, or in other words $\mathbf{Y}$ is available at the decoder. Then, in the same paper Slepian and Wolf [113] showed that only a rate of $R \geqslant H(\mathbf{X}|\mathbf{Y})$ is needed to encode $\mathbf{X}$ (see Fig. 2.2). This corollary, which is but a corner point in the achievable rate region, is of special interest in this thesis and we shall concentrate on it.

We introduce the concept of the typicality of a sequence and the joint typicality of two sequences to sketch the proof of the Slepian-Wolf bound. Given a sequence $\mathbf{x} = (x_1, x_2, ..., x_n)$ drawn from sampling n times variable $\mathbf{X}$, we can distinguish between two kinds of sequences. Sequences whose entropy is close to the entropy of the source and sequences whose entropy is not close to the entropy of the source. The former we call typical sequences, the latter we call non-typical sequences. We define the typical set $\mathcal{A}_\varepsilon^n$ as follows:

$$\mathcal{A}_\varepsilon^n = \{\mathbf{x} : \left| H(\mathbf{X}) + \frac{1}{n} \log p(\mathbf{x}) \right| \leqslant \varepsilon\} \tag{2.39}$$

We say that two sequences $\mathbf{x}$, $\mathbf{y}$ identically drawn from $p(x, y)$ are jointly typical if 1) $\mathbf{x}$ and $\mathbf{y}$ are typical and 2) also the sequence $\mathbf{x}, \mathbf{y}$ seen as an instance of the joint source $\mathbf{X}Y$ is typical:

$$\left| H(\mathbf{XY}) + \frac{1}{n} \log p(\mathbf{x}, \mathbf{y}) \right| \leqslant \varepsilon \tag{2.40}$$

The encoding we use to sketch the proof (following the lines of the proof in [19]) is known as random binning [129]. The encoder creates

2^{nR} indexed bins and distributes uniformly at random all the typical sequences in the bins. In consequence, the probability that any $\mathbf{x}$ is in a specific bin is 2^{-nR} because the sequences are uniformly distributed in the bins.

Let us begin with an informal discussion on the random binning encoding method. There are approximately 2^{nR} encodings indexing an equal number of bins and $2^{nH(\mathbf{X}|\mathbf{Y})}$ jointly typical sequences for a specific typical sequence $\mathbf{y}$. How should a good encoding look like? A good encoding should permit the decoder to distinguish between different input sequences. So if we restrict only to typical sequences, we pay a prize, in the sense that all non-typical sequences are always going to lead to an error. However, if the number of bins is much larger than the number of jointly typical sequences then the probability that two jointly typical sequences are in the same bin is very small. In other words, with high probability all the sequences jointly typical with $\mathbf{y}$, which account for almost the whole probability of the source, have a different encoding.

The encoding is very simple, the encoder just needs to send the index of the bin $i(\mathbf{x})$. The decoder exploits joint typicality, it chooses $\mathbf{x}'$ belonging to bin $i(\mathbf{x})$ such that $\mathbf{x}'$ and $\mathbf{y}$ are jointly typical. There is an error if $(\mathbf{x},\mathbf{y}) \notin \mathcal{A}_\varepsilon^n$ but also if there exists $\mathbf{x}' \neq \mathbf{x}$ which is jointly typical with $\mathbf{y}$ and shares the same bin index. Both sources of error can be bounded, first $p_{e_1} = p\left((\mathbf{x},\mathbf{y}) \notin \mathcal{A}_\varepsilon^n\right) = 1 - p\left((\mathbf{x},\mathbf{y}) \in \mathcal{A}_\varepsilon^n\right)$ and:

$$\begin{aligned}
p\left((\mathbf{x},\mathbf{y}) \in \mathcal{A}_\varepsilon^n\right) &> 1 - p\left(\left|H(\mathbf{XY}) + \frac{1}{n}\log p(\mathbf{x},\mathbf{y})\right| \geqslant \varepsilon\right) \\
&\quad - p\left(\left|H(\mathbf{X}) + \frac{1}{n}\log p(\mathbf{x})\right| \geqslant \varepsilon\right) \\
&\quad - p\left(\left|H(\mathbf{Y}) + \frac{1}{n}\log p(\mathbf{y})\right| \geqslant \varepsilon\right) \\
&= 1 - p\left(\left|-E(\log p(\mathbf{XY})) + \frac{1}{n}\sum_{i=1}^{n}\log p(x_i,y_i)\right| \geqslant \varepsilon\right) \\
&\quad - p\left(\left|-E(\log p(\mathbf{X})) + \frac{1}{n}\sum_{i=1}^{n}\log p(x_i)\right| \geqslant \varepsilon\right) \\
&\quad - p\left(\left|-E(\log p(\mathbf{Y})) + \frac{1}{n}\sum_{i=1}^{n}\log p(y_i)\right| \geqslant \varepsilon\right) \\
&> 1 - 3\delta
\end{aligned} \tag{2.41}$$

where the second equality comes from rewriting the entropy of a random variable as an expectation (see Eq. 2.21), then the inequality holds by the weak law of large numbers and we can choose $\delta, \varepsilon > 0$ such that for large enough n each of the three differences is greater than ε with probability δ.

Let $p(\mathbf{x}|\mathbf{y})$ be the probability that given $\mathbf{y}$, $\mathbf{x}$ is jointly typical with $\mathbf{y}$, and let $\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})$ be the set of sequences jointly typical with $\mathbf{y}$. The cardinal of $\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})$ is upper bounded by $2^{n[H(\mathbf{X}|\mathbf{Y})+2\delta]}$:

$$\begin{aligned} 1 &\geqslant \sum_{\mathbf{x}\in\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})} p(\mathbf{x}|\mathbf{y}) \\ &= \sum_{\mathbf{x}\in\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})} \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})} \\ &\geqslant \sum_{\mathbf{x}\in\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})} \frac{2^{-n[H(\mathbf{XY})+\delta]}}{2^{-n[H(\mathbf{Y})-\delta]}} \\ &= |\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})| 2^{-n[H(\mathbf{X}|\mathbf{Y})+2\delta]} \end{aligned} \tag{2.42}$$

Let p_{e_2} be the second source of error. We can bound p_{e_2} as follows:

$$\begin{aligned} p_{e_2} &= p\left(\mathbf{x}' \neq \mathbf{x} | i(\mathbf{x}) = i(\mathbf{x}'), (\mathbf{x}', \mathbf{y}) \in \mathcal{A}_\varepsilon^n\right) \\ &\leqslant |\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})|\, 2^{-nR} \\ &\leqslant 2^{-n[R-H(\mathbf{X}|\mathbf{Y})-2\delta]} \end{aligned} \tag{2.43}$$

the probability of having $\mathbf{x}'$ different than $\mathbf{x}$ is lower bounded by the number of sequences jointly typical with $\mathbf{y}$ multiplied by the probability that a sequence is in a specific bin.

We have roughly shown that we can encode $\mathbf{X}$ with rate $R > H(\mathbf{X}|\mathbf{Y})$ and make the sources of error as small as desired. The converse [19] also holds, if the probability of error can be made as small as desired, then $R > H(\mathbf{X}|\mathbf{Y})$.

2.3 CHANNEL CODING

2.3.1 *Communications Channel*

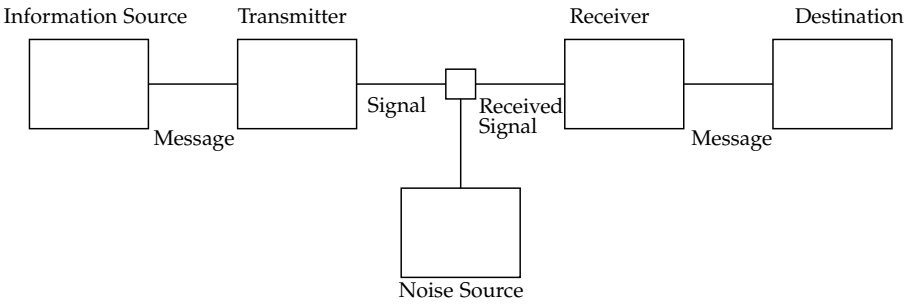

Figure 2.3: This figure reproduces the communications system diagram introduced by Shannon [107].

In this section we address channel coding, but first let us schematically model the communications problem. Fig. 2.3 shows the classical

schema proposed by Shannon [107]. This figure shows five entities: an information source, a transmitter, a noise source, a receiver, and a destination. The communications scheme works as follows:

First the information source generates a message $\mathbf{m}$ from a set of possible messages M. Then, the transmitter takes $\mathbf{m}$ and encodes it into n channel symbols. We define the coding rate R as:

$$R = \frac{\log M}{n} \tag{2.44}$$

The channel is a physical medium of transmission. Mathematically, we can model it as a system taking symbols from input alphabet $\mathcal{X}$ to symbols of output alphabet $\mathcal{Y}$ and characterized by a transition probability matrix that maps the probability of every symbol y if symbol x is sent. The receiver tries to undo the encoding given the noisy received signal and at the end of the scheme the destination receives the $\hat{\mathbf{m}}$ possibly identical to $\mathbf{m}$.

We define C, the capacity of a channel, as the maximum mutual information for all possible input distributions:

$$C = \max_{p(x)} I(\mathbf{X}; \mathbf{Y}) \tag{2.45}$$

2.3.2 *Channel Capacity*

The capacity of a channel specifies the maximum rate at which a source can be reliably sent through a channel. On the other hand, no source with a rate over the capacity of the channel can be sent with a vanishing error probability.

The proof is quite similar to the achievability proof of the Slepian-Wolf bound that we sketched in Sec. 2.2.5. Encoder and decoder share a code-book of 2^{nR} codewords chosen within the $2^{nH(\mathbf{X})}$ typical sequences [77]. The encoder sends a codeword $\mathbf{x}$ drawn with uniform probability. The decoder outputs a word $\hat{\mathbf{x}}$ jointly typical with the received word $\mathbf{y}$. It declares an error if $\mathbf{x}$, $\mathbf{y}$ are not jointly typical and a decoding error can occur if there exists $\mathbf{x}' \neq \mathbf{x}$ jointly typical with $\mathbf{y}$. We know by Eq. 2.41 that the probability of non-joint typicality for long enough n can be made as small as desired.

The intuition behind the achievability proof is simple. The decoder has access to two sets: the set of sequences jointly typical with $\mathbf{y}$, and the set of codewords. If the intersection is to be a single word, every codeword has to be jointly typical with a disjoint set of typical output words.

Approximately, every codeword is jointly typical with $2^{nH(\mathbf{Y}|\mathbf{X})}$ words. Then the number of jointly typical output words with input codewords is upper bounded by $2^{nR+nH(\mathbf{Y}|\mathbf{X})}$, where R is the coding

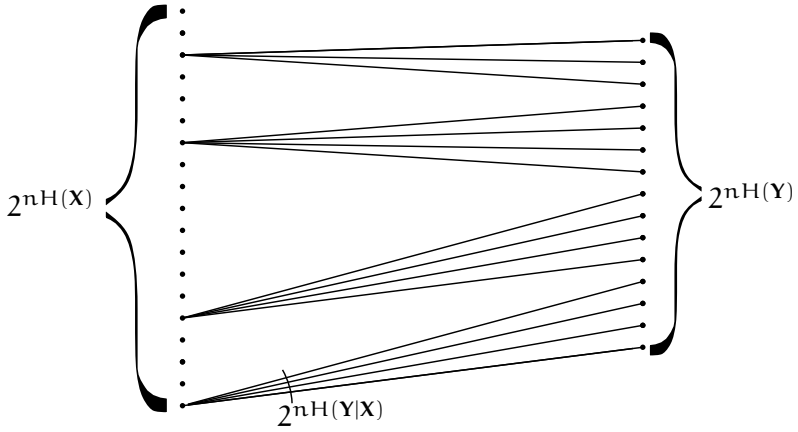

Figure 2.4: Graphical representation of the input and output typical sequences. A good encoding chooses as codewords a subset of the input typical sequences that produces disjoint sets of output typical sequences.

rate. This number should be much smaller than the total number of typical sequences $2^{nH(\mathbf{Y})}$:

$$2^{nR+nH(\mathbf{Y}|\mathbf{X})} < 2^{nH(\mathbf{Y})}$$

which operating returns the expected result:

$$R < I(\mathbf{X};\mathbf{Y})$$

In conclusion, as long as the coding rate is below the mutual information between input and output for n long enough we can construct a code that allows the decoder to distinguish between codewords with a vanishing probability of error.

The converse statement follows from Fano's inequality [35]. The intuition behind this part is that if we think of an encoding that achieves a vanishing error probability, then necessarily $R < I(\mathbf{X};\mathbf{Y})$ [19]. Again the proof is very similar to the converse result in the Slepian-Wolf bound.

2.3.3 *The capacity of some basic channels*

2.3.3.1 *Binary Symmetric Channel*

In the BSC the binary elements or bits are either perfectly transmitted with probability $1-p$ or flipped with probability p.

Let us first find the mutual information between the input $\mathbf{X}$ and the output $\mathbf{Y}$ [19]::

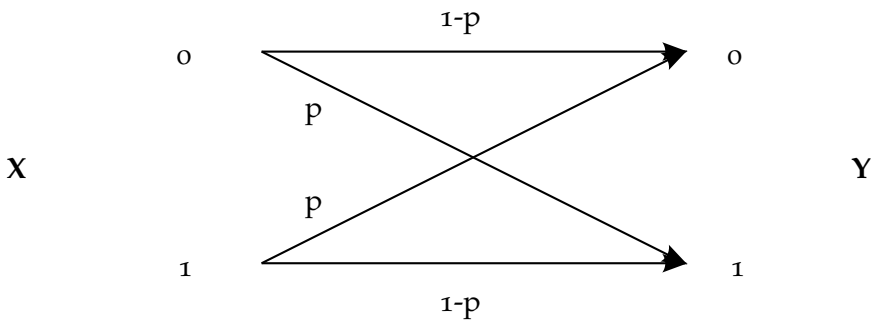

Figure 2.5: Binary Symmetric Channel.

$$
\begin{aligned}
I(\mathbf{X};\mathbf{Y}) &= H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) && (2.46)\\
&= H(Y) - \sum_x p(x) H(\mathbf{Y}|x) && (2.47)\\
&= H(Y) - \sum_x p(x) H(p, 1-p) && (2.48)\\
&= H(Y) - H(p, 1-p) \sum_x p(x) && (2.49)\\
&\leqslant 1 - H(p, 1-p) && (2.50)
\end{aligned}
$$

We obtain the capacity by finding the maximum of the mutual information for all possible input distributions. It can be easily verified that the the uniform distribution reaches the upper bound in Eq. 2.50 and the capacity of the BSC is one minus the binary entropy of p.

2.3.3.2 *Binary Erasure Channel*

The Binary Erasure Channel (BEC) was introduced by Elias in his famous paper "Coding for Two Noisy Channels" [27]. The BEC has two input elements while the output alphabet is composed of three elements: 0, 1, and e, which stands for an erasure in the channel. In this channel the bits are either correctly transmitted with probability $1-p$, or are erased with probability p.

We can first find $H(\mathbf{X}|\mathbf{Y})$:

$$
\begin{aligned}
H(\mathbf{X}|\mathbf{Y}) &= \pi(1-p) H(\mathbf{X}|\mathbf{Y}=0)\\
&\quad + (\pi p + (1-\pi)p)\, H(\mathbf{X}|\mathbf{Y}=e)\\
&\quad + (1-\pi)(1-p) H(\mathbf{X}|\mathbf{Y}=1) && (2.51)\\
&= p && (2.52)
\end{aligned}
$$

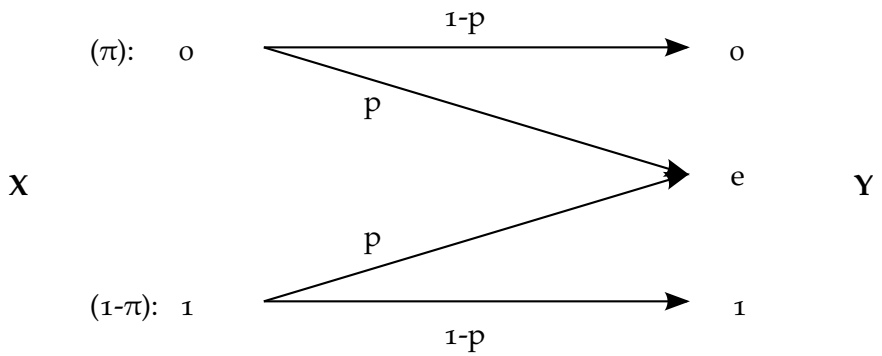

Figure 2.6: Binary Erasure Channel.

where $p(\mathbf{X} = 0) = \pi$.The second equality holds from $H(\mathbf{X}|\mathbf{Y} = 1) = H(\mathbf{X}|\mathbf{Y} = 0) = 0$ and $H(\mathbf{X}|\mathbf{Y} = e) = 1$. We can now plug Eq. 2.51 in Eq. 2.30 and bound from above the mutual information:

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}) &= H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) && (2.53)\\ &= H(\pi, 1-\pi) - p && (2.54)\\ &\leqslant 1 - p && (2.55) \end{aligned}$$

equality in Eq. 2.55 is achieved again by the uniform distribution. That is, for $\pi = \frac{1}{2}$.

It might seem that the capacity of a BSC that flips bits with probability p is greater than the capacity of a BEC that erases bits with probability p. Fig. 2.7 shows that it is the opposite situation. On the range $p \in (0, 0.5)$, the capacity of the BEC is greater than the capacity of the BSC. Bits on the BEC are either perfectly known or perfectly unknown, however, it is not possible to distinguished flipped bits from correct bits in the BSC.

2.3.4 *Degraded channels*

The two families of noisy channels just discussed, the BEC and the BSC, are parametrized by p. It is intuitive that p is a measure of the amount of noise in the channel. If we fix the type of channel, then we say that the channel characterized by $p_1 > p_2$ is a degraded version of the channel characterized by p_2 [19, 101]. We formally define a channel $C(\varepsilon')$ a (physically) degraded version of $C(\varepsilon)$ if:

$$p(yy'|x) = p(y|x)p(y'|y) \tag{2.56}$$

We show graphically in Fig. 2.8 and Fig. 2.9 that if $p_1 > p_2$ both the BEC(p_1) and the BSC(p_1) are respectively degraded versions of BEC(p_2) and the BSC(p_2).

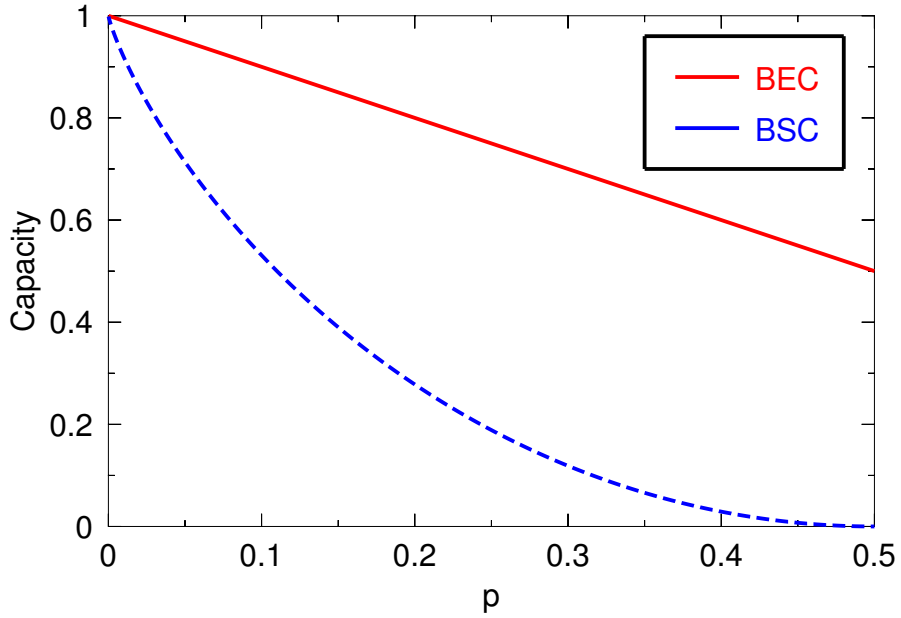

Figure 2.7: The capacity of the BEC and BSC.

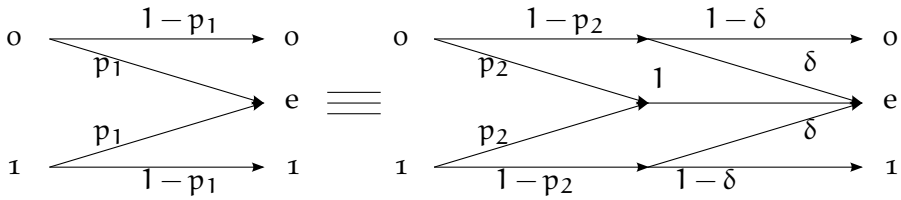

Figure 2.8: We see in this figure that if $p_1 > p_2$ the BEC(p_1) is equivalent to the BEC(p_2) concatenated with a ternary channel where $\delta = (p_1 - p_2)/(1 - p_2)$.

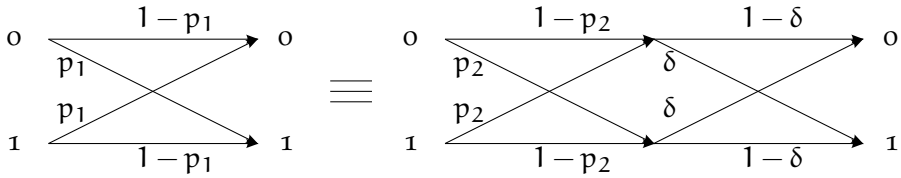

Figure 2.9: We see in this figure that if $p_1 > p_2$ the BSC(p_1) is equivalent to the concatenation of two BSC, one characterized by p_2 and one by $\delta = (p_1 - p_2)/(1 - 2p_2)$.

3

LOW DENSITY PARITY CHECK CODES

Claude Shannon

Born on the planet Earth (Sol III) in the year 1916 A.D. Generally regarded as the father of the Information Age, he formulated the notion of channel capacity in 1948 A.D. Within several decades, mathematicians and engineers had devised practical ways to communicate reliably at data rates within 1% of the Shannon limit...

— Encyclopedia Galactica, 166th ed. [83]

The objective of this chapter is to introduce coding theory and describe some specific topics related to LDPC codes. We first introduce basic notation and concepts of linear error correcting codes. We then describe LDPC codes, their decoding as well as some techniques for the design and optimization of LDPC codes.

The next topic is puncturing techniques for LDPC codes that we use later in the thesis to adapt the coding rate of an information reconciliation protocol. These code construction, design and rate-adaption techniques were presented in [28, 75, 32].

We conclude the chapter with syndrome coding, a very useful technique that allows the use of channel codes for source coding.

3.1 INTRODUCTION TO CODING

3.1.1 *Block Codes*

A code $\mathcal{C}(n, k)$ is called a block code if it maps a source message of k symbols into a codeword of n symbols. We say that it is a code of length n that transmits k symbols from the source, also known as information symbols, with every codeword. We restrict our study to binary codes, and in consequence we may safely replace symbols with bits. A block code is used to transmit codewords through noisy channels, the remaining $n - k$ bits add redundancy to the information bits and help the decoder recover the transmitted codeword. In a discussion regarding a code $\mathcal{C}(n, k)$ we will drop the dimension indexes n and k whenever they are unnecessary. We can already define some properties for binary block codes; the rate of a code is the proportion of information bits in a codeword:

$$R = \frac{k}{n} \tag{3.1}$$

·	0	1
0	0	0
1	0	1

+	0	1
0	0	1
1	1	0

Table 3.1: Arithmetic in $\mathbb{F}_2$.

The Hamming distance separating two codewords, which can be represented as row vectors, $\mathbf{x} = (x_1, ..., x_i, ..., x_n)$, $\mathbf{y} = (y_1, ..., y_i, ..., y_n)$ is defined as:

$$d(\mathbf{x}, \mathbf{y}) = |\{i | x_i \neq y_i\}| \tag{3.2}$$

The $n - k$ redundancy bits are used to place words far from each other distance wise, such that they can be easily differentiated even if slightly corrupted. The minimum distance of a code d_{min} is the minimum distance separating two codewords.

$$d_{min} = \min\{d(\mathbf{x}, \mathbf{y}) | \forall \mathbf{x}, \mathbf{y} \in \mathcal{C}\} \tag{3.3}$$

If a codeword is corrupted by a BSC as long as less than d_{min} bits are flipped we can detect that the word is erroneous. On the other hand, if d_{min} or more bits are flipped then the codeword can be transformed into another correct codeword and the corrupted word could pass as a correct one.

A measure related to the distance of two codewords is the weight of a codeword $w(\mathbf{x})$. It is defined as the distance of $\mathbf{x}$ to the word all zeros:

$$w(\mathbf{x}) = d(\mathbf{x}, 0) = |\{i | x_i = 1\}| \tag{3.4}$$

3.1.2 *Linear Codes*

Linear codes are block codes with a specific algebraic structure. A (binary) block code is a (binary) linear code $\mathcal{C}$ if it forms a vector space of dimension k over $\mathbb{F}_2$, the finite field containing two elements. $\mathbb{F}_2$ has two elements which we can label $0, 1$ and the arithmetic operations are performed modulo 2:

A generator matrix G is a matrix of dimension $k \times n$. The rows of G form a basis of the space induced by $\mathcal{C}$, it defines a linear transformation from 2^k into 2^n, in other words G maps k bits information into a codeword:

$$\mathcal{C} = \{\mathbf{a}G | \mathbf{a} \in \mathbb{F}_2^k\} \tag{3.5}$$

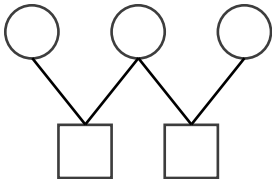

Figure 3.1: This figure depicts the Tanner graph of the repetition code in Ex. 1. The graph has four edges joining bits 1 and 2 with check 1, and bits 2 and 3 with check 2 following H.

Please note the difference between a code $\mathcal{C}$, which defines a set of codewords, from an encoder, which defines a specific map from blocks of k bits to codewords. There can be many possible encoders for a code $\mathcal{C}$.

The generator matrix is said to be in standard form if we can write it as $G = (P|I_k)$, that is the concatenation of P a $n-k \times k$ matrix and the identity matrix of size $k \times k$. If G is in standard form the code is said to be in systematic form and the code maps the k information bits into the last k bits of the codeword. In general a code is said to be systematic if the k information bits are embedded in known positions of the codeword, and a permutation of the bit positions would allow to write all the codewords of $\mathcal{C}$ as $c(\mathbf{x}) = \mathbf{x}|\mathbf{r}$, i.e. the codeword can be seen as as the original word and some redundancy.

H, an $(n-k) \times n$ matrix, is a parity matrix for code C if it is full rank and it verifies $GH^T = 0$. This relation implies that the product of any codeword with the transposed of the parity matrix is also the zero vector of size $n-k$. It can be easily shown that it works both ways, that is, if a word multiplied by H transposed is the zero vector then it is a codeword because necessarily it is spanned by G.

$$\mathbf{x} \in \mathcal{C} \Leftrightarrow \mathbf{x}H^T = \mathbf{0} \tag{3.6}$$

Each of the $n-k$ independent rows in H conforms a linear equation. These equations are called parity check equations because an equation is verified only if the bits involved add up to an even number or equivalently to $0 \mod 2$. We call the syndrome of a word $\mathbf{x} \in \mathbb{F}_2^n$, $s(\mathbf{x})$ the map $s : \mathbb{F}_2^n \to \mathbb{F}_2^k$ defined by Eq. 3.6.

Linear codes can be represented by bipartite graphs known in this context as Tanner graphs [118]. The two disjoint sets are the check nodes and the variable nodes. A set of $n-k$ check nodes represent the set of parity-check equations which define the code; a set of n variable nodes represent the bits. If we number the checks from 1 to $n-k$ and the variable nodes from 1 to n, the graph is formed by drawing an edge between check i and bit j if $H[i,j] = 1$ (see Fig. 3.1). There is a one to one correspondence between both representations as the process can be easily inverted.

Example 1. *We consider one of the most simple examples, the so called repetition code which maps a single bit into a codeword that repeats its value three times. It is a code of rate one third with* G *and* H *given by:*

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

The code has just two codewords (000) *and* (111) *that map* 0 *and* 1 *respectively.*

3.1.3 *Decoding*

After briefly discussing the issues of encoding information symbols into codewords, we regard the opposite operation. A decoder receives y and tries to recover the original source message. If $\mathbf{y}H^T = \mathbf{0}$ then $\mathbf{y}$ is a codeword and the receiver performs the inverse map from codewords into information symbols to obtain the source message. However if $\mathbf{y}H^T \neq \mathbf{0}$ the receiver first needs to choose a candidate codeword given that he received $\mathbf{y}$.

The first decoder that we shall consider is called the Maximum Likelihood (ML) decoder. It behaves in the following way; given $\mathbf{y}$ the decoder chooses $\hat{\mathbf{c}}$ the codeword that maximizes the a priori probability $p(\mathbf{y}|\mathbf{c})$:

$$\hat{\mathbf{c}}_{ML} = \underset{\mathbf{c}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{c}) \tag{3.7}$$

In order to maximize $p(\mathbf{y}|\mathbf{c})$ in Eq. 3.7 the decoder needs to compare all the codewords in the code, this task can only be performed for very short codes as the number of codewords explodes exponentially with k as 2^k. It was shown by Berlekamp et al. [12] that ML decoding is an NP-complete problem for binary linear codes. In terms of complexity classes the algorithms that can provide an answer in polynomial time are called P, while a problem for which the correctness of an answer can be checked in polynomial time are called NP problems. It is thus, "unlikely that anyone will ever discover substantially faster algorithms" than this exponential search, quoting the original paper.

The maximization performed by the ML decoder can be written as a distance comparison in the case of the BSC(ε); in this case ML decoding is equivalent to finding the codeword that minimizes the hamming distance with the received vector:

$$\begin{aligned}\hat{\mathbf{c}}_{\mathrm{ML}} &= \operatorname*{argmax}_{\mathbf{c}} p(\mathbf{y}|\mathbf{c}) \\ &= \operatorname*{argmax}_{\mathbf{c}} \prod_{i=1}^{n} p(y_i|c_i) \\ &= \operatorname*{argmax}_{\mathbf{c}} \varepsilon^{d(\mathbf{y},\mathbf{c})}(1-\varepsilon)^{n-d(\mathbf{y},\mathbf{c})} \\ &= \operatorname*{argmin}_{\mathbf{c}} d(\mathbf{y},\mathbf{c}) \end{aligned} \tag{3.8}$$

The ML decoder maximizes the a priori likelihood but it is not perfect; note that if the number of errors exceeds a threshold the closest codeword can be different to the codeword sent. For instance in Ex. 1 if the codeword (000) is sent and (011) is received the ML decoder outputs (111).

We now describe a second decoder, the Maximum a Posteriori (MAP) decoder. This decoder maximizes the a posteriori probability $p(\mathbf{c}|\mathbf{y})$:

$$\hat{\mathbf{c}}_{\mathrm{MAP}} = \operatorname*{argmax}_{\mathbf{c}} p(\mathbf{c}|\mathbf{y}) \tag{3.9}$$

If the codewords are not all equiprobable both decoders might render different results. However, for a uniform distribution on the codewords, both decoders are identical:

$$\begin{aligned}\hat{\mathbf{c}}_{\mathrm{MAP}} &= \operatorname*{argmax}_{\mathbf{c}} p(\mathbf{c}|\mathbf{y}) \\ &= \operatorname*{argmax}_{\mathbf{c}} \frac{p(\mathbf{c},\mathbf{y})}{p(\mathbf{c})} \\ &= \operatorname*{argmax}_{\mathbf{c}} \frac{p(\mathbf{y})}{p(\mathbf{c})} p(\mathbf{y}|\mathbf{c}) \\ &= \operatorname*{argmax}_{\mathbf{c}} \frac{\sum_{\tilde{\mathbf{c}}} p(\mathbf{y}|\tilde{\mathbf{c}})p(\tilde{\mathbf{c}})}{p(\mathbf{c})} p(\mathbf{y}|\mathbf{c}) \end{aligned} \tag{3.10}$$

of course if all the words are equiprobable $p(\mathbf{c})$ cancels out, and we get:

$$\begin{aligned}\hat{\mathbf{c}}_{\mathrm{MAP}} &= \operatorname*{argmax}_{\mathbf{c}} \underbrace{\left(\sum_{\tilde{\mathbf{c}}} p(\mathbf{y}|\tilde{\mathbf{c}})\right)}_{\text{constant}} p(\mathbf{y}|\mathbf{c}) \\ &= \operatorname*{argmax}_{\mathbf{c}} p(\mathbf{y}|\mathbf{c}) = \hat{\mathbf{c}}_{\mathrm{ML}} \end{aligned} \tag{3.11}$$

which means that both decoders are output the same codeword.

The two decoders that we have defined operate on the whole codeword. We can also define symbol-wise versions of the ML and

MAP decoders. That is, decoders that maximize respectively the a priori or the a posteriori likelihood of a symbol:

$$\hat{c}_{i_{ML}} = \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, p(\mathbf{y}|c_i) \tag{3.12}$$

$$\hat{c}_{i_{MAP}} = \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, p(c_i|\mathbf{y}) \tag{3.13}$$

We can show that the symbol-wise decoders are also equivalent if the codeword source is uniform:

$$\begin{aligned}
\hat{c}_{i_{MAP}} &= \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, p(c_i|\mathbf{y}) \\
&= \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, \frac{p(c_i)}{p(\mathbf{y})} p(\mathbf{y}|c_i) \\
&= \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, \underbrace{\frac{p(c_i)}{\sum_{\mathbf{c}} p(\mathbf{c}) p(\mathbf{y}|\mathbf{c})}}_{\text{constant}} p(\mathbf{y}|c_i) \\
&= \underset{c_i \in \{0,1\}}{\operatorname{argmax}}\, p(\mathbf{y}|c_i) = \hat{c}_{i_{ML}}
\end{aligned} \tag{3.14}$$

The MAP decoder outputs the symbol that maximizes $p(c_i|\mathbf{y})$. This quantity can be simplified to the multiplication of the individual bit a priori probabilities if the channel is memoryless:

$$\begin{aligned}
p(c_i|\mathbf{y}) &= \frac{p(\mathbf{y}, c_i)}{p(\mathbf{y})} \\
&= \frac{1}{p(\mathbf{y})} \sum_{\substack{\mathbf{c} \in \mathcal{C} \\ \mathbf{C}_i = c_i}} p(\mathbf{c}, \mathbf{y}) \\
&= \frac{p(\mathbf{c})}{p(\mathbf{y})} \sum_{\substack{\mathbf{c} \in \mathcal{C} \\ \mathbf{C}_i = c_i}} p(\mathbf{y}|\mathbf{c}) \\
&= \frac{p(\mathbf{c})}{p(\mathbf{y})} \sum_{\substack{\mathbf{c} \in \mathcal{C} \\ \mathbf{C}_i = c_i}} \prod_{j=1}^{n} p(y_j|c_j)
\end{aligned} \tag{3.15}$$

Whatever value $\mathbf{C}_i$ takes, if the source is uniform, $\alpha = p(\mathbf{c})/p(\mathbf{y})$ is a constant value multiplying the sum of products. The argument that maximizes the symbol-wise MAP decoder is independent of α and the Sum Product Algorithm (SPA) that we introduce for decoding LDPC codes in the next section takes its name from the sum-product form of Eq. 3.15.

In general, we are not considering zero error coding [109]. Even if an ML or MAP decoder were available, there is always a non-zero probability of decoding error if words are transmitted through a noisy channel. The Frame Error Rate (FER) measures the ratio of wrong codewords for a given channel and decoder. A related measure is the Binary Error Rate (BER), which measures the ratio of wrong bits for a given channel and decoder. The BER is usually much lower than the FER, and in any case BER $\leqslant$ FER, because even if a wrong codeword is output, the wrong output will be probably close in terms of the the hamming distance, to the right codeword. Depending on the application it is more interesting to consider one or the other figure of merit.

3.1.4 *Coset codes*

We call the set spanned by adding a vector $\mathbf{a} \in \mathbb{F}_2^n$ to the codewords in $\mathcal{C}$, i.e. $\{\mathbf{x}+\mathbf{a} | \mathbf{x} \in \mathcal{C}\}$ a coset of a code $\mathcal{C}$ or simply a coset code. All vectors $\mathbf{a} \in \mathbb{F}_2^n$ are in some coset of $\mathcal{C}$, in effect, $\mathbf{a} = \mathbf{a} + \mathbf{0} \in \mathbf{a} + \mathcal{C}$. Cosets have some interesting properties, we will only show the equivalence one to one between syndromes and cosets (see [52]).

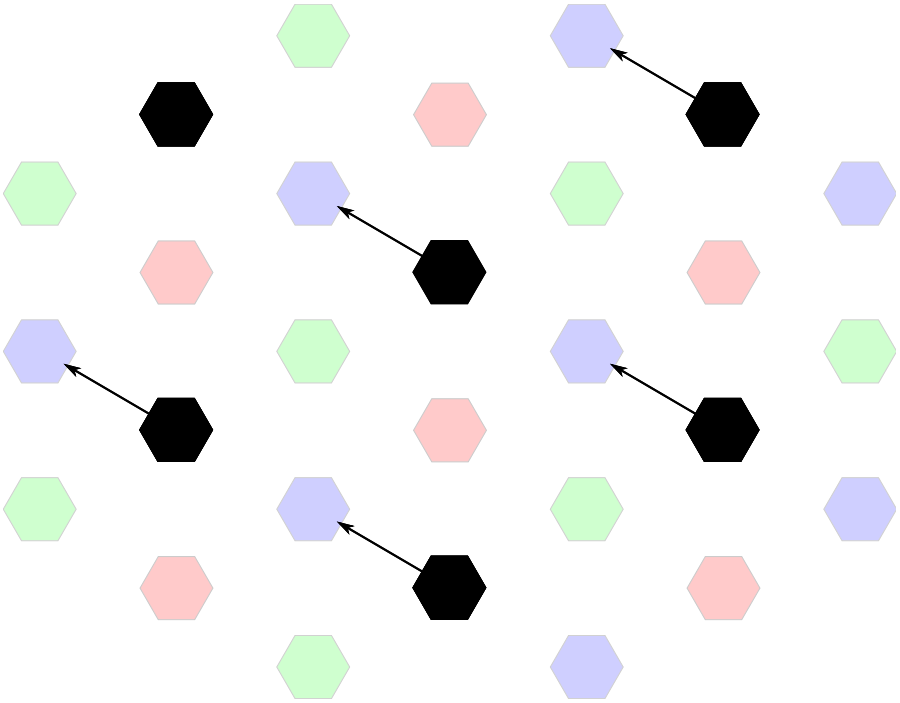

Figure 3.2: For graphical representation of the coset codes, we have chosen a different color for each coset. A fixed vector addition moves vectors from the black coset to the blue coset.

Lemma 6. *If* $\mathbf{x} \in \mathbf{y} + \mathcal{C}$ *then* $\mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$

Proof. If $\mathbf{x} \in \mathbf{y} + \mathcal{C}$ then $\exists \mathbf{c}_1 \in \mathcal{C} | \mathbf{x} = \mathbf{y} + \mathbf{c}_1$. Now let $\mathbf{c}_2 + \mathbf{x} \in \mathbf{x} + \mathcal{C}$ and $\mathbf{c}_3 + \mathbf{y} \in \mathbf{y} + \mathcal{C}$, with $\{\mathbf{c}_2, \mathbf{c}_3\} \in \mathcal{C}$.

$$\mathbf{c}_2 + \mathbf{x} = \mathbf{c}_2 + (\mathbf{y} + \mathbf{c}_1) = (\mathbf{c}_1 + \mathbf{c}_2) + \mathbf{y} \in \mathbf{y} + \mathcal{C} \tag{3.16}$$

$$\mathbf{c}_3 + \mathbf{y} = \mathbf{c}_3 + (\mathbf{x} - \mathbf{c}_1) = (\mathbf{c}_3 - \mathbf{c}_1) + \mathbf{x} \in \mathbf{x} + \mathcal{C} \tag{3.17}$$

We have by Eq. 3.16 that $\mathbf{x} + \mathcal{C} \subset \mathbf{y} + \mathcal{C}$ and by Eq. 3.17 that $\mathbf{y} + \mathcal{C} \subset \mathbf{x} + \mathcal{C}$, hence $\mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$. □

Lemma 7.

$$s(\mathbf{x}) = s(\mathbf{y}) \Leftrightarrow \mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$$

Proof. By the definition of syndrome we know that $H\mathbf{x} = H\mathbf{y}$. This is equivalent of saying that $\mathbf{x} - \mathbf{y} \in \mathcal{C}$. Then $\mathbf{x} = \mathbf{y} + (\mathbf{x} - \mathbf{y}) \in \mathbf{y} + \mathcal{C}$ and by Lem. 6 $\mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$. The arguments can be followed backwards to prove the other direction of the relation. □

We call the minimum weight word in a coset code the coset leader. Let f be a function $\mathbb{F}_2^{n-k} \to \mathbb{F}_2^n$ that given a syndrome outputs the coset leader, f can be used to implement a decoding procedure known as syndrome decoding. If $\mathbf{y} = \mathbf{x} + \mathbf{z}$ are the output, input and noise vectors in a BSC, the receiver can choose $\hat{\mathbf{z}} = f(H\mathbf{y})$, which is the vector closest to $\mathbf{z}$ in the same coset, as his estimation for $\mathbf{z}$ and compute his estimate for $\mathbf{x}$ as $\hat{\mathbf{x}} = y + \hat{\mathbf{z}}$.

3.2 LDPC CODES

3.2.1 *Introduction*

LDPC codes are linear codes with a sparse parity check matrix, sparse in the sense that the density of non zero coefficients is low. The interest in LDPC codes arises from the fact that low complexity, suboptimal algorithms are available for codes with a low density parity check matrix.

Regular LDPC codes where first proposed by Gallager in 1963 in his PhD thesis [38]. However due to their resource requirements, but also because other types of codes were thought to be better for real applications [70], little attention was given to them for almost 30 years until the work of MacKay and Neal [72, 73], Luby et al. [67, 68] and Wiberg [128, 127] among others drew back interest on LDPC codes.

We call an LDPC code regular (d_v, d_c) if every bit is in d_v parity check equations and d_c bits form each parity equation. The parity check matrix in Ex. 1 is not regular as bits 1 and 3 are in one parity check equation while bit 2 is in two equations. Fig. 3.3 shows the parity check matrix and the Tanner graph of a regular code.

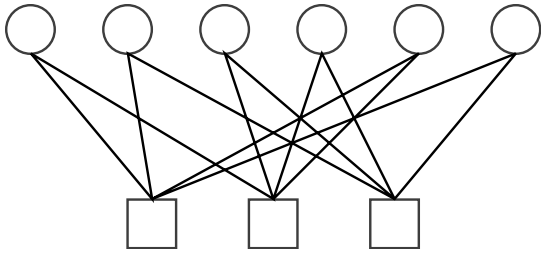

Figure 3.3: The figure depicts the Tanner graph of a regular $(2,4)$ code.

The number of non zero entries, that is the number of edges joining bits and checks can be computed from the bit and the check point of view.

$$n d_v = (n-k) d_c \tag{3.18}$$

This relation and Eq. 3.1 allow us to write the rate of the code as a function of d_c and d_v:

$$R = 1 - \frac{d_v}{d_c} \tag{3.19}$$

In an irregular LDPC code not all bits belong to the same number of equations and/or not all parity check equations, are formed with the same number of bits. We say that a bit is of degree d_v if it belongs to d_v parity check equations while we say that a check is of degree d_c if d_c bits form the parity equation. Let δ_v be the maximum variable degree and δ_c the maximum check degree, we can define $\lambda'(x)$ and $\rho'(x)$ two polynomials that represent the degree distributions of bits and checks:

$$\begin{aligned} \lambda'(x) &:= \sum_{i=2}^{\delta_v} \lambda'_i x^{i-1} \qquad 0 \leqslant \lambda_i \leqslant 1 \\ \rho'(x) &:= \sum_{i=2}^{\delta_c} \rho'_i x^{i-1} \qquad 0 \leqslant \rho_i \leqslant 1 \end{aligned} \tag{3.20}$$

where we denote by λ'_i (ρ'_i) the fraction of bit (check) nodes of degree i. We can extend Eq. 3.18 to incorporate codes with irregular degree distributions:

$$n \sum_{i=2}^{\delta_v} \lambda'_i i = (n-k) \sum_{i=2}^{\delta_c} \rho'_i i \tag{3.21}$$

And we can get a similar equation for the rate:

$$R = 1 - \frac{\sum_{i=2}^{\delta_v} \lambda'_i i}{\sum_{i=2}^{\delta_c} \rho'_i i} \tag{3.22}$$

3.2.2 *Sum Product Algorithm*

We saw in Sec. 3.1 that the problem of ML decoding for binary linear codes is an NP-complete problem. Instead of computing the posterior probability of all codewords the SPA exploits the graph structure of linear codes and locally computes the MAP symbol-wise.

This algorithm is an instance of a Message Passing Algorithm (MPA). MPAs are algorithms that can be described as passing messages through the edges of the Tanner graph.

The SPA exchanges soft values. In contrast with other algorithms such as the Bit Flipping Algorithm [55] that exchange messages taking values in a discrete alphabet the SPA sends messages representing probabilities or in some versions the Log Likelihood Ratio (LLR) of probabilities.

Log Likelihood Ratio

The concept of LLR in the coding context was reviewed by Hagenauer et al. in 1996 [50]. Let $\mathbf{X}$ be a binary random variable taking the values $\{0, 1\}$ with $p(\mathbf{X} = 0) = \varepsilon$ we define the LLR of $\mathbf{X}$ as:

$$l(\mathbf{X}) = \log \frac{\varepsilon}{1 - \varepsilon} \tag{3.23}$$

We can think of the sign of $l(\mathbf{X})$ as the hard decision on $\mathbf{X}$, that is, if we think that $\mathbf{X}$ is more likely to be a 0 or a 1. While we can regard $|L(\mathbf{X})|$ as the reliability of the hard decision. A simple manipulation of Eq. 3.23 gives the following useful relation:

$$p = \frac{e^{l(\mathbf{X})}}{1 + e^{l(\mathbf{X})}} \tag{3.24}$$

It might seem that the LLR fails to keep all the relevant information for decoding. However it is a sufficient statistic for both the MAP and the bit-wise MAP decoder. More formally we say that given a channel $p(\mathbf{Y}|\mathbf{X})$ and a random variable $\mathbf{Z}$ characterized as a function of the output $\mathbf{Z} = f(\mathbf{Y})$, $\mathbf{Z}$ is a sufficient statistic if $p(\mathbf{Y}|\mathbf{X}) = a(\mathbf{X}, \mathbf{Z})b(\mathbf{Y})$ [102]; the reason being that the maximization of the MAP decoder does not depend on $\mathbf{Y}$ (see Eq. 3.9). $\mathbf{L} = f(\mathbf{Y})$ is a sufficient statistic for the bit-wise decoder:

Lemma 8.

$$p(\mathbf{y}|x_i) = a(x_i, l)b(\mathbf{y})$$

Proof.

$$\begin{aligned} p(\mathbf{y}|x_i) &= \frac{p(\mathbf{x})}{p(x_i)} \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = x_i}} \prod_{j=1}^{n} p(y_j|x_j) \\ \frac{p(\mathbf{y}|x_i)}{\prod_{j=1}^{n} p(y_j|\mathbf{X}_j = 1)} &= \frac{p(\mathbf{x})}{p(x_i)} \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = x_i}} e^{\sum_{j=1}^{n} l_j(1-x_j)} \end{aligned}$$

where in the second equation we have divided both sides by $\prod_{j=1}^{n} p(y_j|\mathbf{X}_j = 1)$ and $l_j = \log p(y_j|\mathbf{X}_j = 0)/p(y_j|\mathbf{X}_j = 1)$.

$$\begin{aligned} p(\mathbf{y}|x_i) &= \left(\frac{p(\mathbf{x})}{p(x_i)} \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = x_i}} e^{\sum_{j=1}^{n} -l_j x_j} \right) \\ &\quad \cdot \left(e^{\sum_{j=1}^{n} l_j} \prod_{j=1}^{n} p(y_j|\mathbf{X}_j = 1) \right) \\ p(\mathbf{y}|x_i) &= a(x_i, \mathbf{L} = l) b(\mathbf{y}) \end{aligned} \tag{3.25}$$

□

The variable to check update messages

In the SPA, messages are iteratively exchanged from bits to checks and from checks to bits. The decoding is performed locally. We can draw a graph from a variable node perspective as Fig. 3.4 shows. We have a set of incoming messages, in the first instantiation of the algorirthm they take the form of probabilities on the bit value. These messages arrive from the neighboring check nodes or from an observation in the channel. Using the sum product formula from Eq.3.15 we get the a posteriori probabilities of bit i taking value 0 and 1:

$$\begin{aligned} p_i^0 &= p(\mathbf{C}_i = 0|\mathbf{y}) \\ &= \frac{p(\mathbf{c})}{p(\mathbf{y})} \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = 0}} \prod_{j=1}^{n} p(y_j|\mathbf{C}_j = c_j) \\ &= \alpha \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = 0}} \prod_{j=1}^{n} p(y_j|\mathbf{C}_j = c_j) \end{aligned} \tag{3.26}$$

$$\begin{aligned} p_i^1 &= p(\mathbf{C}_i = 1|\mathbf{y}) \\ &= \frac{p(\mathbf{c})}{p(\mathbf{y})} \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = 1}} \prod_{j=1}^{n} p(y_j|\mathbf{C}_j = c_j) \\ &= \alpha \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i = 1}} \prod_{j=1}^{n} p(y_j|\mathbf{C}_j = c_j) \end{aligned} \tag{3.27}$$

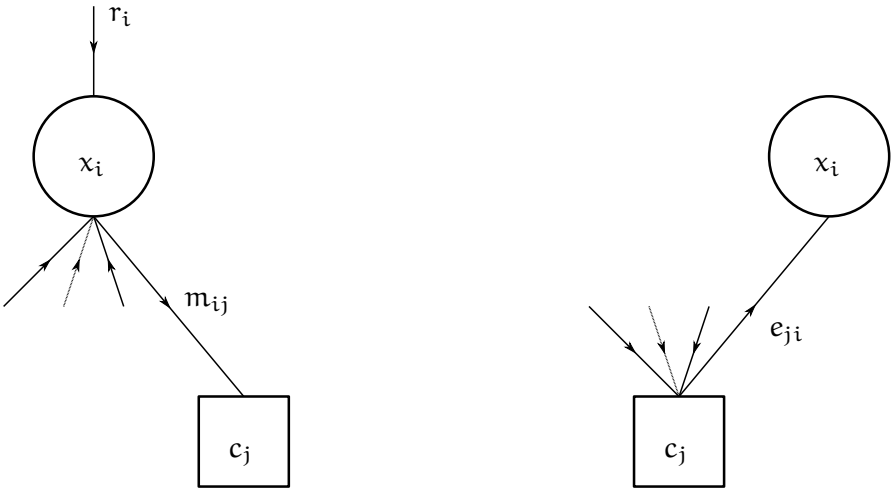

Figure 3.4: The diagram on the left shows the messages from variable nodes to check nodes. The diagram on the right shows the messages from check nodes to variable nodes.

where we can avoid explicitly computing $p(\mathbf{c})/p(\mathbf{y})$ by choosing α such that:

$$p_i^0 + p_i^1 = 1 \tag{3.28}$$

An alternative description of the variable node is as a repetition code [57]. The only possible correct configuration for a variable node is that all of the incoming messages agree on the value of bit i, either 0 or 1. Thus the variable node can be regarded as a code with just two codewords the all zero codeword and the all one codeword, that is, the repetition code. We can rewrite Eq. 3.26 taking this into account:

$$\begin{aligned} p_i^0 &= \alpha \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i=0}} \prod_{j=1}^{n} p(y_j|\mathbf{C_j} = c_j) \\ &= \alpha \prod_{j=1}^{n} p(y_j|\mathbf{C_j} = 0) \end{aligned} \tag{3.29}$$

$$\begin{aligned} p_i^1 &= \alpha \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{C}_i=1}} \prod_{j=1}^{n} p(y_j|\mathbf{C_j} = c_j) \\ &= \alpha \prod_{j=1}^{n} p(y_j|\mathbf{C}_j = 1) \end{aligned} \tag{3.30}$$

This relationship can be further simplified if we change the messages exchanged between checks and nodes and we allow them to exchange LLR values. The associated LLR with posteriori probabilities of bit i can be written as:

$$
\begin{aligned}
l_i &= \log \frac{p_i^0}{p_i^1} \\
&= \log \frac{\alpha \prod_{j=1}^{n} p(y_j|\mathbf{C_j}=0)}{\alpha \prod_{j=1}^{n} p(y_j|\mathbf{C_j}=1)} \\
&= \log \prod_{j=1}^{n} \frac{p(y_j|\mathbf{C_j}=0)}{p(y_j|\mathbf{C_j}=1)} \\
&= \sum_{j=1}^{n} \log \frac{p(y_j|\mathbf{C_j}=0)}{p(y_j|\mathbf{C_j}=1)} \\
&= \sum_{j=1}^{n} e_{ji} \qquad (3.31)
\end{aligned}
$$

where e_{ji} is the incoming LLR value sent from the neighboring check node j to bit i (see Fig. 3.4). The main benefit of the LLR representation is that instead of multiplying probabilities we can add LLR values. We can rewrite 3.31 to match the tanner graph description, let r be the LLR of the channel input and e_{ji} for $j \in \{1, dv\}$ the LLR associated with check j. We can describe the total LLR and the LLR sent to check j as:

$$l_i = r + \sum_{j=1}^{dv} e_{ji} \qquad (3.32)$$

$$m_{ij} = r + \sum_{\substack{j'=1 \\ j' \neq j}}^{dv} e_{j'i} \qquad (3.33)$$

The check to variable update messages

The update messages are sent from checks to their neighbouring bits. They compute, independently of the message received from the bit involved, the probability $p_{j,i}^{ext}$ that the parity check equation j is verified if bit i takes the value 1. Since they transport extrinsic information, they are sometimes called extrinsic messages. The probability of the equation being verified if bit i takes the value 1 is the same as the probability that an odd number of the remaining bits take the value 1. We prove by induction in the next lemma that $p_{j,i}^{ext}$ follows:

$$p_{j,i}^{ext} = \frac{1}{2} - \frac{1}{2} \prod_{\substack{t=1 \\ t \neq i}}^{dc} (1 - 2p_t) \qquad (3.34)$$

where dc is the degree of the check j.

Lemma 9. *Let* p_i *for* $i \in [1, w]$ *be the probability that the bit* i *in a block of size* w *is* 1*. The probability of having an odd number of ones is given by:*

$$p_{odd} = \frac{1 - \prod_{i=1}^{w}(1 - 2p_i)}{2}$$

Proof. The equality holds for $w = 1$:

$$p_{odd} = \frac{1 - (1 - 2p_1)^1}{2} = p_1 \tag{3.35}$$

Let us assume that it holds for $w = n$. The probability of having an odd number of errors in a block of size $n + q$ is the probability of the union of two events: having an odd number of errors in the first n bits and no error in $n + 1$ and having an even number of errors in the first n bits and an error in the bit $n + 1$. We can check that the equality is also verified for $w = n + 1$:

$$\begin{aligned} p_{odd} &= \frac{1 - \prod_{i=1}^{n}(1 - 2p_i)}{2}(1 - p_{n+1}) \\ &\quad + (1 - \frac{1 - \prod_{i=1}^{n}(1 - 2p_i)}{2})p_{n+1} \\ &= \frac{1 - \prod_{i=1}^{n+1}(1 - 2p_i)}{2} \end{aligned} \tag{3.36}$$

□

Again, the LLR representation offers an advantage with respect to the exchange of probabilistic messages:

$$\begin{aligned} e_{ji} &= \log \frac{1 + \prod_{i=1}^{dc-1}(1 - 2p_i)}{1 - \prod_{i=1}^{dc-1}(1 - 2p_i)} \\ &= \{\text{directly from Lem. 10}\} \\ &= \log \frac{1 + \prod_{i=1}^{dc-1} \tanh \frac{m_i}{2}}{1 - \prod_{i=1}^{dc-1} \tanh \frac{m_i}{2}} \\ &= \{\text{directly from Lem. 11}\} \\ &= 2\tanh^{-1} \prod_{i=1}^{dc-1} \tanh \frac{m_i}{2} \end{aligned} \tag{3.37}$$

where we have used two relationships that we proceed to prove.

Lemma 10.

$$\tanh\left(\frac{1}{2} \log \frac{1-p}{p}\right) = 1 - 2p$$

Proof.

$$\begin{aligned}\tanh\left(\frac{1}{2}\log\frac{1-p}{p}\right) &= \frac{e^{-m/2}-e^{m/2}}{e^{-m/2}+e^{m/2}}\\ &= \frac{1-e^{m}}{1+e^{m}}\\ &= 1-2\frac{e^{m}}{1+e^{m}}\\ &= 1-2p \end{aligned} \tag{3.38}$$

where m is the LLR associated with p and we have expanded the hyperbolic tangent in the first equality and used Eq. 3.24 in the last equality. □

Lemma 11.

$$\tanh\left(\frac{1}{2}\log\frac{1+p}{1-p}\right) = p$$

Proof.

$$\begin{aligned}\tanh\left(\frac{1}{2}\log\frac{1+p}{1-p}\right) &= \frac{e^{t/2}-e^{-t/2}}{e^{t/2}+e^{-t/2}}\\ &= \frac{e^{t}-1}{e^{t}+1}\\ &= \frac{\frac{1+p}{1-p}-1}{\frac{1+p}{1-p}+1}\\ &= p \end{aligned} \tag{3.39}$$

where t is defined as:

$$t = \log\frac{1+p}{1-p} \tag{3.40}$$

□

We can further simplify the check to bit update message from Eq. 3.37 with the following map [103]:

$$\begin{aligned}\gamma &: [-\infty,\infty] \to \mathbb{F}_2 \times [0,\infty]\\ \gamma(x) &= (\operatorname{sgn}(x), -\log\tanh\frac{|x|}{2})\end{aligned} \tag{3.41}$$

with the special case of $-\log(0) := +\infty$ and $\operatorname{sgn}(x)$ defined as:

$$\operatorname{sgn}(x) = \begin{cases} 0 & \text{if } x < 0\\ 0 & \text{with probability } 0.5 \text{ if } x = 0\\ 1 & \text{with probability } 0.5 \text{ if } x = 0\\ 1 & \text{if } x > 0 \end{cases} \tag{3.42}$$

the function $\operatorname{sgn}(x)$ has a particular interpretation, if x is an LLR associated with a bit, the output of $\operatorname{sgn}(x)$ is the hard decision on the bit. Now taking into account that the tanh is an odd function:

$$\begin{aligned} e_{ji} &= 2\tanh^{-1}\prod_{i=1}^{dc-1}\tanh\frac{m_i}{2} \\ &= \left(\prod_{i=1}^{dc-1}\operatorname{sgn}(m_i)\right)2\tanh^{-1}\prod_{i=1}^{dc-1}\tanh\frac{|m_i|}{2} \\ &= \left(\prod_{i=1}^{dc-1}\operatorname{sgn}(m_i)\right)2\tanh^{-1}\log^{-1}\log\prod_{i=1}^{dc-1}\tanh\frac{|m_i|}{2} \\ &= \left(\prod_{i=1}^{dc-1}\operatorname{sgn}(m_i)\right)2\tanh^{-1}\log^{-1}\sum_{i=1}^{dc-1}\log\tanh\frac{|m_i|}{2} \\ &= \gamma^{-1}\left(\sum_{i=1}^{dc-1}\gamma(m_i)\right) \end{aligned} \tag{3.43}$$

3.2.2.1 *Output*

The sum-product decoder continues the iteration process from variable to check nodes and from check to variable nodes until a limit of iterations is reached or a valid codeword is found.

3.2.3 *Density Evolution*

No good technique is known to study the behavior of specific instances of LDPC codes so instead we are going to focus on ensembles of codes defined by their variable and check edge distributions ($\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$). These distributions are closely related to the node distributions $\lambda'(x)$ and $\rho'(x)$ that we previously described and it is possible to change from one to another description following:

$$\lambda(x) = \frac{1}{\sum_i i\lambda'_i}\sum_i i\lambda'_i x^{i-1} \tag{3.44}$$

$$\rho(x) = \frac{1}{\sum_i i\rho'_i}\sum_i i\rho'_i x^{i-1} \tag{3.45}$$

using these relations, we can express the code rate from Eq. 3.22 as a function of the coefficients of $\lambda(x)$ and $\rho(x)$:

$$\text{Rate} = 1 - \frac{\sum_i \rho_i/i}{\sum_i \lambda_i/i} \tag{3.46}$$

In this section we study the evolution of messages as variables and checks exchange them over iterations. We are going to track the evolution of messages regarding the probability density functions of variable and check nodes (f_v and f_c) and their related cumulative functions.

We assume that all the messages are independent, which implies that the channel has no memory and that there are no cycles of length $2t$ in the code graph. Regarding the channel, we require it to be output symmetric. Under this condition the probability of error of all codewords is the same [101]. We choose to track the all zeros codeword, which implies that the tracked density is the LLR assuming tha every variable node takes the value 0.

In particular for any message on iteration t:

$$F_c^t(z) = P[\mathbf{E_{ji}^t} \leqslant z] = \int_{-\infty}^{z} f_c^t(x)dx \tag{3.47}$$

$$F_v^t(z) = P[\mathbf{M_{ij}^t} \leqslant z] = \int_{-\infty}^{z} f_v^t(x)dx \tag{3.48}$$

we will drop the superscript t wherever there is no ambiguity. The probability of error is just the probability that $m_{ij}^t < 0$ averaged over all nodes. We could write it as the integral from $-\infty$ to 0 of m_{ij}^t but we should include only half of the mass at 0. The following definition allows for a more compact representation of the probability of error:

$$P_{err}(f_v^t) = \int_{-\infty}^{\infty} f_c^v(x)e^{-\frac{|x|+x}{2}}dx \tag{3.49}$$

Let us examine when on iteration t variable i receives dv messages and outputs to check j $m_{ij} = r + \sum_{\substack{j'=1 \\ j'\neq j}}^{dv} e_{j'}$ as we derived in Eq. 3.33.

Now, we have that the sum of two independent random continuous variables $\mathbf{X}$ and $\mathbf{Y}$ defined over an additive group G is also a random variable $\mathbf{V}$ [110], and the density function of $\mathbf{V}$ is the convolution of the addend density functions.

We can rewrite the variable update equation for densities as:

$$f_{v_i} = f_r \otimes f_c^{\otimes dv-1} \tag{3.50}$$

where $f_c^{\otimes dv-1}$ denotes the convolution of f_c with itself $dv-1$ times. f_r takes the place of r in Eq. 3.33 and stands for the density distribution associated with the channel output.

Example 2. *Let us consider the BEC, let* $l(y) = \log(p(y|\mathbf{X}=0)/p(y|\mathbf{X}=1))$*, then we have:*

$$l(y) = \begin{cases} l(0) = \log\frac{1-p}{0} = \infty \\ l(1) = \log\frac{0}{1-p} = -\infty \\ l(\varepsilon) = \log\frac{p}{p} = 0 \end{cases}$$

Let $\delta_z(x)$ *be the Dirac delta of density one at point* z*. We can write the density of the output of the BEC as:*

$$f_{r_{BEC}}(p) = p\delta_0(x) + (1-p)\delta_\infty(x) \tag{3.51}$$

We can derive the density of the BSC in the same way:

$$f_{r_{BSC}}(p) = p\delta_{\log\frac{p}{1-p}}(x) + (1-p)\delta_{\log\frac{1-p}{p}}(x) \tag{3.52}$$

If we average f_{v_i} taking into account the edge degree distribution we obtain the general update rule:

$$f_v = f_r \otimes \sum_i \lambda_i f_c^{\otimes i-1} \tag{3.53}$$

The check node density function is a little more complicated. We have a relation that is a product of random variables in Eq. 3.37 and we have a sum of random variables transformed by γ in Eq. 3.43. There is no simple way of computing the product of random variables. However it is possible to describe the density of the random variables transformed by γ [102]. We write the distribution function of random variable $\gamma(\mathbf{Z})$ $\Gamma(F_Z)$ the transformation of the random variable $\mathbf{Z}$ by γ as a function $\mathbb{F}_2 \times [0,\infty) \to [0,1]$:

$$\Gamma(F_z)(s,x) = \chi_{\{s=0\}}\Gamma_0(F_Z)(x) + \chi_{\{s=1\}}\Gamma_1(F_Z)(x) \tag{3.54}$$

where $\chi_{\{\}}$ takes the value 1 if the condition under the brackets is verified and takes value 0 otherwise. Recalling Eq. 3.42 if $z > 0$ we have $\Gamma(F_z)(s,x) = \Gamma_0(F_Z)(x)$ and if $z < 0$ $\Gamma(F_z)(s,x) = \Gamma_1(F_Z)(x)$. We define the pseudo distributions $\Gamma_0(F_Z)(x)$ and $\Gamma_1(F_Z)(x)$ as:

$$\begin{aligned} \Gamma_0(F_Z)(x) &= P[\gamma_1(\mathbf{Z}) = 0, \gamma_2(\mathbf{Z}) \leqslant x] \\ &= 1 - P[\gamma_2(\mathbf{Z}) > x] \\ &= 1 - P[-\log\tanh(\frac{\mathbf{Z}}{2}) > x] \\ &= 1 - P[\mathbf{Z} < -\log\tanh(\frac{x}{2})] \\ &= 1 - F_Z(-\log\tanh(\frac{x}{2})) \end{aligned} \tag{3.55}$$

$$\begin{aligned}\Gamma_1(F_Z)(x) &= P[\gamma_1(\mathbf{Z}) = 1, \gamma_2(\mathbf{Z}) \leqslant x] \\ &= P[-\log\tanh(\frac{-\mathbf{Z}}{2}) > x] \\ &= F_Z(\log\tanh(\frac{x}{2})) \end{aligned} \tag{3.56}$$

and we can verify that Γ^{-1} performs the inverse map:

$$\begin{aligned}\Gamma^{-1}(x) &= \chi_{\{x\geqslant 0\}}\Gamma_0(-\log\tanh(\frac{x}{2})) \\ &\quad +\chi_{\{x<0\}}\Gamma_1(-\log\tanh(\frac{-x}{2})) \end{aligned} \tag{3.57}$$

We call the distributions transformed by Γ G-distributions and the associated densities g-densities [102]. The density of $\Gamma(F_z)(s,x)$ is defined as usual:

$$\begin{aligned}\frac{d}{dx}\Gamma(F_z)(s,x) &= \chi_{\{s=0\}}\frac{f_z(-\log\tanh(\frac{x}{2}))}{\sinh x} \\ &\quad +\chi_{\{s=1\}}\frac{f_z(\log\tanh(\frac{x}{2}))}{\sinh x} \end{aligned} \tag{3.58}$$

G-distributions and g-densities have a well defined convolution over their domain, the group $\mathbb{F}_2 \times [0,\infty) \to [0,1]$. Let $g_a(s,x) = \chi_{\{s=0\}}g_a^0(x) + \chi_{\{s=1\}}g_a^1(x)$ and $g_b = \chi_{\{s=0\}}g_b^0(x) + \chi_{\{s=1\}}g_b^1(x)$ be two g-densities, their convolution is a random variable C with g-density given by:

$$\begin{aligned}g_c(s,x) &= \chi_{\{s=0\}}(g_a^0(x) * g_b^0(x) + g_a^1(x) * g_b^1(x)) \\ &\quad +\chi_{\{s=1\}}(g_a^0(x) * g_b^1(x) + g_a^1(x) * g_b^0(x)) \end{aligned} \tag{3.59}$$

which is just a two-dimensional convolution. In one dimension it is the cyclic convolution over $\mathbb{F}_2$ while in the other dimension it is the one sided convolution over the real numbers (operation represented by $*$).

We have described messages from and to the variable nodes as densities of LLRs. Then, the density of the update message from check j of degree d_c to nodes can be written as a convolution of the incoming densities transformed into g-densities and transformed back into LLR densities.

$$f_{c_j} = \Gamma^{-1}\left(\Gamma(f_v)^{\boxtimes d_c - 1}\right) \tag{3.60}$$

A variable node in the graph is connected to a check of degree i with probability ρ_i, then the average density equals:

$$f_c = \Gamma^{-1}\left(\sum_i \rho_i \Gamma(f_v)^{\boxtimes i-1}\right) \tag{3.61}$$

Combining the variable and checks densities allows us to describe the relation that tracks the density evolution of the messages:

$$f_v^{t+1} = f_r \otimes \lambda(\Gamma^{-1}(\rho(\Gamma(f_v^t)))) \tag{3.62}$$

where we have made explicit that the density at iteration $t+1$ is given by the density at iteration t.

We call the asymptotic threshold of a degree distribution the maximum level of noise for which $P_{err}(f_v^t)$ converges to 0. We have not discussed the evolution of the P_{err} on the SPA as the number of iterations increase or the channel becomes noisier, but it can be proved that the SPA is monotonic with respect to both parameters; the performance improves if we increase the iterations and also improves if the noise is reduced (see [102]). The threshold determines the limit of the error-free region as the block length tends to infinity, in practice it allows to choose between different ensembles.

In this thesis we have implemented a discretized version of the density evolution algorithm [17]. This version guarantees that the predicted threshold is a lower bound of the real threshold while offering an easy implementation. This allows us to trade precision for speed while being able to discriminate between two codes.

The Discretized Density Evolution (DDE) quantizes the LLR messages exchanged in the SPA with the quantizing operator Q defined as:

$$Q(w) = \begin{cases} \lfloor \frac{w}{\Delta} + \frac{1}{2} \rfloor & \text{if } w \geqslant \frac{\Delta}{2} \\ \lceil \frac{w}{\Delta} - \frac{1}{2} \rceil & \text{if } w \leqslant -\frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases}$$

where Δ is the quantization interval.

If the exchanged messages are quantized with $Q(w)$ Eq. 3.33 becomes:

$$\hat{m}_{ij} = \hat{r} + \sum_{\substack{j'=1 \\ j' \neq j}}^{dv} \hat{e}_{j'i} \tag{3.63}$$

where $\hat{m}$, $\hat{r}$ and $\hat{e}_{j'i}$ are quantized versions of m, r and $e_{j'i}$. If we track the density of the associated discrete random variable, the density of the sum is given by the (discrete) convolution of the addends:

$$f_{\hat{v}} = f_{\hat{r}} \otimes \sum_i \lambda_i f_{\hat{c}}^{\otimes i-1} \tag{3.64}$$

Let i be a check with $dc > 4$, we can rewrite the check update equation as follows:

$$\begin{aligned} e_{ji} &= 2\tanh^{-1}\left(\prod_{i=1}^{dc-1}\tanh\frac{m_i}{2}\right) \\ &= 2\tanh^{-1}\left(\tanh\frac{m_1}{2}\tanh\left(\frac{1}{2}2\tanh^{-1}\prod_{i=2}^{dc-1}\tanh\frac{m_i}{2}\right)\right) \\ &= \text{etc.} \end{aligned} \tag{3.65}$$

Now it is easy to verify that we can write the following quantized version of the check update formula:

$$\hat{e}_{ji} = R(\hat{m}_1, R(\hat{m}_2, R(..., \hat{m}_{dc-1}))) \tag{3.66}$$

where the operator R is defined as:

$$R(\hat{m}_1, \hat{m}_2) = Q(2\tanh^{-1}(\tanh\frac{\hat{m}_1}{2}\tanh\frac{\hat{m}_2}{2})) \tag{3.67}$$

The density of the check messages cannot be computed by convolution and we have to describe it by inspection.

$$\mathcal{R}(f_{\hat{c}_a}, f_{\hat{c}_b})[k] = \sum_{(i,j)|k\Delta=R(i\Delta,j\Delta)} f_{\hat{c}_a}[i] f_{\hat{c}_b}[j] \tag{3.68}$$

We can write a quantized version of Eq. 3.61 as:

$$f_{\hat{c}} = \sum_i \rho_i \mathcal{R}^{i-1}(f_{\hat{v}}) \tag{3.69}$$

where $\mathcal{R}^2(f_{\hat{v}}) = \mathcal{R}(f_{\hat{v}}, f_{\hat{v}})$ and $\mathcal{R}^i(f_{\hat{v}}) = \mathcal{R}(f_{\hat{v}}, \mathcal{R}^{i-1} f_{\hat{v}})$.

Combining Eq. 3.64 and Eq. 3.69 gives us the DDE update formula:

$$f_{\hat{v}}^{t+1} = \sum_j \rho_j \mathcal{R}^{j-1}\left(f_{\hat{r}} \otimes \sum_i \lambda_i f_{\hat{c}}^{\otimes i-1}\right) \tag{3.70}$$

This quantized version of density evolution is extremely convenient. In this version, the convolution at the variable nodes can be efficiently implemented using the Fourier transform. The R operation also has a fast realization in the form of a look up table.

3.3 OPTIMIZATION OF LDPC CODE DISTRIBUTIONS

DDE is a good technique to compute the threshold of a given family of LDPC codes. The threshold can be regarded as a figure of merit and it can be used to compare families of codes. However DDE does not help

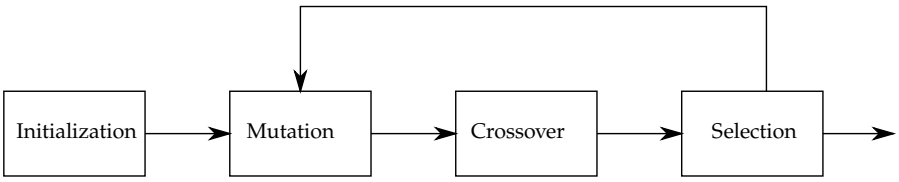

Figure 3.5: Activity diagram of DiffE

in finding a specific family. Finding codes with high thresholds implies searching through the space of possible degree distributions. This is equivalent to finding a maximum value in a real valued non-linear function with real valued parameters, this is a non trivial task.

There have been several proposals for designing LDPC codes. In [120] codes are optimized by curve fitting on extrinsic information transfer charts [119], which provides an approximation to the threshold. In [69] the solution space is highly reduced by optimizing only $\lambda(x)$, with this simplification it is possible to use tools like linear programming.

Another tool that does not simplify neither the threshold computation neither the solution space are non-linear optimization heuristics. In particular Differential Evolution (DiffE) [116], a genetic algorithm, has been used for designing LDPC codes. This solution was successfully applied for the BEC in [111] and for the Additive White Gaussian Noise (AWGN) channel in [103].

3.3.1 *Differential Evolution*

DiffE is a stochastic real parameter optimization algorithm. It was first proposed in 1995 by Storn and Price [115, 23] and draw rapidly attention after getting top positions in several evolutionary optimization contests [23]. The algorithm is, as implied, an evolutionary algorithm which basically means that it works with a population of solutions that evolves through iterations in a random though directed fashion.

The algorithm is used to optimize functions with real parameters and real values; given an objective function $f : X \subseteq \mathbb{R}^D \to \mathbb{R}$ searches a solution $\hat{x}$ such that $f(\hat{x}) \leqslant f(x), \forall x \in X$. Fig. 3.5 shows the work-flow of the algorithm. DiffE works with a population of D-dimensional vectors or *chromosomes*:

$$\mathbf{x_{i,G}} = [x_{1,i,G}, x_{2,i,G}, ..., x_{D,i,G}] \tag{3.71}$$

where G indicates the generation number. The population of the first generation is created randomly in the initialization step. In every generation the chromosomes are perturbed with scaled differences of the vectors producing *donors* $\mathbf{v_{i,G}}$. This process is called mutation. Every vector from the current generation or *target* incorporates a random set of parameters from a donor to produce a *trial* vector $\mathbf{u_{i,G}}$. The mixing

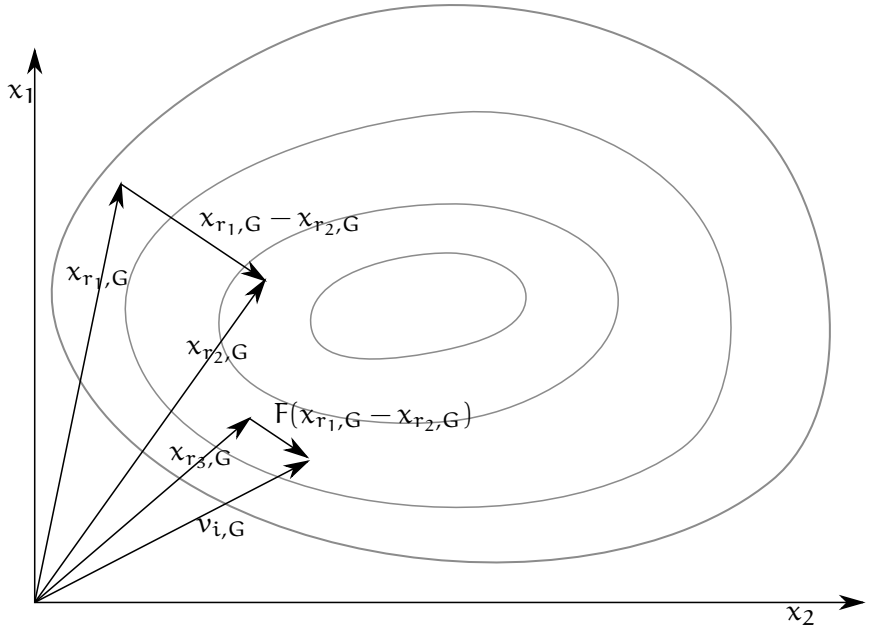

Figure 3.6: Constructing donor vectors with mutation

process is called crossover or recombination. Trial vectors are selected to replace the targets if $f(\mathbf{u_{i,G}}) \leqslant f(\mathbf{x_{i,G}})$. The process is repeated until a limit of iterations is reached or some acceptable candidate is found. We proceed to describe in more detail the algorithm.

In the initialization step N D-dimensional vectors are created. The D parameters are real but they are allowed to have a minimum and a maximum value; we can define $\mathbf{x_{min}} = [x_{min,1}, x_{min,2}, ..., x_{min,D}]$ and $\mathbf{x_{max}} = [x_{max,1}, x_{max,2}, ..., x_{max,D}]$ two vectors holding the bounds for all parameters. In the most general situation there is no knowledge on the solution, in consequence the initial population should cover the solution space as uniformly as possible, let $rand(a, b)$ be a function with uniformly random output in the interval $[a, b]$:

$$x_{i,G,j} = rand(x_{min,j}, x_{max,j}) \tag{3.72}$$

Mutation is performed by adding the weighted difference of two population vectors $\mathbf{x_{r_1,G}}$ and $\mathbf{x_{r_2,G}}$ to a third one $\mathbf{x_{r_3,G}}$. r_1, r_2 and r_3 should be different from each other and also different to i which limits the minimum number of parameters to 4. The donor vectors are constructed as follows:

$$\mathbf{v_{i,G}} = \mathbf{x_{r_3,G}} + F \cdot (\mathbf{x_{r_1,G}} - \mathbf{x_{r_2,G}}) \tag{3.73}$$

where the scale factor F is used to control how quickly the population evolves, typical values are in the range $[0.8, 1]$. Fig. 3.6 shows the effect of the mutation step.

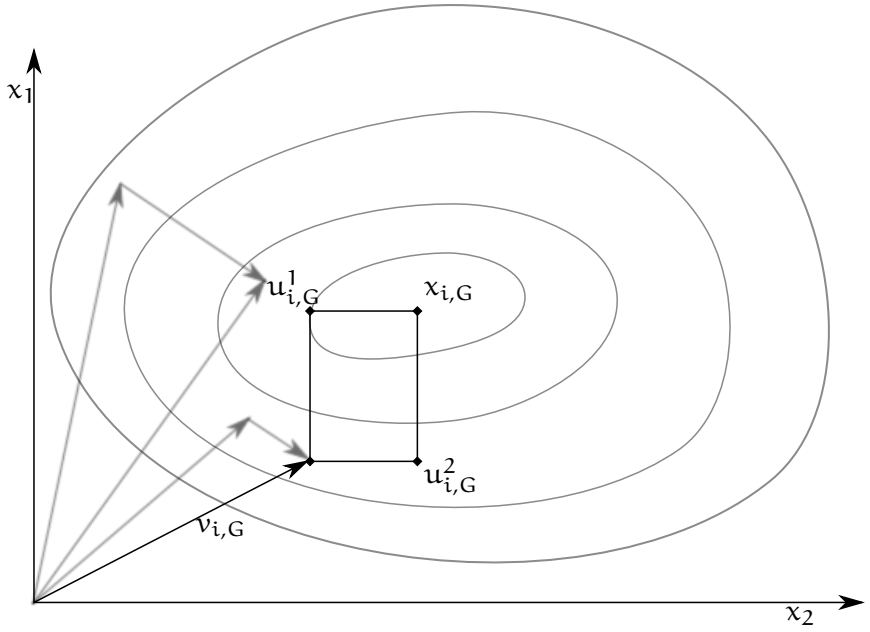

Figure 3.7: Recombination of the donor with the target.

Recombination also known as crossover is used to increase the diversity of the trial population: donor vectors are modified incorporating a small set of parameter values from the target vector. This avoids limiting the search to linear combinations of the current population:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand(0,1) \leqslant CR \\ x_{j,i,G} & \text{if } rand(0,1) > CR \end{cases} \tag{3.74}$$

where the crossover ratio CR is a second control parameter. Fig. 3.6 shows the effect of the crossover step.

A trial vector replaces its target if $f(\mathbf{u}_{i,G}) \leqslant f(\mathbf{x}_{i,G})$, otherwise it is discarded.

3.3.2 *Design of LDPC codes*

The functions $\lambda(x)$ and $\rho(x)$ have $\delta_v + \delta_c - 2$ non zero coefficients. However not all these coefficients are independent: $\lambda(x)$ and $\rho(x)$ define degree distributions and must therefore be normalized, additionally we want all codes to be of the same rate in order to compare their thresholds.

In particular, to ensure that $\lambda(x)$ and $\rho(x)$ define a degree distribution we fix the coefficients corresponding to variable and check nodes of degree 2:

$$\lambda_2 = 1 - \sum_{i=3}^{\delta_v} \lambda_i \tag{3.75}$$

$$\rho_2 = 1 - \sum_{i=3}^{\delta_c} \rho_i \tag{3.76}$$

We can set the code rate using a third coefficient, we use λ_{δ_c}. From (Eq. 3.46) and (Eq. 3.75), one gets:

$$\lambda_{\delta_c} = \frac{\frac{1-\beta}{2} + \sum_{i=3}^{\delta_c} \rho_i (\frac{1}{i} - \frac{1}{2}) - \beta \sum_{i=3}^{\delta_v - 1} \lambda_i (\frac{1}{i} - \frac{1}{2})}{\beta(\frac{1}{\delta_v} - \frac{1}{2})} \tag{3.77}$$

where $\beta = 1 - \text{Rate}$.

These three constraints leave a final number of $D = \delta_v + \delta_c - 5$ parameters each one associated with one of the non fixed coefficients of $\lambda(x)$ and $\rho(x)$. Finally we require the codes to be stable for every crossover probability ε below their threshold [103], the stability condition particularized for the BSC is:

$$\lambda_2 \leqslant \frac{1}{2\sum_i (i-1)\rho_i \sqrt{\varepsilon(1-\varepsilon)}} \tag{3.78}$$

We define an initial population of N vectors of $D = \delta_v + \delta_c - 5$ parameters each one associated with one of the non fixed coefficients of $\lambda(x)$ and $\rho(x)$. The initial population is not taken completely randomly as this would lead to very complex codes with few zero coefficients. We instead allow only a small random amount of coefficients to be non zero.

We have found that the process of finding good codes could be speeded up if a large initial number of codes was taken. The initial random codes were poor and improved slowly with the number of generations. Taking a population size $N_1 = 2000$ for the first generation, and quickly reducing it: $N_2 = 200$ for the second generation and $N_{>3} = 20$ from the third generation, lead to better results.

3.3.3 *Codes*

The results we have obtained with this set of constraints are shown in Table 3.2. For all rates the thresholds are very close to the Shannon limit.

3.4 RATE MODULATION

3.4.1 *Introduction*

Let us consider two common techniques used to manipulate the information rate of a code: shortening and puncturing. Shortening

Table 3.2: Thresholds and degree distributions found for a representative set of rates.

Code rate	Threshold	$\lambda(x)$ & $\rho(x)$
0.70	0.0510	$\lambda(x) = 0.1146x + 0.1440x^2 + 0.0536x^3$
		$+0.0360x^4 + 0.0700x^6 + 0.1128x^7$
		$+0.0558x^8 + 0.4132x^{29}$
		$\rho(x) = 0.39210x^{18} + 0.59116x^{19} + 0.01674x^{20}$
0.65	0.0633	$\lambda(x) = 0.1162x + 0.2046x^2 + 0.0188x^3$
		$+0.0215x^4 + 0.0462x^6 + 0.0552x^7$
		$+0.0873x^8 + 0.0710x^9 + 0.0286x^{10}$
		$+0.3506x^{29}$
		$\rho(x) = 0.46020x^{13} + 0.03061x^{16} + 0.50919x^{17}$
0.60	0.0766	$\lambda(x) = 0.11040x + 0.20804x^2 + 0.14163x^7$
		$+0.14858x^8 + 0.14438x^{25} + 0.08909x^{26}$
		$+0.00748x^{45} + 0.15038x^{70}$
		$\rho(x) = 0.00036x + 0.13063x^9 + 0.31068x^{12}$
		$+0.49341x^{17} + 0.064915x^{18}$
0.55	0.0904	$\lambda(x) = 0.1524x + 0.1938x^2 + 0.0676x^3$
		$+0.0195x^4 + 0.0518x^6 + 0.0552x^7$
		$+0.0846x^8 + 0.561x^{10} + 0.0648x^{23}$
		$+0.2542x^{29}$
		$\rho(x) = 0.98355x^{10} + 0.00452x^{11}$
		$+0.01193x^{12}$
0.50	0.1071	$\lambda(x) = 0.14438x + 0.19026x^2 + 0.01836x^3$
		$+0.00233x^4 + 0.04697x^5 + 0.053943x^7$
		$+0.05590x^8 + 0.01290x^9 + 0.00162x^{10}$
		$+0.06159x^{13} + 0.13115x^{14} + 0.01481x^{16}$
		$+0.00879x^{46} + 0.00650x^{48} + 0.00210x^{54}$
		$+0.00099x^{55} + 0.11178x^{56} + 0.06238x^{57}$
		$+0.05094x^{58} + 0.02230x^{65}$
		$\rho(x) = 0.47575x^9 + 0.46847x^{11} + 0.02952x^{12}$
		$+0.02626x^{13}$

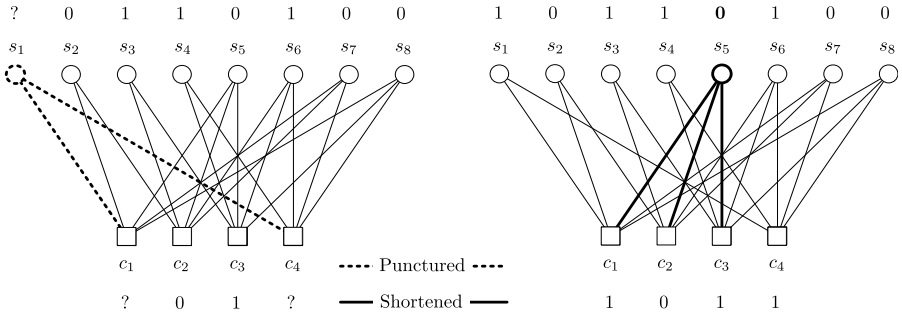

Figure 3.8: Examples of puncturing and shortening strategies applied to a linear code represented by its Tanner graph. In the puncturing example (left) one symbol is deleted from the word and a $\mathcal{C}(8,4)$ code, with rate $R = 1/2$, is converted to a $\mathcal{C}(7,4)$, increasing its rate to $R = 4/7$. In the shortening example (right), one symbol is deleted from the encoding and the same $\mathcal{C}(8,4)$ code is converted to a $\mathcal{C}(7,3)$ code, the rate now decreases to $R = 3/7$.

is a technique used to reduce the coding rate. The codewords that have a 0 in s fixed positions are kept, the rest are discarded, only half of the codewords have a zero at any position. Then the codewords are shrinked by deleting those s positions. The number of codewords is reduced to 2^{k-s} while the word space is also reduced to 2^{n-s}. In consequence an $\mathcal{C}(n,k)$ code is converted into a $\mathcal{C}(n-s,k-s)$ code.

Puncturing is a technique used to increase the coding rate by reducing the codeword length. A set of p bits in fixed positions are deleted from all codewords, i.e. the number of codewords remains unchanged but the space of words is reduced converting a $\mathcal{C}(n,k)$ into a $\mathcal{C}(n-p,k)$ code (see Refs. [47, 91]). A graphical representation, on a Tanner graph, of the procedures just described for puncturing and shortening and its effects on the rate of the sample code is shown in Fig. 3.8.

3.4.2 *Puncturing*

We continue the discussion analyzing a specific puncturing technique. A linear code $\mathcal{C}$ is punctured by deleting a defined set of symbol nodes with positions known both to the encoder and decoder. Therefore, the punctured symbols allow to modulate the relation between the codeword length and the length of the information symbols. In the most general setting, if we consider maximum likelihood decoding, capacity achieving codes can be constructed through puncturing [53].

In order to puncture finite-length codes we differentiate random puncturing and *intentional* puncturing. In the former, symbol nodes to be punctured are randomly chosen, while in the latter it is defined an ordered set of puncturable symbols. The asymptotic performance of random and intentional punctured LDPC codes is studied in [47], and puncturing thresholds have been identified in [91]. Some other

methods delve in the code structure to identify puncturing patterns [48, 123, 31], or examine the graph construction for short-length codes [132, 59].

3.4.3 *Local Intentional Puncturing*

The set of symbols to puncture can be chosen in a random fashion or conforming an established procedure. As commented in the previous section we call them random and intentional puncturing respectively. We can further differentiate intentional puncturing methods between: methods that optimize the asymptotic behavior of families of punctured LDPC codes, and finite length methods that focus in minimizing the impact of puncturing in the decoding of finite length codes. In this section we introduce some notation and describe a finite length method.

3.4.3.1 *Basic Notation and Previous Definitions*

Let $\mathcal{N}(z_j)$ denote the set of symbol nodes adjacent to the check node z_j, such that $\mathcal{N}(z_j) = \{x_k : H_{j,k} = 1, 1 \leqslant k \leqslant n\}$ is the set of symbol nodes that participates in the parity-check equation H_j, and let $\mathcal{M}(x_k)$ be the corresponding set of check nodes adjacent to the symbol node x_k, $\mathcal{M}(x_k) = \{z_j : H_{j,k} = 1, 1 \leqslant j \leqslant m\}$.

Definition 1: Two symbol nodes are said to be *neighbors* if both are directly connected through a common check node, and thus they participate in the same parity-check equation H_j. Graphically it is depicted by a 2-length path consisting of two edges joined by a common check node.

The neighboring set of a symbol node x_k is then given by $\mathcal{G}(x_k) = \{x_i : x_i \in \mathcal{N}(z_j), z_j \in \mathcal{M}(x_k)\}$. Fig. 3.9 shows an example of this concept.

In [48] it is defined the concept of *one-step recoverable* (1-SR) for a symbol node when there is at least one survived node within the set of adjacent check nodes and can, in consequence, be recovered in one step. A check node is considered survived if there are no punctured nodes within the set of adjacent symbol nodes. The definition can be extended to consider nodes that can be recovered in k steps or *k-step recoverable* (k-SR) symbol nodes. The recovery tree of a punctured node is defined as the graph spanning from a punctured note through a survived check node and unfolding the symbols for every check and the checks for every punctured symbols until all ramifications end in an unpunctured symbol. The recovery error probability $P_e(v)$ is the probability that a punctured symbol v is recovered with the wrong message from a survived check. $P_e(v)$ is shown to be an increasing function of the number of symbols in the recovery tree of v (see [48]) for several channels of interest. We introduce the concept of *one-step untainted*, based on a similar definition of 1-SR, to propose a simple

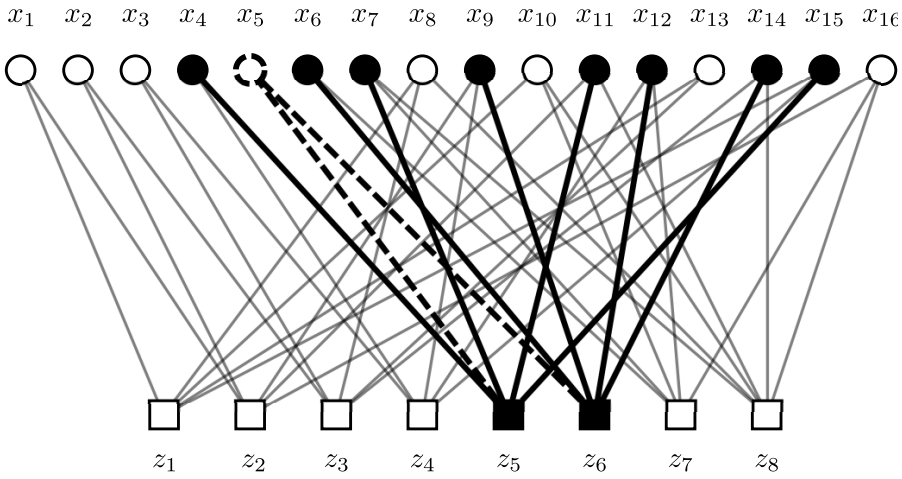

Figure 3.9: In this example, x_5 is a symbol node selected to be punctured, and $\{x_4, x_6, x_7, x_9, x_{11}, x_{12}, x_{14}, x_{15}\}$ is the neighboring set of symbol nodes that will be excluded in following selections. The neighboring set is computed from the set of check nodes adjacent to the selected symbol, $\{z_5, z_6\}$ in the current example.Note that the algorithm could have selected as the first symbol to puncture equiprobably any symbol in the set $\{x_1, x_2, x_3, x_4, x_5, x_6, x_9, x_{12}, x_{13}, x_{15}\}$.

finite length method that chooses symbols such that all the check nodes of a selected symbol are survived nodes.

Definition 3: A symbol node x_k is said to be *one-step untainted* (1-SU) if there are no punctured symbols within its neighboring set $\mathcal{G}(x_k)$.

3.4.3.2 *Proposed Algorithm*

Let $\mathcal{X}_\infty$ be a set of symbol nodes that are not affected by the puncturing of a neighboring symbol, i.e. it is the ensemble including every 1-SU symbol node. And let $\mathcal{Z}_\infty$ be the set containing every check node which is not adjacent to any punctured symbol. Initially, when there are not punctured symbols, $\mathcal{X}_\infty$ and $\mathcal{Z}_\infty$ consist of every symbol and check node, respectively.

Let p be the number of symbols to be punctured, the proposed algorithm is described in Alg. 1.

The algorithm concludes when it chooses p^* symbols to puncture, and thus it obtains the set $\mathcal{P} = \{x_{n_1}, x_{n_2}, ..., x_{n_p}\}$ consisting of the symbol nodes selected in the third step, and $n_1, n_2, ..., n_p \in [1, n]$ the list of symbol indexes to be punctured.

Notice that whenever the check node distribution is regular the selection criterion that the algorithm uses may be simplified. Instead of a symbol node with the smallest neighboring set it can select a symbol x_k with the lowest check node degree $\mathcal{N}(x_k)$.

We have simulated the behavior of punctured codes over the BSC using LDPC codes of 10^4 bits length and two coding rates: $R = 0.5$

Rate	0.50	0.60
λ_2	0.15967	0.11653
λ_3	0.12187	0.12565
λ_4	0.11261	0.10851
λ_5	0.19087	0.05342
λ_7	—	0.07272
λ_8	—	0.03480
λ_9	—	0.07300
λ_{10}	0.07706	—
λ_{11}	—	—
λ_{14}	—	—
λ_{15}	—	—
λ_{18}	—	0.07526
λ_{25}	0.33791	—
λ_{32}	—	0.11710
λ_{45}	—	0.22301
ε_{th}	0.102592	0.0745261
$p^*_{min}{}^a$	3444	2877
$p^*_{max}{}^a$	3551	2978
$p^*_{min}{}^b$	1916	1585
$p^*_{max}{}^b$	1986	1643

Table 3.3: Generating Polynomials. ^{a}Algorithm proposed in [48]. ^{b}Algorithm proposed here.

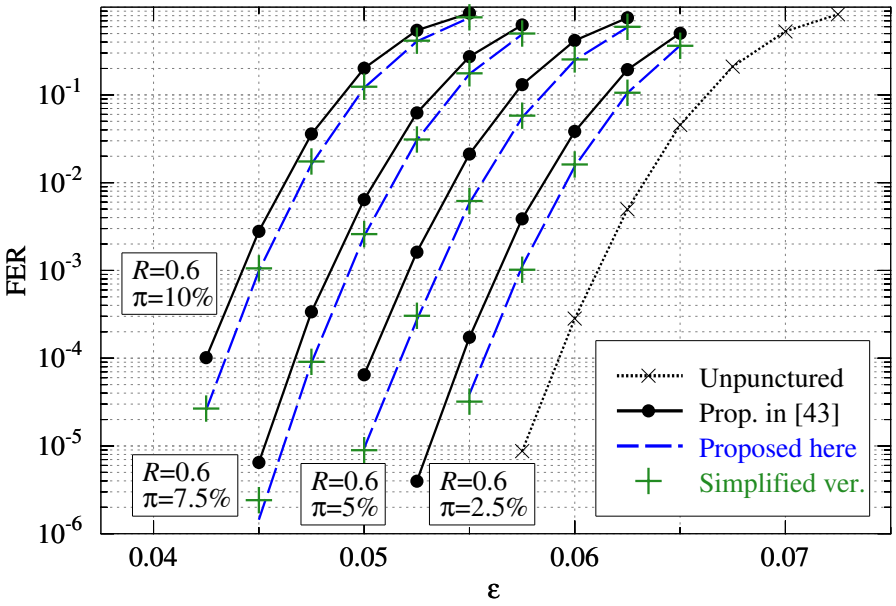

Figure 3.10: FER over the BSC with crossover probability ε for different intentional puncturing strategies. It was used an LDPC code with coding rate $R = 0.6$, and four different proportions of punctured symbols: $\pi = 2.5\%$, $\pi = 5\%$, $\pi = 7.5\%$, and $\pi = 10\%$.

and $R = 0.6$. These codes were constructed using the Progressive Edge Growth (PEG) algorithm as proposed in [54]. The results were computed under iterative decoding, using a sum-product algorithm with serial scheduling and a maximum of 200 iterations.

In order to compare the algorithms we define p^*_{min} and p^*_{max} as the minimum and maximum sizes of p^* over all simulations. In Table 3.3 it can be observed that both, p^*_{min} and p^*_{max}, are smaller in the proposed algorithm than in [48], i.e. this algorithm allows for a smaller number of punctured symbols which also implies a reduction in the achievable rate through puncturing.

Fig. 3.11 shows the FER over the BSC for different intentional puncturing strategies. The use of puncturing patterns as proposed in [48] is compared with the algorithm proposed here. Our algorithm is also compared with the proposed in [123]. In Fig. 3.10 it is shown that the criterion proposed here using the lowest symbol degree for the selection of every punctured symbol is preferable.

3.5 SYNDROME CODING

We finish the first part of the thesis reviewing a coding technique for the problem of source coding with side information (see Sec. 2.2.5 and [134]). We showed in Sec. 2.2.5 that a random binning encoding was enough to achieve the Slepian-Wolf bound, however random binning has no structure and it forces both parties to store the map between

Algorithm 1 Untainted intentional puncturing algorithm

Initialize
$\mathcal{Z}_\infty = \{1, ..., m\}$
$\mathcal{X}_\infty = \{1, ..., n\}$
$j = 1$.
while $j \leqslant p$ and $\mathcal{X}_\infty \neq \emptyset$ **do**
Step 1.– Compute 1-SU under the current pattern
Construct the neighboring set $\mathcal{G}(x_k)$ for all $x_k \in \mathcal{X}_\infty$, $\mathcal{G}(x_k) = \{x_i : x_i \neq x_k, x_i \in \mathcal{N}(z_j), \forall z_j \in \mathcal{M}(x_k) \cap \mathcal{Z}_\infty\}$.
Step 2.– Look for candidates
Make the set of candidates $\Omega \subseteq \mathcal{X}_\infty$, such that $\forall x_p \in \Omega$, $|\mathcal{G}(x_p)| = \min_{x_k \in \mathcal{X}_\infty} |\mathcal{G}(x_k)|$.
Step 3.– Selection for puncturing
Pick a symbol node $x_{n_j} \in \Omega$ (pick one randomly if there exist more than one symbols in Ω).
Step 4.– Updating sets
$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{x_{n_j}\}$
$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{x_i\}$ for each $x_i \in \mathcal{G}(x_{n_j})$
$\mathcal{Z}_\infty = \mathcal{Z}_\infty \backslash \mathcal{M}(x_{n_j})$.
$j = j + 1$
end while

sequences and bins which rapidly becomes unfeasible with the code length.

Wyner proposed in [129] to use the $s(\mathbf{x}) = H\mathbf{x}$ the syndrome in a linear code with an appropriate rate as the bin index. This encoding adds a strong structure to the bins as we have seen in Sec. 3.1.4, but also allows to use channel codes and in particular LDPC codes as we will see later. The decoder outputs the word in the coset specified by s with minimum hamming distance to $\mathbf{y}$.

Once described the syndrome coding technique we prove that syndrome coding and typical decoding achieve the Slepian-Wolf bound, that is, there is no fundamental loss in restricting the encoding of $\mathbf{x}$ to the syndrome of a linear code. The technique used is similar to the one MacKay [71] uses for proving that linear codes are (channel) capacity achieving.

Let H be a full rank binary matrix of size $n \times m$, an encoder sends $s(\mathbf{x})$, the syndrome of an n length sequence $\mathbf{x}$ through a noiseless channel and the decoder, having access also to $\mathbf{y}$, decodes $\hat{\mathbf{x}}$ if $H\hat{\mathbf{x}} = s$ and $\hat{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.

The syndrome size is m and the coding rate achieved is:

$$R = \frac{m}{n} \tag{3.79}$$

There are two types of errors. We have an error if $\mathbf{x}$ is not jointly typical with $\mathbf{y}$, this error source is the same as in the random binning

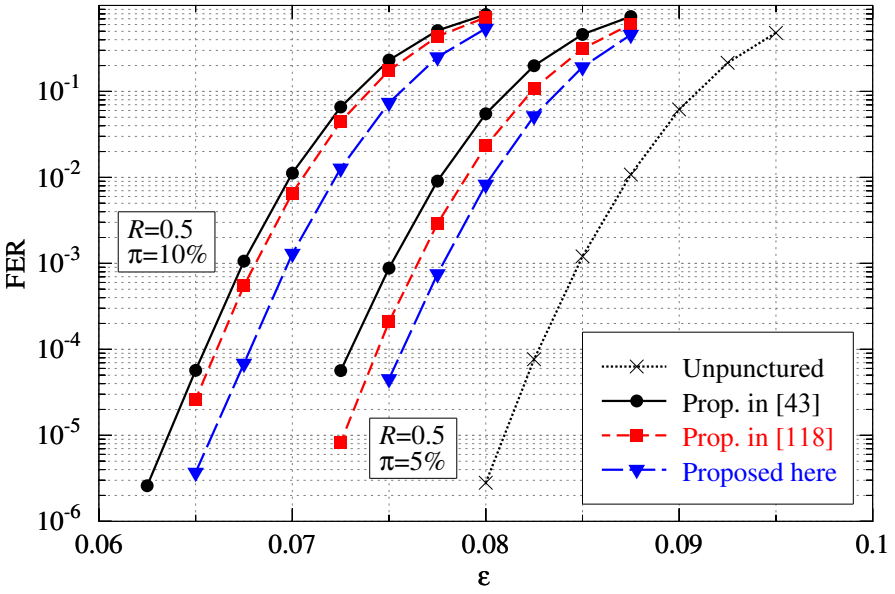

Figure 3.11: FER over the BSC with crossover probability ε. It was used one LDPC code with coding rate $R = 0.5$, and two different proportions of punctured symbols: $\pi = 5\%$ and $\pi = 10\%$.

proof and we know from Eq. 2.41 that it can be made arbitrarily small for n long enough.

In the random binning coding we said that there was an error if it existed $\hat{\mathbf{x}} \neq \mathbf{x}$ jointly typical with $\mathbf{y}$ which shared the same bin index as $\mathbf{x}$. We bounded this error by the size of the set of sequences jointly typical with $\mathbf{y}$ times the probability that $\hat{\mathbf{x}}$ shares the same bin index as $\mathbf{x}$ (see Eq. 2.43).

With syndrome coding there is an error if exists $\hat{\mathbf{x}} \neq \mathbf{x}$ which verifies $H\hat{\mathbf{x}} = s$ and is jointly typical with $\mathbf{y}$. We can bound it in a similar way by the size of the set of sequences jointly typical with $\mathbf{y}$ $|\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})|$, times the probability that $\hat{\mathbf{x}}$ shares the same syndrome as $\mathbf{x}$ $P[H\hat{\mathbf{x}} = H\mathbf{x}]$. The probability that a random sequence verifies a parity equation on a random independently selected subset of bits is $1/2$, consequently the probability that the sequence verifies t parity check equations is $(1/2)^t$ and:

$$\begin{aligned} P_{e_2} &\leqslant |\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})| P[H\hat{\mathbf{x}} = H\mathbf{x}] \\ &= |\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})| P[H(\hat{\mathbf{x}} - \mathbf{x}) = 0] \\ &= |\mathcal{A}_\varepsilon^n(\mathbf{X}|\mathbf{y})| 2^{-m} \\ &\leqslant 2^{-n[R - H(\mathbf{X}|\mathbf{Y}) - 2\delta]} \end{aligned} \tag{3.80}$$

which reproduces Eq. 2.43, in this case the interpretation is that as long as $m > nH(\mathbf{X}|Y)$ for n long enough there exists a code with vanishing error probability.

In a real scenario neither the minimum distance decoder nor the typical decoder can be implemented. It was shown by Liveris et al.

in [65] that LDPC codes can be used within Wyner's coset scheme. The SPA was modified to take into account decoding against syndromes different than zero.

In a linear code all the codewords verify the parity check equations, that is the bits in the equation must add up to $0 \mod 2$, and the checks j sends to the bit i p_{odd} the probability that the equation is verified if bit i takes the value 1 or the associated LLR. We proved in Lem. 9 that this probability is equivalent to the probability that an odd number of the bits take value 1.

In a coset code, all the words verify a syndrome s, that is if $s_j = 0$ the bits in the equation add up to $0 \mod 2$ and if $s_j = 1$ the bits in the equation add up to $1 \mod 2$. In the first case the exchanged messages don't need to be modified, in the second case the probability that the equation is verified if bit i takes the value 1 is equivalent to the probability that an even number of bits take value 1: $p_{even} = 1 - p_{odd}$ and the associated LLR takes the form:

$$L(p_{even}) = \frac{p_{even}}{1 - p_{even}} = \frac{1 - p_{odd}}{p_{odd}} = -L(p_{odd}) \tag{3.81}$$

Part II

OPTIMIZATION OF INFORMATION RECONCILIATION

4 SECRET KEY DISTILLATION

A channel with perfect authenticity but no privacy can be used to repair the defects of a channel with imperfect privacy but no authenticity.

— Charles H. Bennett et al [9]

4.1 INTRODUCTION

4.1.1 *Computational Security*

Security has mattered to man since he was able to store information. It is known that as soon as in ancient Egypt, around 1900 BC, scribes used alternate hieroglyphics in order to make religious texts more difficult to read [58]. The first true account of cyphered or hidden information is thought to be circa 1500 BC, in Mesopotamia [58], where an encrypted tablet was found hiding a recipe for pottery glazes. Since these first attempts, not much more elaborate than ingenious modifications of the text, until now, there has been an active interest on security.

The objective of (information) security is to allow parties to interact with data only as established by legitimate users. There are several related concepts that can be required by different users or for different goals. The objective pursued by the methods described in the previous paragraph is data confidentiality; i.e. the Mesopotamians and to some extent the Egyptian scribes meant to store information in such a way that only a valid entity could understand it. There are other security goals beyond confidentiality; two examples are integrity and authenticity. That is, users might want to be sure that a message has not been corrupted in any way or they might require a system or a protocol to provide genuine, authentic data.

The general model to achieve confidentiality consists in altering the raw message or plain-text so that an eavesdropper is unable to get any meaningful information. Formally the plain-text m is chosen from a discrete set of messages and altered or encrypted into a cypher-text e with an encrypting function f_{enc} and a secret key k such that

$$e = f_{enc}(m, k) \tag{4.1}$$

The plain-text can be recovered from the cypher-text with a decoding function f_{dec} and k:

$$m = f_{dec}(e, k) \tag{4.2}$$

If we leave out of our analysis real devices and implementations, the security of a cryptographic system resides in the difficulty that an eavesdropper faces for recovering the original message without having access to the secret key. It is common to suppose the eavesdropper to be limited in her computing resources and hence incapable of solving some problems. In particular, most crypto-systems require the existence of some one-way trapdoor functions.

A function f is said to be one-way if given x in the domain of f the image can be computed by a polynomial time algorithm, while given an image y in the range of f there is no algorithm that can compute a preimage in polynomial time. A trapdoor one-way function is a one-way function in the sense that there is no algorithm that can find a preimage in polynomial time except if some additional information k is known.

Rivest, Shamir and Adleman's algorithm (RSA) [104], arguably the most used encryption method relies on prime multiplication as a trapdoor one-way function. Rabin's crypto-system [93] as well as ElGamal's [26] are secure in the same sense as long as modular square roots and discrete logarithms are difficult to compute. However it remains to be proved if these functions are indeed one-way or not, even the existence of any one-way function is unknown, as proving their existence would imply that $P \neq NP$ a well known open problem in complexity theory [18].

On the other hand, it is known that a quantum computer can solve the above mentioned problems in polynomial time. In particular, Shor's algorithm [112] can compute discrete logarithms and factorize numbers in polynomial time. Quantum computers are in an embryonic stage and the current prototypes are proofs of concept capable of operating with just a few states. For instance, recently a team at the University of Bristol [92] developed a chip that implemented Shor's algorithm and was able to factor the number 15.

The security paradigm which relies on the computational resources available to the eavesdropper is known as computational security. We have discussed its theoretical weaknesses which can be summarized in the idea that it does not offer any theoretical guarantees on its security. However, computational security is very convenient: some crypto-systems are very easy to implement and some security primitives are possible only if a possible eavesdropper has limited resources. In fact, nowadays computational security based crypto-systems are ubiquitous and can be found in most commercial products. Even from the security point of view, algorithms such as

RSA have been exposed many years to the attacks of theorists and no critical flaw has been found other than implementation problems.

4.1.2 *Information Theoretic Security*

Security can also be studied without any assumption on the eavesdropper capabilities. That is, Eve, an eavesdropper, is supposed to have unlimited resources. This security paradigm is known as Information Theoretic Security (ITS) and hereafter is the only one that we shall consider.

Claude Shannon opens the field of ITS with his 1949 paper Communication Theory of Secrecy Systems [108]. In his model, two parties want to exchange a message m with the help of a shared key k in the presence of an eavesdropper. Alice composes a cypher-text $e = f_{enc}(m, k)$ and sends it to Bob. Shannon defines perfect secrecy or as we call it ITS if an illegitimate party, having access to e, does not see an increase in the probability of guessing the right m, i.e. e does not leak any information about m, for this to happen the number of cypher-texts should at least equal the number of messages. This informal definition implies that the messages are equally likely. Instead, if we associate every message with a mass probability the concept of entropy arises naturally.

Suppose that we have a discrete set of messages, m_1, m_2, ..., m_n with probabilities $p(m_1)$, $p(m_2)$, ..., $p(m_n)$. The function $e = f_{enc}(k, m)$ transforms a message into cypher-text e with the use of a key k selected from a discrete set of keys. The eavesdropper can intercept e and compute the a posteriori probabilities of m, $p(m|e)$. A crypto-system is said to be information theoretically secure if for all cypher-texts the a priori and a posteriori probabilities of all messages remain unchanged. If it were not the case there would exist a cypher-text for which the eavesdropper would gain some insight on which was the message sent. More precisely, a crypto-system is said to be perfectly secure if and only if $p(e|m) = p(e)$ for all e and m, that is if e and m are independent. If we think in the entropy associated with the random variables $\mathbf{E}$ and $\mathbf{M}$, representing the cypher-text and message distributions respectively, this same condition can be more compactly written as:

$$H(\mathbf{E}|\mathbf{M}) = H(\mathbf{E}) \tag{4.3}$$

The model described by Shannon is limited as it requires the sender and the eavesdropper to share a key before securely communicating, which is not possible in many scenarios. However, it is straightforward to generalize Eq. 4.3. We call a secret object —be it a secret key or a secret message— information theoretically secure if the information available to a possible eavesdropper does not reduce

the entropy of the secret. Though many scenarios can be studied under the prism of ITS, the two main objects of study are confidential communications and SKD protocols. In the most basic ITS confidential communications model, a legitimate sender is connected to a legitimate receiver and an eavesdropper through two different though correlated noisy channels, secret transmission is possible in general if the sender adds some randomness to his message such that only the legitimate receiver can decode it. In a SKD protocol two legitimate parties share some common randomness source and wish to distill a secret key with ITS.

4.2 SECRET KEY DISTILLATION

The main ingredient needed for SKD is a source of correlated randomness. But beyond this requirement there are several assumptions and hypotheses that model different scenarios under which SKD can be studied.

One common assumption is that all the parties have access to the outcomes of a specific experiment repeated many times. If this assumption holds the parties can safely regard an average behavior as the law of large numbers guarantees that the joint outcome will be typical with high probability (see Sec. 2.2.5). However assuming an iid scenario might be unrealistic in some situations, in these cases SKD can be considered for a single outcome of a joint distribution. This second, more restrictive, scenario is sometimes referred as one-shot distillation [121].

A source of correlated randomness is, in many cases, not enough to distill a random key. The output of the random source is a raw key that is neither shared by the legitimate parties, neither secret to the eavesdropper. In order to complete the distillation process the legitimate parties need access to a public channel —if it were private they would not need the randomness source—. Over a public channel the parties can discuss and distill a secret key. The public channel is some times supposed to be authentic or, similarly, it is supposed that the legitimate parties have short common secret key that they can use to authenticate their messages with ITS [126, 114]. It is not clear if the legitimate parties can, in some scenarios, use non-ITS authentication schemes and still distill a information theoretic secret key. However there are strong indications in the form of explicit attacks that ITS authentication schemes should always be used [87]. It should be noted that even if the public channel is not authentic the legitimate parties might be able, if the common randomness source verifies some criteria, to distill a secret key [78, 79, 80].

Finally, the communications on the public channel might be one-way or two-ways. We have chosen to focus on the one-way communications version of the different scenarios, the practical advantages

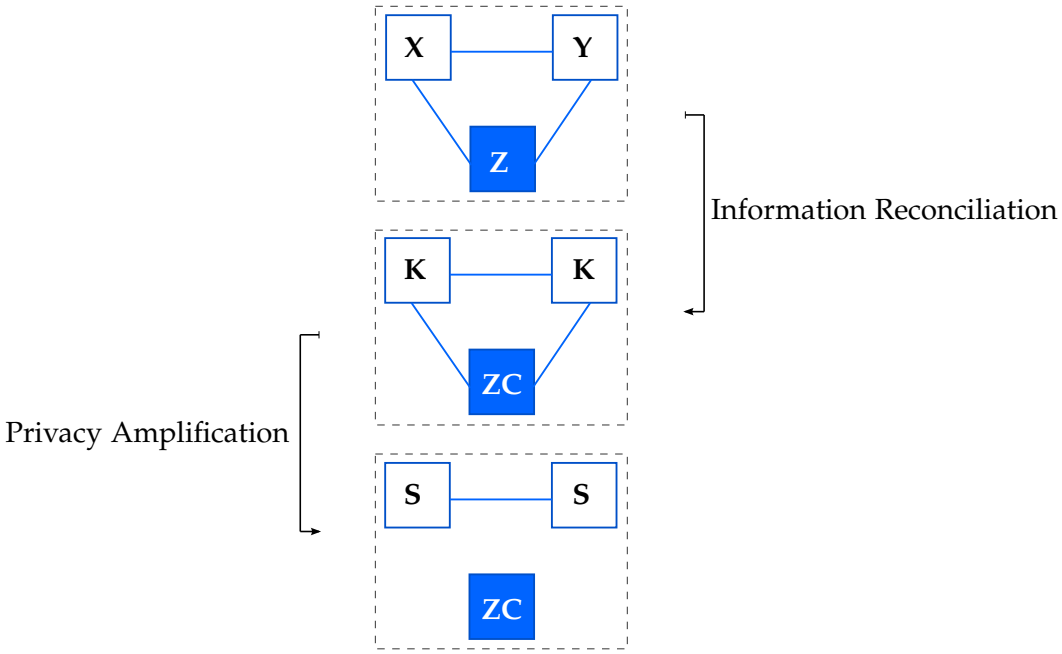

Figure 4.1: The secret key distillation process is divided in two steps: information reconciliation and privacy amplification.

of these models is evident if we think in the reduced distillation complexity, network requirements, etc. our optimization of the distillation process in Chap. 6 is of special interest if only one-way communications are available. However, it should be noted that two-way communications can be used to distill a key in scenarios where one-way secret key distillation is not possible [82] and, in general, the amount of distillable secret key with two-way communications is equal or greater than with one-way communications [124, 42, 43].

4.3 INFORMATION RECONCILIATION AND PRIVACY AMPLIFICATION

There are two questions that allow to gain insight in the key distillation process [97]. The first one is what would be the minimum length of an encoding of variable $\mathbf{X}$, $H^{\varepsilon}_{\text{enc}}(\mathbf{X}|\mathbf{Y})$ such that a decoder with access to side information $\mathbf{Y}$ can recover $\mathbf{X}$ with success probability at least $1-\varepsilon$.

The second question we can ask is the length of the longest random key that can be extracted from $\mathbf{X}$, $H^{\varepsilon}_{\text{ext}}(\mathbf{X}|\mathbf{Y})$, such that the key is uniformly distributed and independent of a random variable $\mathbf{Y}$.

Protocols that distill a secret key usually divide the distillation process in two different phases. In the first one, known as information reconciliation or simply reconciliation, Alice and Bob exchange redundant information over the public channel in order to eliminate any discrepancy in their correlated sequences, $\mathbf{X}$ and $\mathbf{Y}$ respectively.

At the end of the reconciliation phase both parties have agreed on a shared string $\mathbf{K}$, though in many cases $\mathbf{K} = \mathbf{X}$. It is easy to see that, by definition, $H^{\varepsilon}_{enc}(\mathbf{K}|\mathbf{Y})$ represents a tight lower bound on the minimum length of the messages exchanged to reconcile $\mathbf{X}$ and $\mathbf{Y}$ with error probability smaller than ε.

On the second phase, known as privacy amplification, Alice and Bob shrink their strings in order to wipe any information of the previously shared key that the eavesdropper could have on $\mathbf{K}$ through $\mathbf{Z}$ or through any communication $\mathbf{C}$ exchanged over the public channel with information about the strings. In this case $H^{\varepsilon}_{ext}(\mathbf{K}|\mathbf{Y})$ stands by definition as the maximum number of random bits that a privacy amplification procedure can extract.

This construction allows to split the secret key distillation process into two easier problems. However artificial it might seem, the division is not necessarily suboptimal. On the contrary, it is explicitly used to reach the secret key bounds in all one-way scenarios [82, 81, 97, 98]

4.4 SCENARIOS

4.4.1 *One-Shot Secret Key Distillation*

One-shot distillation, traditionally considered much more complex than the repetition scenario, was studied in detail by Renner et al. in [97]. The tight bounds for one-way key distillation presented here are the main results of the paper. Two legitimate parties Alice and Bob wish to distill a secret key in the presence of an eavesdropper Eve. Alice, Bob and Eve hold a single outcome of the joint experiment given by $P_{\mathbf{XYZ}}$, additionally Alice can send public messages to Bob over a public, noiseless and authentic channel.

We say that Alice and Bob distill an ε-secure key if they run a protocol that outputs the keys $\mathbf{S_A}$, $\mathbf{S_B}$ to Alice and Bob respectively, and these keys are identical, uniformly distributed and independent of any knowledge the eavesdropper has. In particular if $p(\mathbf{S_A} \neq \mathbf{S_B}) < \varepsilon_1$ and $d(\mathbf{S_A}|\mathbf{Z}) < \varepsilon_2$ Alice and Bob hold an $(\varepsilon_1 + \varepsilon_2)$-secure key. The length of the longest ε-secure key that Alice and Bob can distill if they limit to one-way communications from Alice to Bob is denoted by $S^{\varepsilon}(\mathbf{X} \to \mathbf{Y}||\mathbf{Z})$.

It turns out, see [97] for a formal proof, that $H^{\varepsilon}_{enc}(\mathbf{X}|\mathbf{Y})$ and $H^{\varepsilon}_{ext}(\mathbf{X}|\mathbf{Y})$ in the one-shot scenario are both tightly bounded by the smooth max-entropy and min-entropy defined in Sec. 2.2.4:

$$H^{\varepsilon}_{0}(\mathbf{X}|\mathbf{Y}) \leqslant H^{\varepsilon}_{enc}(\mathbf{X}|\mathbf{Y}) \leqslant H^{\varepsilon_1}_{0}(\mathbf{X}|\mathbf{Y}) + \log \frac{1}{\varepsilon_2} \tag{4.4}$$

$$H^{\varepsilon}_{\infty}(\mathbf{X}|\mathbf{Y}) \geqslant H^{\varepsilon}_{ext}(\mathbf{X}|\mathbf{Y}) \geqslant H^{\varepsilon_1}_{\infty}(\mathbf{X}|\mathbf{Y}) - 2\log \frac{1}{\varepsilon_2} \tag{4.5}$$

where in both relations $\varepsilon = \varepsilon_1 + \varepsilon_2$.

The secret key rate in the one shot scenario directly follows from the optimization of these two relations:

$$M^{\varepsilon}(\mathbf{XY}|\mathbf{Z}) = \sup_{P_{UV|X}} H_{\infty}^{\varepsilon}(\mathbf{U}|\mathbf{ZV}) - H_0^{\varepsilon}(\mathbf{U}|\mathbf{YV}) \tag{4.6}$$

that is, Alice is free to preprocess $\mathbf{X}$ and obtain random variables that are specially prepared to maximize the final secret key via information reconciliation and privacy amplification. The preprocessing can be summarized by two random variables: $\mathbf{U}$ that she keeps and $\mathbf{V}$ that she sends (publicly) to Bob. We can then bound $S^{\varepsilon}(\mathbf{X} \to \mathbf{Y}||\mathbf{Z})$ by:

$$M^{\varepsilon_1}(\mathbf{XY}|\mathbf{Z}) - O(\log \frac{1}{\varepsilon_2}) \leqslant S^{\varepsilon}(\mathbf{X} \to \mathbf{Y}||\mathbf{Z}) \leqslant M^{\varepsilon}(\mathbf{XY}|\mathbf{Z}) \tag{4.7}$$

The lower bound holds because any ε_1 and ε_2 can be chosen such that the key is ε-secure and by construction is a lower bound. The upper bound follows because $M^{\varepsilon}(\mathbf{XY}|\mathbf{Z})$ verifies a set of conditions which imply that its value can not be increased by the execution of any protocol [97].

If Alice and Bob hold identical keys $\mathbf{X}$, by Eq. 4.5, the amount of ε-secret key that can be extracted is lower bounded by:

$$S^{\varepsilon}(\mathbf{X} \to \mathbf{X}||\mathbf{Z}') \geqslant H_{\infty}^{\varepsilon_1}(\mathbf{X}|\mathbf{Z}') - 2\log \frac{1}{\varepsilon_2} \tag{4.8}$$

We can further develop the relation to measure the effect of information reconciliation [84]. The knowledge of the eavesdropper $\mathbf{Z}'$ can be decomposed in $\mathbf{Z}$ her original knowledge and $\mathbf{C}$ an encoding of $\mathbf{X}$ sent through the public channel

$$H_{\infty}^{\varepsilon_a+\varepsilon_b}(\mathbf{X}|\mathbf{ZC}) \geqslant H_{\infty}^{\varepsilon_a}(\mathbf{X}|\mathbf{Z}) - H_0(\mathbf{C}) - \log(\frac{1}{\varepsilon_b}) \tag{4.9}$$

where $H_0(\mathbf{C})$ can be regarded as the number of bits of the conversation on the public channel. The encoding is lower bounded by $H_0^{\varepsilon}(\mathbf{X}|\mathbf{Y})$ but essentially every extra bit used for information reconciliation reduces one bit the length of the final secret key.

4.4.2 *Source Type Model with Wiretapper*

The source type model with wiretapper or simply model SW was first introduced by Ahlswede and Csiszár in [5]. In this scenario the three parties Alice, Bob and an eavesdropper Eve hold the outcomes of n repetitions of an experiment given by P_{XYZ}.

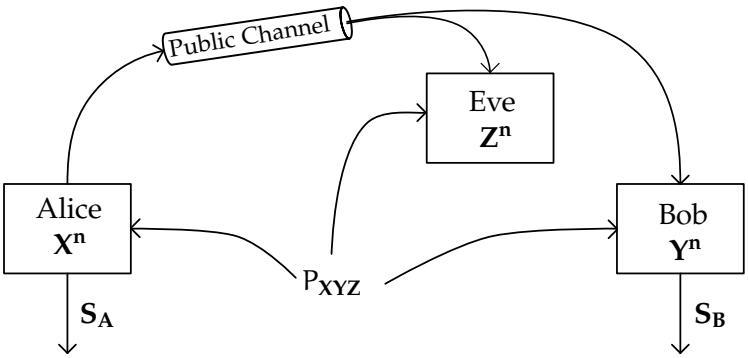

Figure 4.2: Ahlswede and Csiszár's model SW.

Fig. 4.2 shows that Alice can send additional information through a public channel. After a given number of uses of the public channel, we denote as $\mathbf{C}$ the set of messages that Alice sends over the public channel to Bob, Alice and Bob estimate their shared keys to be $\mathbf{S_A}$ and $\mathbf{S_B}$ respectively by using an agreed protocol.

Definition 3. *A strong secret key rate* R_s *is achievable if for large enough* n *and for every* $\varepsilon > 0$ *the legitimate parties can distill a key pair (*$\mathbf{S_A}$ *and* $\mathbf{S_B}$*) that meets simultaneously the following restrictions [62]:*

$$\Pr[\mathbf{S_A} \neq \mathbf{S_B}] < \varepsilon \tag{4.10}$$

$$I(\mathbf{C}, \mathbf{Z^n}; \mathbf{S_A}) < \varepsilon \tag{4.11}$$

$$H(\mathbf{S_A}) > n \cdot \mathbf{R_s} - \varepsilon \tag{4.12}$$

$$H_0(\mathbf{S_A}) < H(\mathbf{S_A}) + \varepsilon \tag{4.13}$$

This definition of secret key rate is strong compared to previous definitions in which the convergence of the conditions was asymptotic and not absolute. In [81] it is shown that both sets of conditions share the same bounds for secret key generation.

The largest achievable secret rate is called the secret key capacity.

$$S^n(\mathbf{X} \to \mathbf{Y}||\mathbf{Z}) = \sup_{p_{UV|X}} H(\mathbf{U}|\mathbf{ZV}) - H(\mathbf{U}|\mathbf{YV}) \tag{4.14}$$

the main difference with respect to the one-shot scenario is that in the asymptotic case the smooth min-entropy and max-entropy converge

to the standard entropy measure. In fact if Alice and Bob's outcomes correspond to n independent outcomes of the same experiment we have [97]:

$$\lim_{\varepsilon\to 0}\lim_{n\to\infty}\frac{H^{\varepsilon}_{enc}(\mathbf{X^n}|\mathbf{Y^n})}{n}=\lim_{\varepsilon\to 0}\lim_{n\to\infty}\frac{H^{\varepsilon}_{ext}(\mathbf{X^n}|\mathbf{Y^n})}{n}=H(\mathbf{X}|\mathbf{Y}) \quad (4.15)$$

The achievability proof in [81] is based on universal families of hash functions [126].

Definition 4. *A family of functions $H : \mathcal{A} \to \mathcal{B}$ is called universal if $\forall x, y \in \mathcal{A}$ and a function h drawn uniformly from H, $h(x) = h(y)|x \neq y$ holds with probability less or equal than $1/|\mathcal{B}|$.*

Now, if the knowledge of the eavesdropper can be bounded in the sense that the conditional collision entropy (see Sec. 2.2.4) on the key $\mathbf{S_A}$ given $\mathbf{Z}'$ is at least c for any value that $\mathbf{Z}'$ takes, Bennett et al. [11] show that the legitimate parties can extract approximately $H_2(\mathbf{S_A}|\mathbf{Z}')$ secret bits:

Theorem 1. *Let $\mathbf{X}$ and $\mathbf{Z}'$ be two correlated random variables. If $H_2(\mathbf{S_A}|\mathbf{Z}') > c$, then the entropy of a key $\mathbf{S_A}$ generated by the application of a function h_U uniformly chosen at random from a universal family of hash functions $H : \mathcal{X} \to \{0,1\}^k$ is given by:*

$$H(\mathbf{S_A}|\mathbf{U}\mathbf{Z}' = z') \geqslant k - 2^{k-c}/\log 2 \quad (4.16)$$

which wipes all the information from the eavesdropper provided that Alice and Bob can estimate $H_2(\mathbf{S_A}|\mathbf{Z}')$.

The effects of the $|\mathbf{C}|$ redundancy bits shared on the conditional Renyi entropy can be bounded using a security parameter t with probability $1 - 2^{-(t/2-1)}$ [16]:

$$H_2(\mathbf{X}|\mathbf{Z}' = zc) \geqslant H_2(\mathbf{X}|\mathbf{Z} = z) - |\mathbf{C}| - t \quad (4.17)$$

measuring the interest of good information reconciliation. Every redundancy bit used in the information reconciliation phase reduces the final secret key.

4.4.3 *Channel Type Model with Wiretapper*

Let us consider now the channel-type model with wiretapper for secret key agreement introduced by Ahlswede and Csiszár [5] as shown in Fig. 4.3. In this model a legitimate party, Bob, and an eavesdropper, Eve, are both connected to another legitimate party, Alice, through a Discrete Memoryless Channel (DMC). Alice generates a discrete sequence of n values, $\mathbf{X^n}$, while Bob and Eve observe the correlated

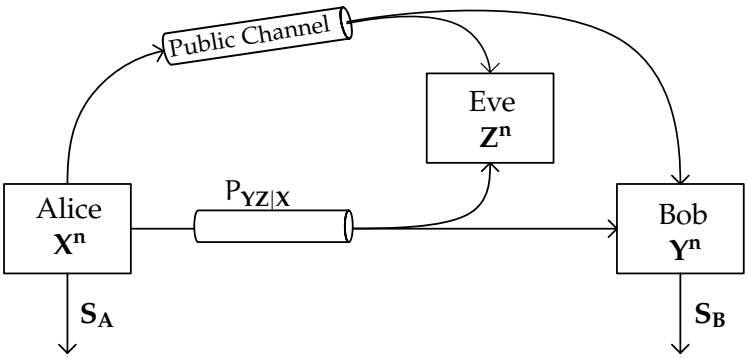

Figure 4.3: Ahlswede and Csiszár's model CW.

outputs, $\mathbf{Y^n}$ and $\mathbf{Z^n}$ respectively, obtained after the transmission of $\mathbf{X^n}$ over the DMC. Both outputs are characterized by the transition probability $P_{YZ|X}$, with each component of the sequences being the outcome of an independent use of the channel. Alice and Bob have also access to a public but authenticated channel used to distill a shared secret key from their correlated sequences. The definition of a secret key rate in this circumstances is identical to the source model with wiretapper that we discussed in Sec. 4.4.2.

The largest achievable secret rate is called the secret key capacity $\mathbf{C_S}$. It is first derived in by [5]:

$$C_S(\mathbf{X} \to \mathbf{Y}||\mathbf{Z}) = \max_{P_{UVX}} [H(\mathbf{U}|\mathbf{ZV}) - H(\mathbf{U}|\mathbf{YV})] \tag{4.18}$$

where we see that with respect to Eq. 4.14 the key is maximized for all random variables $\mathbf{X}$.

Let us consider the effect of imperfect information reconciliation in the channel model with wiretapper. As a first step we review the privacy amplification result that allow to take into account the impact of reconciliation in the final key. An extractor is a function that, with a small amount of random bits acting as catalyst, obtains a number of almost uniformly distributed random bits from a source.

Theorem 2. *Given three constants $\delta, \Delta_1, \Delta_2 \geqslant 0$, after n uses of a binary symmetric channel ruled by $P_{\mathbf{Z}'|\mathbf{X}}$, if Eve's min-entropy on $\mathbf{X}$ is known to be bounded as $H_\infty(\mathbf{X}|\mathbf{Z}' = z') \geqslant \delta n$, there exists ([81]) an extractor function $E : F_2^n \times F_2^u \to F_2^k$, with $u \leqslant \Delta_1 n$ and $k \geqslant (\delta - \Delta_2)n$, such that if Alice and Bob agree on secret key $\mathbf{S_A} = E(\mathbf{X}, \mathbf{U})$, where $\mathbf{U}$ is a sequence of u random uniform bits, the entropy of $\mathbf{S_A}$ is given by:*

$$H(\mathbf{S_A}|\mathbf{UZ}' = z') \geqslant k - 2^{-n^{1/2-o(1)}} \tag{4.19}$$

The effects of the $|\mathbf{C}|$ redundancy bits shared on the conditional min-entropy can also be bounded using a security parameter t with probability $1-2^{-t}$ [81]:

$$H_\infty(\mathbf{X}|\mathbf{Z}'=zc) \geqslant H_\infty(\mathbf{X}|\mathbf{Z}=z) - |\mathbf{C}| - t \tag{4.20}$$

the effect is, as expected, identical to the source model with wiretapper: every redundancy bit used for information reconciliation reduces the length of the final key in one bit.

4.4.4 *Quantum Key Distribution*

QKD is probably the main practical application of SKD. In a QKD protocol [7, 41, 106], two legitimate parties, Alice and Bob, aim at sharing an information theoretic secret key, even in the presence of an eavesdropper Eve. In the quantum part of such a protocol, Alice and Bob exchange quantum signals, e.g. single photons, which carry classical information. For instance, Alice encodes a classical bit onto the polarization or the phase of a photon and sends this photon to Bob who measures it. After repeating this step n times, Alice and Bob share two strings, $\mathbf{X}$ and $\mathbf{Y}$. Eve has access to a quantum system $\mathbf{Z}$.

In any realistic implementation of a QKD protocol, $\mathbf{X}$ and $\mathbf{Y}$ suffer discrepancies mainly due to losses in the channel and noise in Bob's detectors but which are conservatively attributed to the action of an eavesdropper. Therefore, any QKD protocol must include a classical post-processing step in order to extract a secret key from the correlated strings $\mathbf{X}$ and $\mathbf{Y}$. This SKD process is similar to the models previously introduced: in a first step the legitimate parties reconcile the strings obtained from their randomness source and in a second step they produce a smaller but more secure key.

Let us give some basic definitions about the quantum counterparts of information measures [86, 95]. A state in a quantum system with d degrees of freedom is described by ρ a trace one, positive-semidefinite and self-adjoint operator in $\mathcal{H}^d$, a d-dimensional Hilbert space . That is, a state ρ, verifies:

$$\mathrm{Tr}(\rho) = \sum_{i=1}^{n} \rho_{ii} = 1 \tag{4.21}$$

$$\overline{x}\rho x \geqslant 0, \forall x \in \mathbb{C}^n \tag{4.22}$$

$$\rho = \overline{\rho}^T \tag{4.23}$$

where ρ^T represents the transposed of ρ and $\overline{\rho}$ the conjugate of ρ. We denote by $\mathcal{P}(\mathcal{H}^d)$ the set of all operators describing quantum states in $\mathcal{H}^d$.

We define the distance between two states by:

$$D(\rho, \sigma) = \frac{1}{2} Tr(\sqrt{\overline{(\rho - \sigma)}(\rho - \sigma)}) \tag{4.24}$$

The equivalent of the entropy of a random variable in the quantum world is the von Neumann entropy of a state ρ, it is defined as:

$$S(\rho) = -Tr(\rho \log \rho) \tag{4.25}$$

Given $\rho^{\mathbf{X}}$ and $\rho^{\mathbf{Y}}$, two quantum states that are both part of a larger system represented by $\rho^{\mathbf{XY}}$, the quantum joint entropy and mutual information are defined by:

$$S(\mathbf{X}, \mathbf{Y})_\rho = S(\rho^{\mathbf{XY}}) = -Tr(\rho^{\mathbf{XY}} \log \rho^{\mathbf{XY}}) \tag{4.26}$$

$$S(\mathbf{X}; \mathbf{Y})_\rho = S(\rho^{\mathbf{X}}) + S(\rho^{\mathbf{Y}}) - S(\rho^{\mathbf{XY}}) \tag{4.27}$$

Let $\rho_{\mathbf{XY}} \in \mathcal{P}(\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}})$ and $\sigma_{\mathbf{Y}} \in \mathcal{P}(\mathcal{H}_{\mathbf{Y}})$. The conditional quantum min-entropy of $\rho_{\mathbf{XY}}$ relative to $\sigma_{\mathbf{Y}}$ is given by:

$$H_\infty(\rho_{\mathbf{XY}}|\sigma_{\mathbf{Y}}) = -\log \min \lambda \in \mathbb{R} | \lambda id_{\mathbf{X}} \otimes \sigma_{\mathbf{Y}} - \rho_{\mathbf{XY}} \geqslant 0 \tag{4.28}$$

Let $\rho_{\mathbf{XY}} \in \mathcal{P}(\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}})$. The conditional quantum min-entropy of $\rho_{\mathbf{XY}}$ given $\mathcal{H}_{\mathbf{Y}}$ is defined as:

$$H_\infty(\rho_{\mathbf{XY}}|\mathbf{Y}) = \sup_{\sigma_{\mathbf{Y}}} H_\infty(\rho_{\mathbf{XY}}|\sigma_{\mathbf{Y}}) \tag{4.29}$$

We finally consider the smooth generalization of the conditional min-entropy in the quantum setting. Let $\rho_{\mathbf{XY}} \in \mathcal{P}(\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}})$ be a bipartite quantum state and $\varepsilon \geqslant 0$. The smooth min-entropy of $\rho_{\mathbf{XY}}$ relative to $\sigma_{\mathbf{Y}}$ and given $\mathcal{H}_{\mathbf{Y}}$ is, respectively, given by:

$$H_\infty^\varepsilon(\rho_{\mathbf{XY}}|\sigma_{\mathbf{Y}}) = \sup_{\hat{\rho}_{\mathbf{XY}}} H_\infty(\hat{\rho}_{\mathbf{XY}}|\sigma_{\mathbf{Y}}) \tag{4.30}$$

where the supremum is found over all $\hat{\rho}_{\mathbf{XY}}$ such that $D(\rho_{\mathbf{XY}}, \hat{\rho}_{\mathbf{XY}}) \leqslant \varepsilon$.

$$H_\infty^\varepsilon(\rho_{\mathbf{XY}}|\mathbf{Y}) = \sup_{\sigma_{\mathbf{Y}}} H_\infty^\varepsilon(\rho_{\mathbf{XY}}|\sigma_{\mathbf{Y}}) \tag{4.31}$$

We could address both the one-shot and the source model scenarios of QKD. In the former, the size $\mathbf{K}$ of the secret is given by a Csiszar-Körner-type formula [22]:

$$S_q^n(\mathbf{XY}|\mathbf{Z}) = I(\mathbf{X};\mathbf{Y}) - S(\mathbf{X};\mathbf{Z}) \tag{4.32}$$

where the result from Csiszar and Körner has been generalized [24] to quantum settings by replacing the mutual information $I(\mathbf{X};\mathbf{Z})$ by its quantum counterpart $S(\mathbf{X};\mathbf{Z})$. Two different measures of information are used in this formula because the assumptions made on Alice's, Bob's and Eve's capabilities are different. Eve is not supposed to be restricted to classical correlations and could for instance use quantum technologies (quantum computer, quantum memory) to perform her attack. The secret key from Eq. 4.32 is valid only in the asymptotic case. However a real system has only access to finite resources, which means that Alice and Bob not only have access to bounded computational power but also they have to distill a secret key from a finite number of quantum systems. It is thus clear the interest of the one-shot scenario in the context of QKD.

In the finite length scenario, we consider ε-security in the same sense that we defined it in the one-shot scenario. For some $\varepsilon > \varepsilon_1 > 0$ the amount of ε-secure key that the legitimate parties can distill is upper bounded by [105]:

$$H_\infty^{\varepsilon_1}(\mathbf{X}|\mathbf{ZC}) - 2\log\frac{1}{\varepsilon_{PA}} \geqslant l^\varepsilon \tag{4.33}$$

where ε_{PA} represents the probability that the privacy amplification fails.

We can measure the net impact of information reconciliation by a decrease in the smooth min-entropy. It is shown in [105] that:

$$H_\infty^{\varepsilon_1}(\mathbf{X}|\mathbf{ZC}) \geqslant H_\infty^{\varepsilon_1}(\mathbf{X}|\mathbf{Z}) - \text{leak} \tag{4.34}$$

where leak is a purely classical term that tracks the length of bits correlated with $\mathbf{X}$ and $\mathbf{Y}$ that have been revealed:

$$\text{leak} = H_0(\mathbf{C}) - H_\infty(\mathbf{C}|\mathbf{XY}) \tag{4.35}$$

The main effect of an imperfect reconciliation is clearly a reduction of the secret key rate, which in turn, in terms of the figures of merit of a QKD protocol, limits the distance range over which secret keys can be distilled [89, 106]. This is the reason why the reconciliation should be as efficient as possible.

5 INFORMATION RECONCILIATION

The block parity disclosure approach [...]
forces Alice and Bob to sacrifice at least one bit
in each block on the altar of privacy.

— Charles H. Bennett et al. [10]

5.1 INTRODUCTION

In this chapter we compare several practical information reconciliation protocols, the objective is to show that though there are several ad-hoc protocols proposed for the task, adapted error correcting codes are an ideal solution from the efficiency point of view. In order to compare the different reconciliation methods we concentrate in reconciliation methods for correlated discrete sources even if the ideas presented here can be easily extrapolated to other scenarios. For instance, recently, they have been considered for the reconciliation of continuous-variable QKD [56].

Beyond the reconciliation efficiency, there are two other parameters to consider when evaluating the quality of a information reconciliation procedure: that is the computational complexity and the interactivity. The first one stresses that a real information reconciliation procedure must be feasible. Any sufficiently long random linear code of the appropriate rate could solve the problem [134], however optimal decoding is in general an NP-complete problem [12]. The interactivity of a reconciliation protocol should also be taken into account because, specially in high latency scenarios, the communications overhead can pose a severe burden on the performance of a SKD protocol.

The rest of the chapter is organized as follows: in section 5.2 we describe what would be an optimal protocol, in section 5.3, we review the literature on information reconciliation from the first protocols and their improvements to Cascade protocol which is currently the solution adopted in most implementations, in section 5.4, we describe some other proposals and optimizations and in section 3.3, we present a reconciliation technique based on LDPC codes optimized for the BSC. This technique was introduced in [28].

5.2 INFORMATION RECONCILIATION IS ERROR CORRECTION

Let Alice and Bob be two parties with access to correlated strings that can be regarded as the outcomes of a joint experiment given by instances of two random variables, $\mathbf{X}$ and $\mathbf{Y}$ respectively. Information

reconciliation is the process by which Alice and Bob extract common information from their correlated sources. In a practical setting Alice and Bob hold $\mathbf{x}$ and $\mathbf{y}$, two n-length strings that are the outcome of one or many repetitions of the random experiment. They wish to agree in some string $\mathbf{s} = f(\mathbf{x}, \mathbf{y})$ through one-way or bidirectional conversation [121]. The conversation $\phi(\mathbf{x}, \mathbf{y})$ is also a function of the outcome strings, and its quality can be measured by two parameters: the length of the conversation $\mathbf{c} = |\phi(\mathbf{x}, \mathbf{y})|$ and the probability that the reconciliation scheme fails.

More precisely, we say that a reconciliation protocol $R(\mathbf{x}, \mathbf{y}) = [\mathbf{s_x}, \mathbf{s_y}, \mathbf{c}]$ is a protocol that produces the strings $\mathbf{s_x}$ and $\mathbf{s_y}$ from the strings $\mathbf{x}$ and $\mathbf{y}$ exchanging the string $\mathbf{c}$ through the public channel. A protocol R is said to be ε-robust [14] if:

$$\exists n_0 | \forall n \geqslant n_0 \sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} p(\mathbf{x}, \mathbf{y}) p(\mathbf{s_x} \neq \mathbf{s_y}) \leqslant \varepsilon \tag{5.1}$$

in the one-shot scenario the protocol is ε-robust simply if for n, the length of the instances $\sum_{\mathbf{x},\mathbf{y} \in \{0,1\}^n} p(\mathbf{x}, \mathbf{y}) p(\mathbf{s}_x \neq \mathbf{s}_y) \leqslant \varepsilon$.

Once it has been separated from privacy amplification, the problem is reduced to one of Slepian-Wolf coding [113] (see Fig. 2.2). Wyner's coset scheme is a good solution for the compression of binary sources with side information (see Sec. 3.5). The efficiency of an information reconciliation protocol sending a sequence $\mathbf{c}$ through the public channel to help Bob recover $\mathbf{x}$ using side information $\mathbf{y}$ with probability higher than $1 - \varepsilon$, can be measured using a quality parameter f^ε:

$$f^\varepsilon = \frac{|\mathbf{c}|}{nH(X|Y)} \tag{5.2}$$

Let R be an ε-robust reconciliation protocol, as a direct consequence of the Slepian-Wolf bound, we can prove that the reconciliation efficiency is equal or greater than one [14]:

$$\lim_{n \to \infty} f^\varepsilon = \frac{|\mathbf{c}|}{nH^\varepsilon(\mathbf{X}|\mathbf{Y})} \geqslant 1 \tag{5.3}$$

in consequence, we say that a protocol is optimal if $f = 1$.

The definition of the reconciliation efficiency in the one-shot scenario is similar:

$$f^\varepsilon = \frac{|\mathbf{c}|}{H_0^\varepsilon(\mathbf{X}|\mathbf{Y})} \tag{5.4}$$

in this scenario f^ε is greater than one by definition. If we can write $p_{XY} = (p_{UV})^n$ for some $\mathbf{UV}$ we have by Eq. 4.15 that the one-shot efficiency converges to the efficiency in the repetition scenario.

Hereafter we drop the ε superscript; whenever relevant for the discussion we will specify its value.

We discussed in Sec. 3.5 the appropriateness of (linear) error correcting codes for the Slepian-Wolf problem. In consequence error correcting codes can be used for information reconciliation. Let R be the coding rate of a code $\mathcal{C}(n-k)$ the reconciliation efficiency when using the code to reconcile chains $\mathbf{x}$ and $\mathbf{y}$ is given by:

$$f_{\mathcal{C}} = \frac{n-k}{n \cdot H(\mathbf{X}|\mathbf{Y})} = \frac{1-R}{H(\mathbf{X}|\mathbf{Y})} \tag{5.5}$$

In some scenarios Alice's and Bob's strings can be regarded as the input and output of a BSC, characterized by the crossover probability ε. For instance, most QKD protocols encode the information in discrete binary variables [7, 10], although there are many proposals on continuous variable protocols [94, 46, 37]. Errors on the quantum channel are normally uncorrelated and symmetric or, if prior to the reconciliation Alice and Bob apply a random permutation, they can behave as such [44]. This is the case we will consider here. For this reason, $\mathbf{X}$ and $\mathbf{Y}$ can be seen, respectively, as the input and the output of a BSC. In a typical implementation of a QKD protocol, Alice and Bob have access to the channel characteristics. In particular, the crossover probability ε of the BSC is supposed to be known by the legitimate parties. Then, the efficiency parameter f can be described as the relationship between the length of the conversation and the optimal value $n \cdot H(\mathbf{X}|\mathbf{Y}) = n \cdot H(\varepsilon, 1-\varepsilon)$:

$$f_{\text{BSC}}(\varepsilon) = \frac{1-R}{H(\varepsilon, 1-\varepsilon)} \tag{5.6}$$

5.3 PREVIOUS WORK

5.3.1 *First protocol*

The first protocol for information reconciliation in the context of QKD was proposed by Bennett et al. [8, 9]. The objective of [9] is to discuss how to use a channel with perfect authenticity but no privacy to repair a channel with imperfect privacy but no authenticity. The imperfections of the privacy channel can be of any kind but are attributed to an eavesdropper, Bennett et al. propose a full secret key distillation protocol composed by an error detection step followed by an information reconciliation procedure and ending with a privacy amplification step.

For error detection Alice selects a function $f : \{0,1\}^n \to \{0,1\}^k$ where f belongs to a universal family of hash functions. Alice sends to Bob $f(\mathbf{x})$ and a description of f. If $f(\mathbf{x}) = f(\mathbf{y})$ Alice and Bob can assume that the strings coincide with an error probability bounded from above $p_e \leqslant 2^{-k}$. For the information reconciliation step they propose several

alternatives depending on the noise in the private channel. Then a privacy amplification step with universal hash functions creates a secret key.

If just a few errors are thought to have occurred Bob can try to compute $f(\mathbf{z})$ for all $\mathbf{z}$ such that $d(\mathbf{y}, \mathbf{z}) \leqslant t$, in other words, Bob can try to compute $f(\mathbf{z})$ for all $\mathbf{z}$ at distance at most t from $\mathbf{y}$. This first procedure is called bit twiddling, and except if there are a very small number of errors rapidly becomes unfeasible.

If more errors are thought to have occurred then error detection can be postponed to information reconciliation.

Alice applies an error correcting code in systematic form $\mathcal{C}$ to $\mathbf{x}$ and sends only the the redundancy through the public channel to Bob.

The protocol described is generic and can be applied with any linear error correcting code in systematic form. For low error probability ($\varepsilon \leqslant 0.01$) the authors propose the use of Hamming codes [71] complemented later, in the postponed error detection step, with bit twiddling. If $\varepsilon \geqslant 0.01$ a convolutional code [71] could behave better since the decoding effort escalates better than with Hamming codes and MAP decoding. This first protocol achieves a efficiency that was approximated by the following function in [9]:

$$f = \frac{\log(1 + 2\sqrt{p(1-p)})}{h(p)} \tag{5.7}$$

Table 5.1: Encoding rate and efficiency of the protocol in [9].

ε	R	R_{opt}	f
0.001	0.9116	0.9886	7.75
0.010	0.7382	0.9192	3.24
0.030	0.5765	0.8056	2.18
0.050	0.4781	0.7136	1.82
0.100	0.3219	0.5310	1.45
0.250	0.1000	0.1887	1.11
0.400	0.0146	0.0290	1.01

5.3.2 *The primitives*

In this section we review three primitives that serve as building blocks for the following information reconciliation protocols [88]. The primitives are distributed algorithms that allow Alice and Bob to perform a simple task on their strings, we introduce the primitives Dichot, Parity and Confirm.

The Parity primitive is used by Alice and Bob to compare the parity of a specific subset of their strings. Parity is used by all the rest of the primitives.

Algorithm 2 The Parity($\mathbf{a}$,$\mathbf{b}$,π,n_1,n_2) primitive

Require: $|\mathbf{a}| = |\mathbf{b}|$ and $|\mathbf{a}| > 0$
  $\hat{\mathbf{a}} \leftarrow \pi(\mathbf{a})$
  $\hat{\mathbf{b}} \leftarrow \pi(\mathbf{b})$
  Alice calculates $p_A \leftarrow \sum_{i=n_1}^{n_2} \hat{\mathbf{a}}[i]$
  Alice sends p_A to Bob
  Bob calculates $p_B \leftarrow \sum_{i=n_1}^{n_2} \hat{\mathbf{b}}[i]$
  Bob sends p_B to Alice
  return $p_A + p_B$

The Confirm primitive is used by Alice and Bob to check if their strings differ. Alice and Bob choose a random subset of their strings and check if the parity of the subset coincides. If the strings differ Alice and Bob find a mismatch in the parity with probability 1/2.

Algorithm 3 The Confirm($\mathbf{a}$,$\mathbf{b}$) primitive

Require: $|\mathbf{a}| = |\mathbf{b}|$ and $|\mathbf{a}| > 0$
  Alice and Bob choose a random subset of their chains given by the first $\lceil |\mathbf{a}|/2 \rceil$ of permutation π
  $p \leftarrow$Parity($\mathbf{a}$,$\mathbf{b}$,π,1,$\lceil |\mathbf{a}|/2 \rceil$)
  return p, π

Dichot is a recursive binary search algorithm that allows Alice and Bob to find an error if there is an odd number of errors. Given two strings Dichot compares the parity of the first half, if it coincides if performs Dichot on the second half, else it performs Dichot on the first, see Fig. 5.1. Note that if there is an even number of errors there is no guarantee on the behavior of Dichot.

5.3.3 *The BBBSS protocol*

A second proposal by Bennett et al [10] is embedded in a full QKD protocol description. The information reconciliation step in this protocol, which we shall call Bennett, Bessette, Brassard, Salvail and Smolin's Information Reconciliation Protocol (BBBSS) following the names of the authors, exploits the public channel's interactivity and improves the efficiency of the first proposal. We describe the whole procedure in Alg. 5.

It consists in a multi-pass procedure. On each pass Alice and Bob agree on a random permutation for their strings, then divide the strings in blocks of length k_i. k_i was empirically found such that in pass i a block of length k_i is unlikely of having more than one

Algorithm 4 The Dichot($\mathbf{a}$,$\mathbf{b}$,π,n_1,n_2) primitive

Require: $|\mathbf{a}| = |\mathbf{b}|$ and $|\mathbf{a}| > 0$
  if $n_1 = n_2$ **then**
    Alice sends $\mathbf{a}$ and Bob sets $\mathbf{b} \leftarrow \mathbf{a}$
    return $\pi(n_1)^{-1}$
  else
    $p \leftarrow$Parity($\mathbf{a}$,$\mathbf{b}$,π,n_1,$\lfloor \frac{n_1+n_2}{2} \rfloor$)
    if $p \neq 0$ **then**
      return Dichot($\mathbf{a}$,$\mathbf{b}$,π,n_1,$\lfloor \frac{n_1+n_2}{2} \rfloor$
    else
      return Dichot($\mathbf{a}$,$\mathbf{b}$,π,$\lfloor \frac{n_1+n_2}{2} \rfloor + 1$,$n_2$)
    end if
  end if

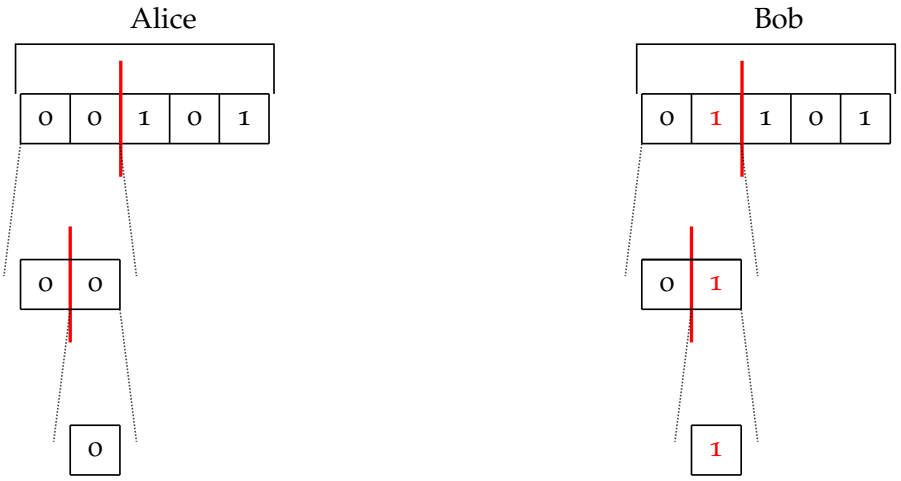

Figure 5.1: Example of Dichot on blocks with mismatching parities. In this example $\mathbf{a} = \{01100\}$ and $\mathbf{b} = \{01101\}$. For simplicity we omit the permutation and the substring indexes. There is an odd number of errors. Alice and Bob run Dichot($\mathbf{a}$,$\mathbf{b}$). Dichot divides the chains in two halves and checks the parities of the first two halves: Parity($\{011\}$,$\{011\}$). As the parities coincide it recursively runs Dichot($\{00\}$,$\{01\}$). In this second run of Dichot it again divides the chains in two halves and checks the parities of the first two halves: Parity($\{0\}$,$\{0\}$). Again the parities coincide and Dichot is called a third time: Dichot($\{0\}$,$\{1\}$). This time with the strings of length one Alice and Bob find the error in the fifth bit.

.

Algorithm 5 The BBBSS($\mathbf{x}$,$\mathbf{y}$,p_{diff}) protocol

Require: $|\mathbf{x}| = |\mathbf{y}|$ and $|\mathbf{x}| > 0$
  Set $\mathbf{a_0} \leftarrow \mathbf{x}$
  Set $\mathbf{b_0} \leftarrow \mathbf{y}$
  for $i = 0 \rightarrow 2$ **do**
    Alice and Bob choose a random permutation function π_i
    $k_i = f_{BBBSS}(p_{diff})$
    for $l = 0 \rightarrow \lceil n/k_i \rceil$ **do**
      $n_1 \leftarrow lk_i$
      $n_2 \leftarrow min((l+1)k_i, n)$
      Alice and Bob set $p \leftarrow$Parity($\mathbf{a_i}$,$\mathbf{b_i}$,π_i,n_1,n_2)
      if $p \neq 0$ **then**
        Alice and Bob execute Dichot($\mathbf{a_i}$,$\mathbf{b_i}$,π_i,n_1,n_2)
      end if
    end for
    Alice and Bob construct $\mathbf{c_i}$ with one bit from every block created in step i
    $\mathbf{a_{i+1}} \leftarrow \mathbf{a_i} - \mathbf{c_i}$
    $\mathbf{b_{i+1}} \leftarrow \mathbf{b_i} - \mathbf{c_i}$
    if $|\mathbf{a_{i+1}}| = 0$ **then**
      END
    end if
  end for
  Set $j \leftarrow 0$
  while $j \leqslant 20$ **do**
    Alice and Bob set $p, \pi_i \leftarrow$Confirm($\mathbf{a_i}$,$\mathbf{b_i}$)
    if $p \neq 0$ **then**
      Alice and Bob execute Dichot($\mathbf{a_i}$,$\mathbf{b_i}$,π_i,1,$\lceil|\mathbf{a}|/2\rceil$)
      $j \leftarrow 0$
    end if
    Alice and Bob construct $\mathbf{c_i}$ with one bit from every block created in step i
    $\mathbf{a_{i+1}} \leftarrow \mathbf{a_i} - \mathbf{c_i}$
    $\mathbf{b_{i+1}} \leftarrow \mathbf{b_i} - \mathbf{c_i}$
    if $|\mathbf{a_{i+1}}| = 0$ **then**
      END
    end if
    $i \leftarrow i + 1$
    $j \leftarrow j + 1$
  end while

erroneous bit. They compare the parities of each block and for those blocks with different parity they perform a binary search (see Alg. 4) exchanging additionally $\log k$ bits.

Alice and Bob discard a bit from every block. Several passes are performed with increasing block length until most errors are removed.

Then a new strategy is applied on each pass: Alice and Bob compute the parity of a random substring, performing also a Dichotomic search whenever it differs. This second strategy corresponds with applying the Confirm primitive (see Alg. 3). This procedure can be executed several times, if Confirm is executed s consecutive times without finding an error the probability that there are still errors in Bob's chain is $1/2^s$.

The efficiency of the algorithm depends on choosing an appropriate size of block. If the size is very small, most blocks are errorless and unnecessary parities are exchanged between Alice and Bob. On the other hand if the block size is too big in many blocks there will be an even number of errors and they will remain undetected. In [10] k_i is optimized empirically to a value that can be approximated by the function $k_0 = 0.55/p$, where p is the crossover probability and $k_i = \lceil 1.4k_{i-1} \rceil$. The scheme was refined in [131], k_i was analytically found such that the number of parities exchanged to remove an error on each pass are minimized. We reproduce their analysis.

Let the probability of detecting an error in a block be p_{odd} which is the same probability as having an odd number of errors in the block, we can calculate exactly p_{odd} as:

$$p_{odd} = \frac{1 - (1 - 2p)^{k_i}}{2} \tag{5.8}$$

this follows from Lem. 9 if $\forall i, p_i = p$.

Until pass i, z errors have been found and t bits have been discarded, then the error probability on the remaining chain is:

$$p_i = \frac{np - z}{n - t} \tag{5.9}$$

Now the average number of disclosed parities t_i and corrected errors z_i during pass i is:

$$t_i = \frac{n-t}{k_i} + p_{odd}\frac{n-t}{k_i}\log p_i = \frac{n-t}{k_i}(1 + p_{odd}\log p_i) \tag{5.10}$$

$$z_i = p_{odd}\frac{n-t}{k_i} \tag{5.11}$$

The relation between t_i and z_i gives the average number of parities that need to be exchanged in order to correct one error:

$$\frac{t_i}{z_i} = \frac{\frac{n-t}{k_i}(1 + p_{odd} \log p_i)}{p_{odd} \frac{n-t}{k_i}} \tag{5.12}$$

which plugging in p_{odd} becomes:

$$\frac{t_i}{z_i} = \log p_i + \frac{2}{1-(1-2p)^{k_i}} \tag{5.13}$$

there is no known formula for minimizing Eq. 5.13 but it can be easily optimized numerically.

Algorithm 6 The Yamazaki($\mathbf{x}$,$\mathbf{y}$,p_{diff}) protocol

Require: $|\mathbf{x}| = |\mathbf{y}|$ and $|\mathbf{x}| > 0$
  $\mathbf{a_0} \leftarrow \mathbf{x}$
  $\mathbf{b_0} \leftarrow \mathbf{y}$
  $i \leftarrow 0$
  $j \leftarrow 0$
  while $j \leqslant 11$ **do**
    Alice and Bob choose a random permutation function π_i
    $k_i = f_{Yamazaki}(p_{diff})$
    for $l = 0 \rightarrow \lceil n/k_i \rceil$ **do**
      $n_1 \leftarrow lk_i$
      $n_2 \leftarrow \min((l+1)k_i, n)$
      $p \leftarrow$Parity($\mathbf{a}$,$\mathbf{b}$,π_i,n_1,n_2)
      if $p \neq 0$ **then**
        Dichot($\mathbf{a}$,$\mathbf{b}$,π_i,n_1,n_2)
        $j \leftarrow 0$
      end if
    end for
    Alice and Bob construct $\mathbf{c_i}$ with one bit from every block created in step i
    $\mathbf{a_{i+1}} \leftarrow \mathbf{a_i} - \mathbf{c_i}$
    $\mathbf{b_{i+1}} \leftarrow \mathbf{b_i} - \mathbf{c_i}$
    if $|\mathbf{a_{i+1}}| = 0$ **then**
      END
    end if
    $i \leftarrow i + 1$
    $j \leftarrow j + 1$
  end while

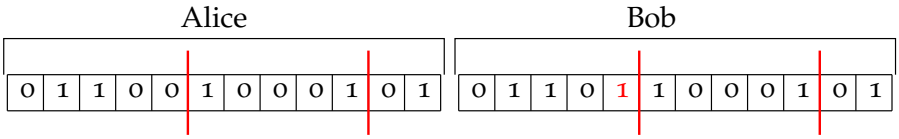

Figure 5.2: Cascade division in blocks

5.3.4 *The Cascade protocol*

As mentioned in the introduction, the most widely used protocol for error correction in the QKD context is Cascade. Proposed by Brassard and Salvail in their seminal paper "Secret key reconciliation by public discussion" [14] , this protocol is an evolution from BBBSS.

Cascade runs for a fixed number of passes. As in BBBSS, in each pass, Alice and Bob divide their strings $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ and $\mathbf{y} = \{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}\}$ into blocks of equal length k_i. If we let $\mathbf{x}_i^l$ and $\mathbf{y}_i^l$ stand for Alice and Bob's l-th block in pass i:

$$K_i^l = K_i[lk_i, \min((l+1)k_i, n)] \tag{5.14}$$

The initial block length depends on the estimated error probability, p, and it is doubled when starting a new pass. For each block they compute Parity. If there is a parity mismatch it implies an odd number of errors, and Alice and Bob compute Dichot on the block. The first pass of Cascade is exactly the same as the first pass of BBBSS except that in Cascade no bits are discarded this difference allows to correct more bits in the following passes.

Whenever an error is found after the first pass, it uncovers an odd number of errors masked on the preceding passes and the algorithm returns to correct those errors previously undetected. The position p where an error has been found belonged to different blocks in the preceding passes. Alice and Bob apply the primitive Cascor to find errors in these blocks in an optimized fashion (see Alg. 8). Let $\mathcal{C}$ be the set of such blocks with an odd number of errors.

$$\mathcal{C} = \{K_i^l | p \in [lk_i, \min((l+1)k_i, n)]\} \tag{5.15}$$

Alice and Bob can now choose the smallest block in $\mathcal{C}$ and perform a binary search to find and correct another error. This new error will imply adding or removing blocks from $\mathcal{C}$. The process continues until $\mathcal{C}$ is emptied. This cascading process gives name to the protocol.

The value of the initial block size k_1 is a critical parameter. In [14] a numerical procedure is derived to choose k_1 such that the probability that there are errors in a block from pass 1 K_1^l is exponentially reduced with the number of passes. In [20] this numerical procedure is approximated as $k_1 \approx 0.73/e$, e being the estimated error probability.

Yamazaki et al. propose in two different papers [130, 117] improvements on the analysis of the initial block size that allow to improve the reconciliation efficiency. We review their proposals, following also the ideas in [63], as the last main contributions to the family of protocols initiated with BBBSS. Their proposal in [117] follows the observation that most errors are detected in the first two passes of Cascade and approximately half of them are corrected on each of both first passes. From this hypothesis it is possible to minimize the parities exchanged by optimizing k_1 and k_2. Now, if half the errors are corrected on the first pass the number of parities exchanged in that pass are:

$$L_1 = \frac{n}{k_1} + \frac{np}{2} \log k_1 \tag{5.16}$$

The remaining half of the errors are corrected in pass 2. In this pass every detected error reveals a second error undetected in pass 1. In consequence the errors are corrected in pairs and an error in pass 2 implies exchanging $\log k_2$ parities while an error detected in a block from pass 1 implies exchanging $\log k_1$ parities. The expected number of parities exchanged in pass 2 follows:

$$L_2 = \frac{n}{k_2} + \frac{np}{4} [\log k_1 + \log k_2] \tag{5.17}$$

With this conditions the minimization of the parities gives a closed formula for the optimal lengths in passes 1 and 2:

$$k_1 = \lfloor \frac{4 \ln 2}{3p} \rfloor \tag{5.18}$$

$$k_2 = \lfloor \frac{4 \ln 2}{p} \rfloor \tag{5.19}$$

Later, Yamazaki et al. improve their estimation by calculating the average number of errors that are corrected in pass 1 and then estimating the errors corrected in pass 2 as its complementary. In particular if a block in pass 1 has an odd number of errors with probability p_{odd}, then the expected number of corrected errors in pass 1 and pass 2 T_1 and T_2, respectively, are:

$$T_1 = \frac{n}{k_1} p_{odd} \tag{5.20}$$

$$T_2 = np - T_1 = np - \frac{n}{k_1} p_{odd} \tag{5.21}$$

The expression of the exchanged parities varies slightly:

$$L_1 = \frac{n}{k_1} + \frac{np_{odd}}{k_1} \log k_1 \tag{5.22}$$

$$L_2 = \frac{n}{k_2} + \frac{1}{2}\left[np - \frac{n}{k_1}p_{odd}\right][\log k_1 + \log k_2] \tag{5.23}$$

There is no closed formula that minimizes the sum $L_1 + L_2$, however it is easy to minimize the sum numerically and obtain optimal values for the lengths k_1 and k_2.

Algorithm 7 The Cascade(**x**,**y**) protocol

Require: $|\mathbf{x}| = |\mathbf{y}|$ and $|\mathbf{x}| > 0$
  Set $\mathbf{a_0} \leftarrow \mathbf{x}$
  Set $\mathbf{b_0} \leftarrow \mathbf{y}$
  for $i = 1 \rightarrow 3$ **do**
    Alice and Bob choose a random permutation function π_i
    $k_i = f_{Cascade}(p_{diff})$
    for $l = 0 \rightarrow \lceil n/k_i \rceil$ **do**
      $n_1 \leftarrow lk_i$
      $n_2 \leftarrow \min((l+1)k_i, n)$
      $p \leftarrow$Parity(**a**,**b**,π_i,n_1,n_2)
      if $p \neq 0$ **then**
        e=Dichot(**a**,**b**,π_i,n_1,n_2)
        if $i \geqslant 1$ **then**
          Cascor(**a**,**b**,i,e,π_1,π_2,...,π_i)
        end if
      end if
    end for
  end for

Algorithm 8 The Cascor(**x**,**y**,i,e,π_1,π_2,...,π_i) protocol

Require: $|\mathbf{x}| = |\mathbf{y}|$ and $|\mathbf{x}| > 0$
  Construct $\mathcal{C}$ the set of all blocks that contain the bit e. $C = \{\mathbf{a_i^l} | \mathbf{a}[e] \in \mathbf{a_i^l}\}$
  while $C \neq \emptyset$ **do**
    $l = \min_{i'} \mathbf{a_{i'}^l}, \in \mathcal{C}$
    e′=Dichot($\mathbf{a_{i'}^l}$,,$\mathbf{b_{i'}^l}$,)
    Update $\mathcal{C}$ with all blocks that contain the bit e'
  end while

It should be noted that Cascade is highly interactive even when carefully implemented. Since many exchanges between Alice and Bob are required to reconcile a string, the time overhead for these communications can severely limit the achievable key generation rate. This

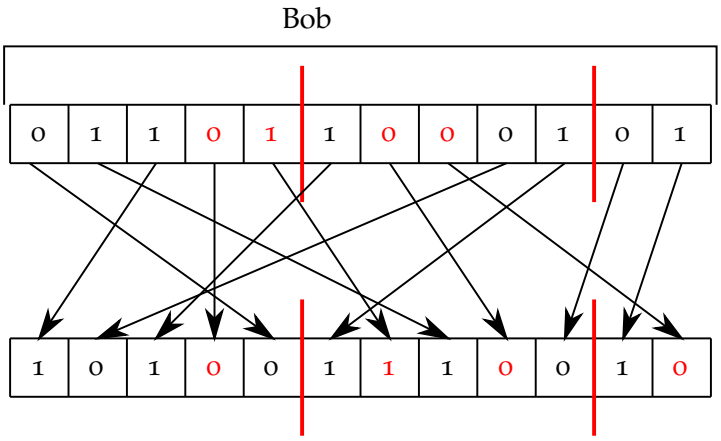

Figure 5.3: Discovering an error uncovers hidden errors in the preceding steps.

could for instance be the case in free space QKD implemented between a satellite and a base station and even more when the communication between Alice and Bob is performed over a network connection with a high latency.

Despite this limitation, Cascade is certainly the most widely used reconciliation protocol in practical discrete variables QKD setups. One of its interests is its relative simplicity and the fact that it performs reasonably well in terms of efficiency. As we shall see, most of the alternative solutions developed after Cascade have focused on reducing the level of interactivity, usually at the expense of reconciliation efficiency. This is the reason why we have used Cascade as the essential element of comparison with the solutions that we develop in the next chapter.

5.4 OTHER WORK ON INFORMATION RECONCILIATION PROTOCOLS

Many variations around the principle of interactive reconciliation used in Cascade have been proposed, in order to limit the interactivity. Among the most notable works, we can cite the Winnow protocol [15]. Like Cascade, Winnow splits the binary strings to be reconciled into blocks but instead of correcting errors by iterative binary search, the error correction is based on the Hamming code. Winnow's interest lies in the reduction of the amount of required communication to three messages per iteration [121]. In the first communication called the parity test step Alice and Bob exchange the parities of every block. After that, they exchange the syndrome of a Hamming code for correcting single errors in every block with a parity mismatch. The protocol incorporates a privacy maintenance procedure by discarding one bit per parity revealed (i.e. m bits are discarded when a syndrome of length m is exchanged).

Winnow is thus significantly faster that Cascade but unfortunately, its efficiency is lower for error rates below 10 %, i.e. in the parameter

range useful for practical QKD. Recently, some interesting improvements have been proposed for selecting an optimum block length in this protocol [51].

Another interesting development has been conducted by Liu [64] who has proposed a protocol that optimizes the information exchanged per corrected bit. Liu's protocol is in essence very similar to Cascade. Its objective is to minimize the information sent on the public channel to correct one error during a pass. This protocol however remains highly interactive.

Some QKD protocols provide Alice and Bob with correlated continuous random variables and specific work on key reconciliation has been conducted in this context, beginning with the work on Sliced Error Correction [122] used to convert continuous variables into binary strings. It is also mainly in the context of continuous variables that modern coding techniques have been used within information reconciliation protocols: turbo codes in [85, 121] and LDPC codes in [13, 61].

5.5 LDPC

In contrast with continuous-variable information reconciliation, not much has been done to adapt modern coding techniques to the discrete case. Forward error correction has the advantage of being very well known and even attaining the theoretical limit for some channels [102]. Also, and of great importance for SKD, it requires a single message, namely the syndrome of $\mathbf{X}$ for the code being used, to correct the discrepancies. Relevant references are Watanabe et al. [125] who proposed using LDPC codes for their information reconciliation procedure, BBN Niagara [34] and the work for free space QKD by Duligall et al. [25], all of which use LDPC codes. However [34] only provides a single point comparing the performance of LDPC codes and Cascade, showing a net decrease of the communication overhead but a slightly decrease in the efficiency while [25] does not provide any information on the results of their use of LDPC codes.

Let us now we consider the experimental performances of using a set of LDPC codes developed for the BSC. LDPC codes are decoded with the belief propagation algorithm [38]. We have considered the efficiency of reconciliation for $\varepsilon = 10^{-3}$, that is, the remaining frame error probability is below $\cdot 10^{-3}$, though the remaining errors could be handled very efficiently by concatenation with a Bose, Ray-Chaudhuri and Hocquenghem codes (BCH) code of very high rate (typically 0.998 [2]).

It is worth noting that classical error-correcting codes were explicitly considered and said to be inadequate for information reconciliation in many of the first works [8, 9, 10, 20]. These limitations were consistent with the computational resources available at the time and

explain why alternative methods were considered for information reconciliation.

As explained in Section 5.1 the performance of a reconciliation protocol can be evaluated by measuring the amount of information disclosed in this process. For chains modeled as the input and output of a BSC with a crossover probability p, an ideal reconciliation protocol would reveal a fraction $h(p)$ while a real protocol reveals $f(p)h(p)$.

We have represented the reconciliation efficiency $f(p)$ on Fig. 5.4 for Cascade and for a set of LDPC codes. The results that we have found with Cascade are very similar to those of Crepeau [20] or Brassard and Salvail [14]: Cascade performs well at low bit error rates where its efficiency differs only by 10% from the Shannon limit of 1. However, its efficiency decreases gradually as the crossover probability increases.

A quick observation reveals that, in contrast with Cascade, the reconciliation efficiency $f(p)$ exhibits a saw behavior when our set of LDPC codes is used. The reason for this is that we have chosen a discrete number of codes. As each code has a threshold, a string with a measured error probability p will be corrected with the code having the smallest threshold greater than p. The saw effect will be reduced as the number of LDPC codes used is increased.

As we can see on this figure, optimized LDPC codes can perform better than Cascade as soon as the error rate is above 4%. With our discrete set of LDPC codes, the performances are always better than Cascade when the error rate is above 5%. This gain of performance can significantly impact on the achievable secret key generation rate in practical QKD.

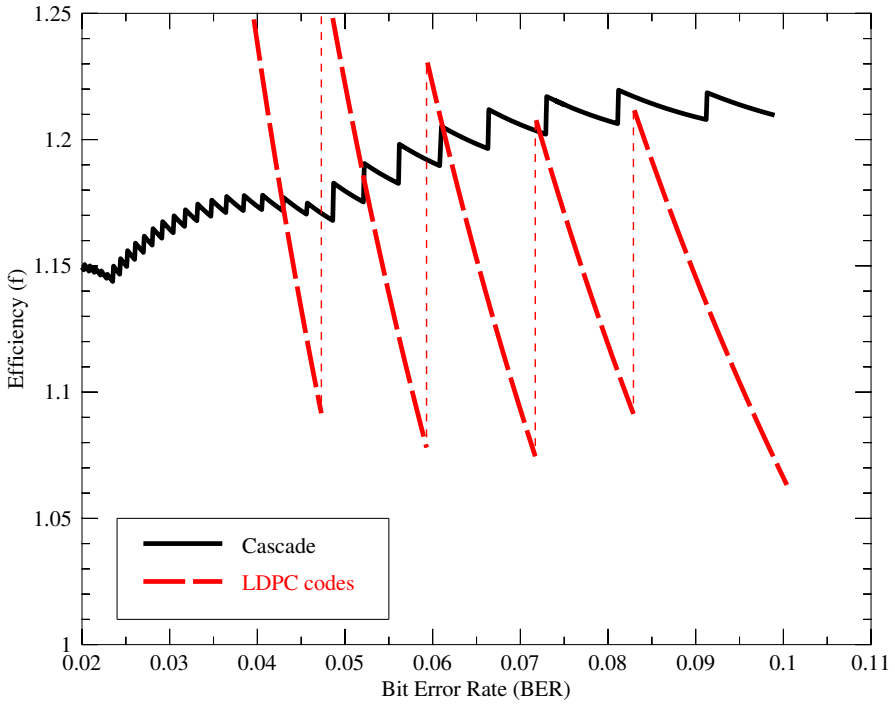

Figure 5.4: Reconciliation Efficiency $f(p)$ achieved using our discrete set of LDPC codes described in Table 3.2 compared to Cascade.

6

RATE ADAPTIVE INFORMATION RECONCILIATION

We wish to change the code rate, i.e., the number of check bits, and hence the correction power of the code during transmission of an information frame according to source and channel needs. For practical purposes, we would like to have not just switching between a set of encoders and decoders, but one encoder and one decoder which can be modified without changing their basic structure.

— Joachim Hagenauer [49]

6.1 INTRODUCTION

Although linear codes are a good solution for the reconciliation problem, since they can be tailored to a given error rate, their efficiency degrades when it is not known beforehand. This is the case in QKD, where the error rate is an a priori unknown that is estimated for every exchange. The Quantum Bit Error Rate (QBER) might vary significantly in two consecutive key exchanges, specially when the quantum channel is transported through a shared optical fibre that can be used together with several independent classical or quantum channels that can add noise. To address this problem there are two different options: (i) it is possible to build a code once the error rate has been estimated, and (ii) a pre-built code can be modified to adjust its information rate. The computational overhead would make the first option almost unfeasible except for very stable quantum channels, something difficult to achieve in practice and impossible in the case of a shared quantum channel in a reconfigurable network environment [60]. We proceed to describe the use of the second strategy as the easiest and most effective way to obtain a code for the required rate, for which we describe a protocol that adapts pre-built codes in real time while maintaining an efficiency close to the optimal value.

We introduced this one-way reconciliation technique in [31, 29] and analyzed its security in [30]. The protocol is protected by the patent ES 2389217 B2 [33]. Outside the scope of this thesis, a new two-way protocol exploiting these ideas was developped in [74, 76].

6.2 RATE MODULATION

Puncturing and shortening, described in Sec. 3.4 are two common strategies used to adapt the rate of a linear code. These techniques may be regarded as the transmission of different parts of the codeword over different channels (see Fig. 6.1).

Since puncturing is a process by which p codeword symbols are eliminated, it can be seen as a transmission over a BEC with erasure probability of 1, $\mathrm{BEC}(1)$, or it also can be regarded as a transmission over a BSC with maximum error $\mathrm{BSC}(0.5)$. It should be no surprise that both extreme channels have the same capacity:

$$C_{BSC}(0.5) = C_{BEC}(1) = 0 \tag{6.1}$$

Shortening is a process by which s codeword symbols are known with absolute certainty, as such it can be seen as a transmission over a BEC with erasure probability of 0, $\mathrm{BEC}(0)$, or in the same fashion it can be presented as a transmission over a noiseless BSC $\mathrm{BSC}(0)$. Again both capacities coincide:

$$C_{BSC}(0) = C_{BEC}(0) = 1 \tag{6.2}$$

The remaining symbols are transmitted by the real channel which in the present paper can be modeled by a binary symmetric channel with crossover probability ε, $\mathrm{BSC}(\varepsilon)$

Supposing that R_0 is the original coding rate, the modulated rate is then calculated as:

$$R = \frac{R_0 - \sigma}{1 - \pi - \sigma} = \frac{k - s}{n - p - s} \tag{6.3}$$

where π and σ represent the ratios of information punctured and shortened respectively.

Both strategies, puncturing and shortening, can be applied simultaneously. Given a $\mathcal{C}(n, k)$ code and $n' \leqslant n$ bits, if puncturing and shortening are applied with a constant number d of punctured and shortened symbols, a single code can be used to protect the n' bits for different error rates. There are two consequences of applying a constant d: (i) there is a limit to the minimum and maximum achievable information rates. These limits, expressed as a function of $\delta = d/n$, define the correction interval:

$$0 \leqslant R_{min} = \frac{R_0 - \delta}{1 - \delta} \leqslant R \leqslant \frac{R_0}{1 - \delta} = R_{max} \leqslant 1 \tag{6.4}$$

(ii) puncturing and shortening procedures cause an efficiency loss [47]. Therefore, there is a trade-off between the achievable information rates

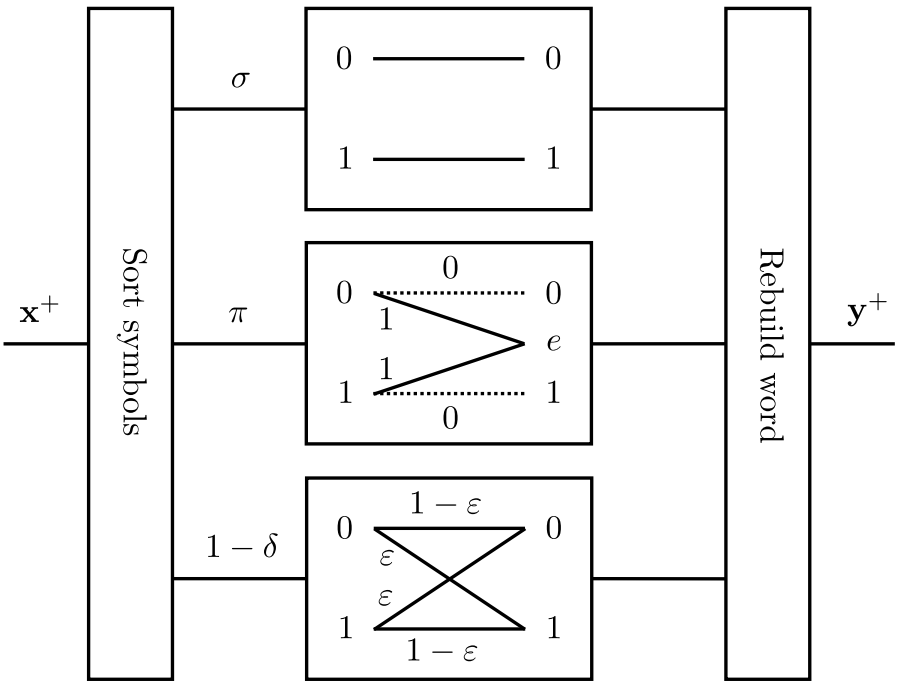

Figure 6.1: Puncturing and shortening on a LDPC code results in the division of the original binary symmetric channel used to reconcile Alice's **x** string with Bob's **y** into three different channels: a binary erasure channel with erasure probability of 1 (for the fraction π of punctured symbols), a BEC with erasure probability of 0 (for the fraction σ of shortened symbols) and a binary symmetric channel with crossover probability ε (for the rest of the symbols).

and reconciliation efficiency. One way to compensate both constraints is to use multiple codes to define different correction intervals as shown in the next section, Section 6.3.

This efficiency loss, caused by high levels of puncturing and shortening, can be avoided if a set of n codes $\mathcal{C}_i$ with different information rates is used: $R_0(\mathcal{C}_1) \leqslant R_0(\mathcal{C}_2) \leqslant R_0(\mathcal{C}_n)$. The target error range can then be partitioned into, $[R_{min}(\mathcal{C}_1), R_{max}(\mathcal{C}_1)] \cup [R_{min}(\mathcal{C}_2), R_{max}(\mathcal{C}_2)] \cup \ldots \cup [R_{min}(\mathcal{C}_n), R_{max}(\mathcal{C}_n)]$, not necessarily with the same size. The number of intervals depends on the width of the error rate range to cover and on the desired efficiency. The compromise between the width of the interval covered and the achieved efficiency in the one code case is transferred to a compromise between efficiency and the added complexity of managing several codes. We can study this effect with the DDE described in Sec. 3.2.3. To take into account puncturing and shortening we can modify the initial density (see Eq. 3.52):

$$f_r(p) = (1-\delta) f_{r_{BSC}}(p) + \pi\delta_0(x) + \sigma\delta_\infty(x) \tag{6.5}$$

Fig. 6.2 shows the computed efficiency thresholds for several families of codes with different coding rates. It can be observed how different values of δ offer a trade-off between the covered range of rates and the achieved efficiency.

6.3 PROTOCOL

We now proceed to describe a rate-compatible information reconciliation protocol using puncturing and shortening techniques as described above.

Step 0: Raw key exchange. Alice and Bob obtain a raw key. The key exchange may be modeled as follows. Alice sends to Bob the string $\mathbf{x}$, an instance of a random variable $\mathbf{X}$, of length $\ell = n - d$ through a BSC with crossover probability ε. Bob receives the correlated string, $\mathbf{y}$, but with discrepancies to be removed in the following steps.

Step 1: Pre-conditions. Prior to the key reconciliation process Alice and Bob agree on the following parameters: (i) a pool of shared codes of length n, constructed for different coding rates; (ii) the size of the sample, t, that will be used to estimate the error rate in the communication; and (iii) the maximum number of symbols that will be punctured or shortened to adapt the coding rate, $d = p + s = n\delta$.

Step 2: Error rate estimation. Bob chooses randomly a sample of t bits of $\mathbf{y}$, $\alpha(\mathbf{y})$, and sends them and their positions, $\beta(\mathbf{y})$, to Alice through a noiseless channel. Using the positions received from Bob, $\beta(\mathbf{y})$, Alice extracts an equivalent sample in $\mathbf{x}$, $\alpha(\mathbf{x})$, and estimates the

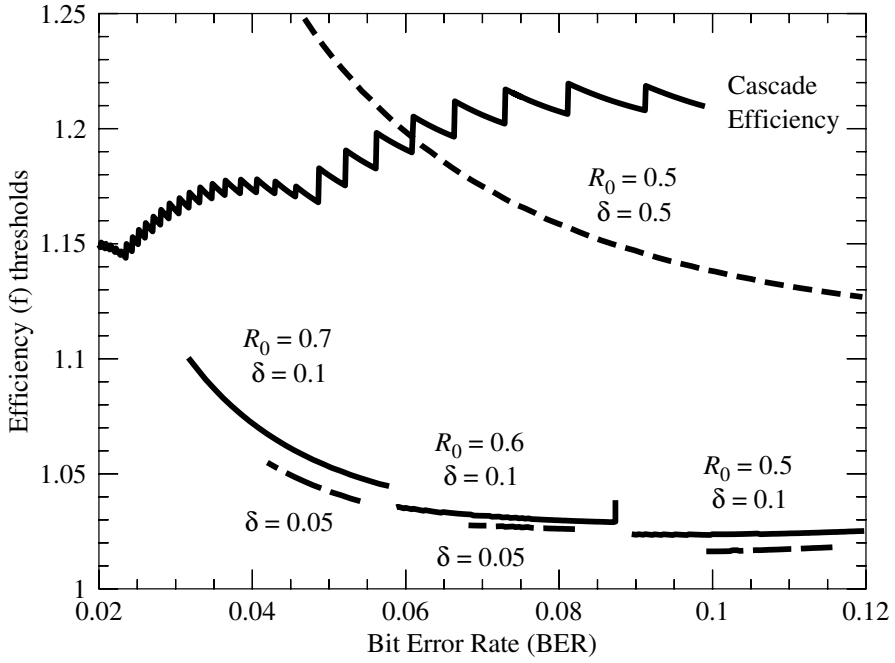

Figure 6.2: Efficiency thresholds for different codes with information rates, $R_0 = 0.5$, 0.6 and 0.7. Two δ values, 0.1 (solid line) and 0.05 (dashed) have been used to adapt the rate for each code. As a comparison, a single code covering all of the range of interest, with rate $R_0 = 0.5$ and $\delta = 0.5$, is presented to show how the efficiency degrades for high δ values, although a broader range is covered. The codes have been optimized using the density evolution algorithm for the BSC. The Cascade efficiency was calculated using the same sample size (2×10^5). The block size used in the first step, k_1, is given by $k_1 = \lceil 0.73/\varepsilon \rceil$ (optimized in [20]) and doubled in every subsequent step $k_n = 2k_{n-1}$. The sawtooth behavior of the Cascade efficiency reflects the points where k_1 changes.

crossover probability for the exchanged key by comparing the two samples:

$$\varepsilon' = \frac{\alpha(\mathbf{x}) + \alpha(\mathbf{y})}{t} \tag{6.6}$$

Once Alice has estimated ε', she knows the theoretical rate for a punctured and shortened code able to correct the string. Now she computes the optimal rate corresponding to the efficiency of the code she is using: $R = 1 - f(\varepsilon')h(\varepsilon')$; where h is the binary Shannon entropy function and f the efficiency. Then she can derive the optimal values for puncturing and shortening, p and s respectively, as:

$$\begin{aligned} s &= \lceil (R_0 - R(1 - d/n)) \cdot n \rceil \\ p &= d - s \end{aligned} \tag{6.7}$$

Step 3: Coding. Alice creates a string $\mathbf{x}^+ = g(\mathbf{x}, \sigma_{\varepsilon'}, \pi_{\varepsilon'})$ of size n. The function g defines the $n - d$ positions to take the values of string $\mathbf{x}$, the p positions to be assigned random values, and the s positions to have values known by Alice and Bob. She then sends $s(\mathbf{x}^+)$, the syndrome of $\mathbf{x}^+$, to Bob as well as the estimated crossover probability ε'.

This process can be regarded as jointly coding (and decoding) the original strings sent through a BSC(ε) with p bits sent through a BEC with erasure probability 1, and s bits sent through a noiseless channel (see Fig. 6.1).

Step 4: Decoding. Bob can reproduce Alice's estimation of the optimal rate R, the positions of the p punctured bits, and the positions and values of the s shortened bits. Bob then creates the corresponding string $\mathbf{y}^+ = g(\mathbf{y}, \sigma_{\varepsilon'}, \pi_{\varepsilon'})$. He should now be able to decode Alice's codeword with high probability, as the rate has been adapted to the channel crossover probability. Bob sends an acknowledgment to Alice to indicate if he successfully recovered $\mathbf{x}^+$.

6.4 SECURITY

The security of sp-protocols is addressed in this section. We demonstrate that the use of an sp-protocol does not impose any constraint on the achievable secret key rate. Moreover, from this demonstration it is possible to infer that the quality of the information reconciliation procedure depends only on the quality of the error correction code. We begin with the proof of a lemma (Lem. 12, 13, 14, 15) that allows to exploit the random construction of the punctured and shortened bits in the proposed protocol. Then, we consider the security of the protocol for the four different SKD scenarios discussed in Ch. 4.

Lemma 12. *Let* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$ *be three random variables, if* $\mathbf{Y}$ *is independent from variables* $\mathbf{X}$ *and* $\mathbf{Z}$ *the joint min-entropy of* $\mathbf{X}$ *and* $\mathbf{Y}$ *conditioned to* $\mathbf{Z}$ *can be expressed by:*

$$H_\infty(\mathbf{XY}|\mathbf{Z}) = H_\infty(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \tag{6.8}$$

Proof.

$$H_\infty(\mathbf{XY}|\mathbf{Z}) = \min_z H_\infty(\mathbf{XY}|z) \tag{6.9}$$

$$= -\min_z \log \max_{xy} P(xy|z) \tag{6.10}$$

$$= -\min_z \log \max_{xy} P(x|z)P(y|z) \tag{6.11}$$

$$= -\min_z \left[\log \max_x P(x|z) + \log \max_y P(y|z)\right] \tag{6.12}$$

$$= H_\infty(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \tag{6.13}$$

where Eq. 6.11 derives from the consideration that $\mathbf{X}$ and $\mathbf{Y}$ being independent variables, and Eq. 6.13 from $\mathbf{Y}$ and $\mathbf{Z}$ being independent variables.

□

Lemma 13. *Let* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$ *be three random variables, if* $\mathbf{Y}$ *is independent from variables* $\mathbf{X}$ *and* $\mathbf{Z}$ *the joint collision entropy of* $\mathbf{X}$ *and* $\mathbf{Y}$ *conditioned to* $\mathbf{Z}$ *can be expressed by:*

$$H_2(\mathbf{XY}|\mathbf{Z}) = H_2(\mathbf{X}|\mathbf{Z}) + H_2(\mathbf{Y}) \tag{6.14}$$

Proof.

$$\begin{aligned} H_2(\mathbf{XY}|\mathbf{Z}) &= \sum_z p(z) H_2(\mathbf{XY}|z) \\ &= -\sum_z p(z) \log \left(\sum_{xy} p(xy|z)^2 \right) \\ &= -\sum_z p(z) \log \left(\sum_x p(x|z)^2 \sum_y p(y|z)^2 \right) \\ &= \sum_z p(z) \left[H_2(\mathbf{X}|z) + H_2(\mathbf{Y}|z) \right] \\ &= H_2(\mathbf{X}|Z) + H_2(\mathbf{Y}) \end{aligned} \tag{6.15}$$

□

Lemma 14. *Let* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$ *be three random variables and* $\varepsilon > \varepsilon' > 0$, *if* $\mathbf{Y}$ *is independent from variables* $\mathbf{X}$ *and* $\mathbf{Z}$. *Then:*

$$H_\infty^\varepsilon(\mathbf{XY}|\mathbf{Z}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \tag{6.16}$$

Proof. We follow Renner's procedure to proof the superadditivity and subadditivity of smooth quantum min-entropy in [95]. $\exists p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\hat{\mathbf{Y}}}$ with $\delta(p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\mathbf{XZ}}) < \varepsilon'$ and $\delta(p_{\hat{\mathbf{Y}}}, p_{\mathbf{Y}}) < \varepsilon'' = \varepsilon - \varepsilon'$ such that:

$$H_\infty(\hat{\mathbf{X}}|\hat{\mathbf{Z}}) = H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) \tag{6.17}$$

$$H_\infty(\hat{\mathbf{Y}}) = H_\infty^{\varepsilon''}(\mathbf{Y}) \tag{6.18}$$

We have by Lem. 12 that:

$$\begin{aligned} H_\infty(\hat{\mathbf{X}}\hat{\mathbf{Y}}|\hat{\mathbf{Z}}) &= H_\infty(\hat{\mathbf{X}}|\hat{\mathbf{Z}}) + H_\infty(\hat{\mathbf{Y}}) \\ &= H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty^{\varepsilon''}(\mathbf{Y}) \\ &\geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \end{aligned} \tag{6.19}$$

where the inequality holds because $\delta(\mathbf{Y}, \mathbf{Y}) = 0 < \varepsilon''$ We can finish the proof if $H_\infty^\varepsilon(\mathbf{XY}|\mathbf{Z}) \geqslant H_\infty(\hat{\mathbf{X}}\hat{\mathbf{Y}}|\hat{\mathbf{Z}})$. This condition holds because $\delta(p_{\hat{\mathbf{Y}}\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\mathbf{YXZ}}) < \varepsilon$:

$$\begin{aligned} \delta(p_{\hat{\mathbf{Y}}\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\mathbf{YXZ}}) &= \delta(p_{\hat{\mathbf{Y}}} \times p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\mathbf{Y}} \times p_{\mathbf{XZ}}) \\ &\leqslant \delta(p_{\hat{\mathbf{Y}}} \times p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\hat{\mathbf{Y}}} \times p_{\mathbf{XZ}}) + \delta(p_{\hat{\mathbf{Y}}} \times p_{\mathbf{XZ}}, p_{\mathbf{Y}} \times p_{\mathbf{XZ}}) \\ &= \frac{1}{2} \sum_{xyz} |p_{\hat{\mathbf{Y}}}(y)(p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}(xz) - p_{\mathbf{XZ}}(xz))| \\ &\quad + \frac{1}{2} \sum_{xyz} |p_{\mathbf{XZ}}(xz)(p_{\hat{\mathbf{Y}}}(y) - p_{\mathbf{Y}}(y))| \\ &= \delta(p_{\hat{\mathbf{X}}\hat{\mathbf{Z}}}, p_{\mathbf{XZ}}) + \delta(p_{\hat{\mathbf{Y}}}, p_{\mathbf{Y}}) \\ &< \varepsilon' + \varepsilon'' \end{aligned} \tag{6.20}$$

where the first inequality follows from the triangle inequality for the variational distance (see Eq. 2.8). □

Lemma 15. *Given a composite system with three elements* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$ *and let* $\varepsilon > \varepsilon' > 0$. *If the state of the system can be described by a product state of the form* $\rho_{\mathbf{Y}} \otimes \rho_{\mathbf{XZ}}$, *where* $\rho_{\mathbf{Y}}$ *is the operator representation of a random variable* $\mathbf{Y}$ *distributed by* $p_{\mathbf{Y}}$, *i.e.* $\rho_{\mathbf{Y}} = \sum_y p_{\mathbf{Y}}(y) |y\rangle \langle y|$. *Then:*

$$H_\infty^\varepsilon(\mathbf{XY}|\mathbf{Z}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \tag{6.21}$$

where $H_\infty(\mathbf{Y})$ *is the classical min-entropy associated with the random variable* $\mathbf{Y}$.

Proof. We can trivially represent $\rho_{\mathbf{Y}}$ over $\mathcal{P}(\mathcal{H}_{\mathbf{Y}} \otimes \mathcal{H}^1)$ with the density matrix $\rho_{\mathbf{Y}} \otimes id_1$. With this representation we can apply Renner's superadditivity theorem in [95] for product states:

$$H_\infty^\varepsilon(\rho_{\mathbf{XZ}} \otimes \rho_{\mathbf{YI}}|\mathbf{ZI}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty^{\varepsilon''}(\mathbf{Y}|\mathbf{I}) \tag{6.22}$$

where $\varepsilon = \varepsilon' + \varepsilon''$.

The next inequality follows because we can choose $\sigma_{\mathbf{I}} = id_1$ and $\hat{\rho}_{\mathbf{YI}} = \rho_{\mathbf{YI}}$.

$$\begin{aligned} H_\infty^{\varepsilon''}(\rho_{\mathbf{YI}}|\mathbf{I}) &= \sup_{\sigma_{\mathbf{I}}} \sup_{\hat{\rho}_{\mathbf{YI}}} H_\infty(\hat{\rho}_{\mathbf{YI}}\sigma_{\mathbf{I}}) \\ &\geqslant H_\infty(\rho_{\mathbf{YI}}|id_1) \end{aligned} \tag{6.23}$$

Now:

$$\begin{aligned} H_\infty(\rho_{\mathbf{YI}}|id_1) &= -\log \min \lambda | \lambda id_{\mathbf{Y}} \otimes id_1 - \rho_{\mathbf{YI}} \geqslant 0 \\ &= -\log \min \lambda | \lambda id_{\mathbf{Y}} - \rho_{\mathbf{Y}} \geqslant 0 \\ &= -\log \min \lambda | \lambda id_{\mathbf{Y}} - \sum_y p_{\mathbf{Y}}(y) |y\rangle \langle y| \geqslant 0 \\ &= -\log \max_y p_{\mathbf{Y}}(y) \\ &= H_\infty(\mathbf{Y}) \end{aligned} \tag{6.24}$$

where in the first equation we have applied the definition of min-entropy from Eq. 4.28, in the second the tensor product with the identity leaves the state unchanged and the fourth equation follows because the smallest λ that makes $\lambda id_{\mathbf{Y}} - \sum_y p_{\mathbf{Y}}(y) |y\rangle \langle y|$ non-negative is the maximum probability in $p_{\mathbf{Y}}$.

We prove the result putting together Eq. 6.22, Eq. 6.23 and Eq. 6.24:

$$\begin{aligned} H_\infty^\varepsilon(\mathbf{XY}|\mathbf{Z}) &= H_\infty^\varepsilon(\rho_{\mathbf{XZ}} \otimes \rho_{\mathbf{YI}}|\mathbf{ZI}) \\ &\geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty^{\varepsilon''}(\mathbf{Y}|\mathbf{I}) \\ &\geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\rho_{\mathbf{YI}}|id_1) \\ &= H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{Y}) \end{aligned} \tag{6.25}$$

□

We finish the security section proving, in the four security scenarios reviewed in Ch. 4, that the sp-protocol does not reveal any more information than a reconciliation protocol using a code with the same coding rate would reveal.

Theorem 3. *Given a code $\mathcal{C}(n, k)$, a security constant t, the public communication $\mathbf{C}$, and $\mathbf{Z}$ the eavesdropper information, then the min-entropy of the variable $\hat{\mathbf{X}}$ constructed by the sp-protocol, is with probability $1 - 2^{-t}$ greater or equal than that of using an adapted error correcting code of rate R to reconcile $\mathbf{X}$ and $\mathbf{Y}$ minus the security constant:*

$$H_\infty(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1 - R) - t \tag{6.26}$$

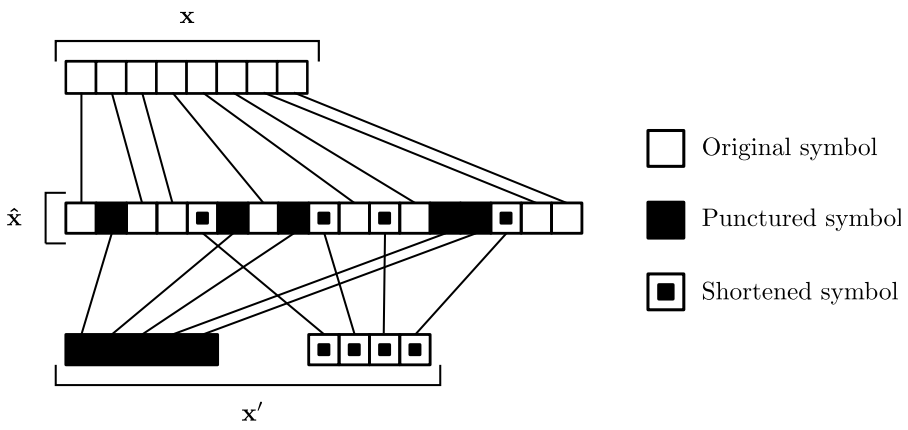

Figure 6.3: Extended string construction. The figure shows how the extended string $\hat{\mathbf{x}}$ is constructed from a random permutation of two strings: the original string to be reconciled, $\mathbf{x}$, and a string consisting of punctured and shortened symbols, $\mathbf{x}'$.

Proof. Directly given by Eq. 4.20:

$$H_\infty(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty(\hat{\mathbf{X}}|\mathbf{Z}) - |\mathbf{C}| - t \tag{6.27}$$

Distinguishing in $\hat{\mathbf{X}}$ part of the variable that corresponds to the sequence to be reconciled, $\mathbf{X}$, and the additional variable used to extend the original sequence, $\mathbf{X}'$ (see its correspondence with strings in Fig. 6.3):

$$= H_\infty(\mathbf{XX}'|\mathbf{Z}) - |\mathbf{C}| - t \tag{6.28}$$

Since $\mathbf{X}'$ is independent of $\mathbf{Z}$ and $\mathbf{X}$ by construction, Lem. 12 can be applied:

$$= H_\infty(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{X}') - |\mathbf{C}| - t \tag{6.29}$$

The entropy of $H_\infty(\mathbf{X}')$ takes the value of the number of random $p+s$ bits:

$$= H_\infty(\mathbf{X}|\mathbf{Z}) + |\mathbf{X}|\frac{\pi+\sigma}{1-\pi-\sigma} - |\mathbf{C}| - t \tag{6.30}$$

The length of the conversation $|\mathbf{C}|$ is $s+n-k$, which in the proposed protocol stand for the s shortened bits and the syndrome of X'. It can be written as a function of the size of $\mathbf{X}$, π and σ:

$$= H_\infty(\mathbf{X}|\mathbf{Z}) + |\mathbf{X}|\frac{\pi+\sigma}{1-\pi-\sigma} - |\mathbf{X}|\frac{(1-R_0)+\sigma}{1-\pi-\sigma} - t \tag{6.31}$$

and thus

$$= H_\infty(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) - t \tag{6.32}$$

□

Theorem 4. *Given a code $\mathcal{C}(n,k)$, a security constant t, the public communication $\mathbf{C}$, and $\mathbf{Z}$ the eavesdropper information, then the collision entropy of the variable $\hat{\mathbf{X}}$ constructed by the sp-protocol, is with probability $1-2^{-(t/2-1)}$ greater or equal than that of using an adapted error correcting code of rate R to reconcile $\mathbf{X}$ and $\mathbf{Y}$ minus the security constant:*

$$H_2(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_2(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) - t \tag{6.33}$$

Proof. From Eq. 4.17:

$$H_2(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_2(\hat{\mathbf{X}}|\mathbf{Z}) - |\mathbf{C}| - t \tag{6.34}$$

The same argument as in Th. 3 follows:

$$= H_2(\mathbf{X}\mathbf{X}'|\mathbf{Z}) - |\mathbf{C}| - t \tag{6.35}$$

We now apply Lem. 13:

$$= H_2(\mathbf{X}|\mathbf{Z}) + H_2(\mathbf{X}') - |\mathbf{C}| - t \tag{6.36}$$

and, operating:

$$= H_2(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) - t \tag{6.37}$$

□

Theorem 5. *Given a code $\mathcal{C}(n,k)$, $\varepsilon_1,\varepsilon_2 > 0$, the public communication $\mathbf{C}$, and $\mathbf{Z}$ the eavesdropper information, then the smooth min-entropy of the variable $\hat{\mathbf{X}}$ constructed by the sp-protocol, is greater or equal than that of using an adapted error correcting code of rate R to reconcile $\mathbf{X}$ and $\mathbf{Y}$ minus a security constant:*

$$H_\infty^{\varepsilon_1+\varepsilon_2}(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) - \log\frac{1}{\varepsilon_2} \tag{6.38}$$

with $\varepsilon_1 > \varepsilon' > 0$

Proof. From Eq. 4.9 and Lem. 14:

$$H_\infty^{\varepsilon_1+\varepsilon_2}(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{X}') - |\mathbf{C}| - \log\frac{1}{\varepsilon_2} \tag{6.39}$$

which after some manipulation becomes:

$$= H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) - \log\frac{1}{\varepsilon_2} \tag{6.40}$$

□

Theorem 6. *Given a code* $\mathcal{C}(n, k)$*, and* $\varepsilon > 0$*, the public communication* $\mathbf{C}$*, and a composite quantum system described by the operator* ρ_{XYZ} *where* $\mathbf{Z}$ *represents the eavesdropper's system, then the min-entropy of* $\hat{\mathbf{X}}$*, is greater or equal than that of using an adapted error correcting code of rate* R *to reconcile* $\mathbf{X}$ *and* $\mathbf{Y}$*:*

$$H_\infty^\varepsilon(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) \tag{6.41}$$

with $\varepsilon > \varepsilon' > 0$*.*

Proof. From Eq. 4.34 and Lem. 15:

$$H_\infty^\varepsilon(\hat{\mathbf{X}}|\mathbf{ZC}) \geqslant H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) + H_\infty(\mathbf{X}') - |\mathbf{C}| \tag{6.42}$$

which operating becomes:

$$= H_\infty^{\varepsilon'}(\mathbf{X}|\mathbf{Z}) - |\mathbf{X}|(1-R) \tag{6.43}$$

□

6.5 SIMULATION RESULTS

In this section we discuss the efficiency of the rate-compatible information reconciliation protocol for strings that can be regarded as the input and output of a BSC. We compare the results of the protocol to regular LDPC codes as proposed in Sec. 5.5 and to Cascade.

Fig. 6.4 shows the efficiency, calculated as defined in Eq. (5.2), in the reconciliation process simulated for three different alternatives: (i) using the Cascade protocol, (ii) using LDPC codes without adapting the information rate, and (iii) using LDPC codes adapting the information rate with the rate-compatible protocol proposed here. The target error range selected is $[0.055, 0.11]$, where a high efficiency protocol is a must. Low cross over probabilities do not demand a close to optimal efficiency since other requisites, such as the throughput, are more critical in obtaining a high secret key rate. In order to achieve a efficiency close to 1, the error range $[0.055, 0.11]$ has been divided into two correction intervals: $R_0(\mathcal{C}_1) = 0.5$, $R_0(\mathcal{C}_2) = 0.6$ and $\delta = 0.1$. The codes have been constructed using families of LDPC codes specifically optimized for the BSC.

The construction process has been optimized using a modified progressive edge-growth algorithm for irregular codes with a detailed check node degree distribution [74]. A codeword length of 2×10^5 bits has been used.

The results show that there is a small price to pay for the rate adaptation. LDPC codes without puncturing and shortening behave slightly better near their threshold, however for the δ value chosen the penalty is very small and the rate-compatible protocol allows to reconcile strings in all the range with $f \leqslant 1.1$. The unmodulated LDPC

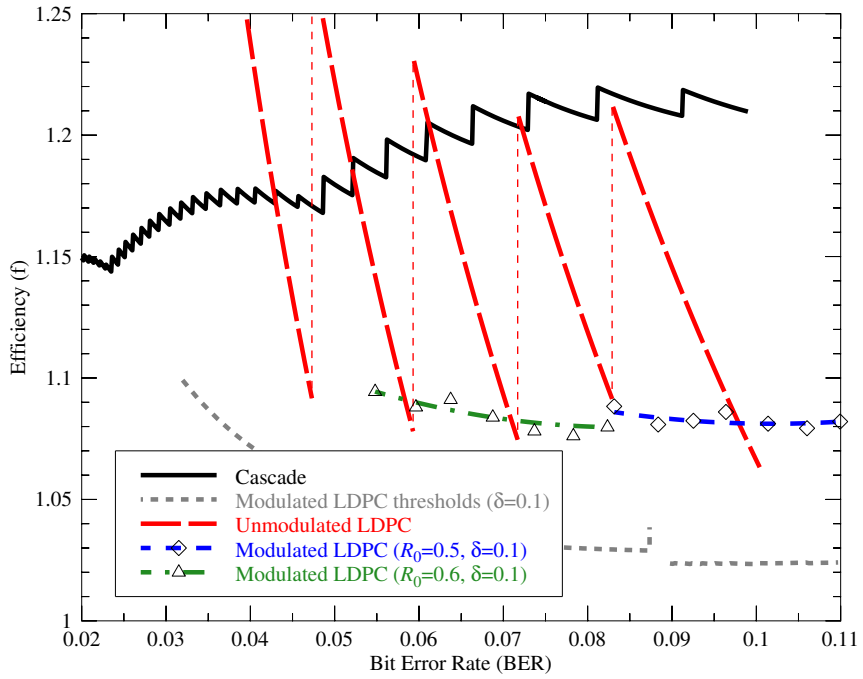

Figure 6.4: Computed efficiency for medium to high error rates, a typical range expected in shared quantum channel environments, long distances or high losses scenarios, such as in networks, and where obtaining high efficiency is critical. The solid line is the Cascade efficiency. Its parameters are the same than for Fig. 6.2. The dotted line represents the modulated LDPC thresholds. For all LDPC results shown here $\delta = 0.1$. The long, thick, dashed lines joined by thin dashed lines is the efficiency of an unmodulated code. Short dash and dash-dotted lines are the results for the modulated codes. Dash-dotted is for a rate $R_0 = 0.6$ and short dash are for $R_0 = 0.5$, triangles and diamonds are used to mark the computed points. The smooth and efficient behavior of the modulated, rate adapted codes, as compared to the unmodulated version is to be noted. The gain in efficiency over Cascade allows for an extended usability range of the system at high error rates.

codes exhibit an undesirable saw behavior that can lead to efficiencies worse than that of Cascade unless many different codes are calculated, incurring in an unacceptable penalty in Central Processing Unit (CPU) time. The new protocol works at a much better efficiency than Cascade, that performs in all the tested range with $f \geqslant 1.17$.

7 CONCLUSION

If your quanta are broke. We fix 'em.

— Seth Lloyd [66]

This thesis discusses some improvements in the distillation of information theoretically secure secret keys. In contrast to computational security, ITS allows the legitimate parties to assume that their keys remain secure independently of any unforeseen technical or theoretical developments.

Several scenarios allow to distill information theoretically secure secret keys. The common feature among them is that they act as a source of correlated randomness. The key distillation process can be divided in two steps: information reconciliation and privacy amplification. Information reconciliation allows to establish a common string while in the privacy amplification step a shorter but more secure key is created. Both steps are highly coupled: in essence every bit exchanged in the information reconciliation step implies that one additional bit has to be removed of the final key in the privacy amplification step.

The problem of correcting the discrepancies between the strings of the legitimate parties in SKD is known as the problem of source coding with side information by the information theory community. Under this paradigm, the theoretical limits of information reconciliation are given by the Slepian-Wolf bound. In some models of SKD the strings can be modeled as the input and output of a BSC, if the assumption holds the theoretical limits of information reconciliation can be reached with linear error correcting codes. Information reconciliation is basically error correction.

In this thesis we have adopted a pragmatic approach towards error correction and developed specific techniques well suited for SKD purposes. In the real scenario of QKD we have to deal with a broad range of error rates, ranging from 1% to 11%. Moreover, information reconciliation has to be performed in near real time, limiting the number of accesses to the communications channel. As opposed to the eavesdropper that should safely be supposed to have access to unlimited resources, the legitimate parties are equipped with a finite amount of resources. LDPC codes were then selected as the framework to develop a practical information reconciliation scheme.

In Ch. 3 we design LDPC codes for the BSC with thresholds close to the theoretical limit. The results in Ch. 5 show that each code, adapted to a specific error rate, provides a close to optimal solution for reconciliation provided that the strings can be modeled as the input and output of the adequate BSC. The reconciliation efficiency, however,

drops sharply as the error rate of the BSC moves away from the design point. To solve this issue, we propose the sp-protocol in Ch. 6, a simple protocol based on puncturing and shortening LDPC codes. This protocol limits the information gathered by the eavesdropper to the same amount of information than an adapted code would reveal; even if more data is exchanged on the public channel. The extra data exchanged increases the required bandwidth but keeps the interactivity requirements to zero, compared to the heavy use of two-ways communications that cascade-like protocols require. We have shown that this is the case in several ITS models. In particular some QKD protocols can use the sp-protocol as an efficient information reconciliation primitive.

The sp-protocol allows the legitimate parties to reconcile their chains with a continuous efficiency curve, and as the efficiency of LDPC codes under puncturing and shortening can be analytically described and optimized, the results proved in this thesis allow to address the information reconciliation problem as a code design problem. The results obtained on Ch. 6 for the sp-protocol indicate that efficiency values close to the theoretical limits can be obtained.

This new framework allows to consider the information reconciliation step following the random distribution of SKD protocols as a code design problem. The ideas can be applied to any protocol beyond the specific setting that we have analyzed, e.g. recently the sp-protocol has been proposed for the reconciliation of continuous-variable QKD [56]. We believe that this opens the doors to consider simpler and possibly better schemes to process all the classical part of SKD protocols as a whole.

ACRONYMS

AMS American Mathematical Society

AWGN Additive White Gaussian Noise

BCH Bose, Ray-Chaudhuri and Hocquenghem codes

BBBSS Bennett, Bessette, Brassard, Salvail and Smolin's Information Reconciliation Protocol

BEC Binary Erasure Channel

BER Binary Error Rate

BSC Binary Symmetric Channel

CPU Central Processing Unit

DDE Discretized Density Evolution

DMC Discrete Memoryless Channel

DiffE Differential Evolution

FER Frame Error Rate

iid Independent Identically Distributed

ITS Information Theoretic Security

MAP Maximum a Posteriori

ML Maximum Likelihood

MPA Message Passing Algorithm

LDPC Low Density Parity Check

LLR Log Likelihood Ratio

PEG Progressive Edge Growth

QBER Quantum Bit Error Rate

QKD Quantum Key Distribution

RSA Rivest, Shamir and Adleman's algorithm

SKD Secret Key Distribution

SPA Sum Product Algorithm

BIBLIOGRAPHY

[1] The information theory. *Fortune (Magazine)*, pages 136–158, Dec. 1953.

[2] ETSI EN 302 307. DVB-S2, 2006.

[3] J. Aczel and Z. Daroczy. *On measures of information and their characterizations*. Academic Press, 1975.

[4] J. Aczél, B. Forte, and C. T. Ng. Why the shannon and hartley entropies are 'natural'. *Advances in Applied Probability*, 6(1):pp. 131–146, 1974. ISSN 00018678.

[5] R. Ahlswede and I. Csiszar. Common randomness in information theory and cryptography. i. secret sharing. *IEEE Transactions on Information Theory*, 39(4):1121 –1132, Jul. 1993.

[6] R. B. Ash. *Information Theory*. Dover Publications, 1965.

[7] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[8] C. H. Bennett, G. Brassard, and J. M. Robert. How to reduce your enemy's information (extended abstract). In *Advances in Cryptology*, pages 468–476. Springer-Verlag, 1986.

[9] C. H. Bennett, G. Brassard, and J. M. Robert. Privacy Amplification by Public Discussion. *SIAM Journal on Computing*, 17(2): 210–229, 1988.

[10] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *Journal of Cryptology*, 5(1): 3–28, 1992. ISSN 0933-2790.

[11] C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, Nov. 1995.

[12] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (Corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[13] M. Bloch, A. Thangaraj, S. W. McLaughlin, and J. M. Merolla. Ldpc-based gaussian key reconciliation. In *IEEE Information Theory Workshop*, pages 116 –120, Mar. 2006.

[14] G. Brassard and L. Salvail. Secret-Key Reconciliation by Public Discussion. In *CRYPTO 93 - Workshop on the theory and application of cryptographic techniques*, volume 765 of *Lecture Notes in Computer Science*, pages 410–423. Springer-Verlag, 1994.

[15] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson. Fast, efficient error reconciliation for quantum cryptography. *Physical Review A*, 67(5):052303–+, May 2003.

[16] C. Cachin. *Entropy Measures and Unconditional Security in Cryptography*. PhD thesis, ETH Zurich, 1997. Reprint as vol. 1 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-185-7, Hartung-Gorre Verlag, Konstanz, 1997.

[17] S. Y. Chung, G. D. Forney, and T. J. Richardson. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications Letters*, 5:58–60, 2001.

[18] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, August 1991.

[20] C. Crépeau. Réconciliation et distillation publiques de secret, 1995.

[21] I. Csiszár. Axiomatic characterizations of information measures. *Entropy*, 10:261–273, 2008.

[22] I. Csiszar and J. Korner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24(3):339 – 348, may 1978.

[23] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4 –31, feb. 2011.

[24] I. Devetak and A. Winter. Distillation of secret key and entanglement from quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2053):207–235, 2005.

[25] J. L. Duligall, M. S. Godfrey, K. A. Harrison, W. J. Munro, and J. G. Rarity. Low cost and compact quantum cryptography. *New Journal of Physics*, 8:249, 2006.

[26] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO 84 - Advances in cryptology*, pages 10–18. Springer-Verlag New York, Inc., 1985.

[27] P. Elias. Coding for Two Noisy Channels. In *The 3rd London Symposium on Information Theory*, pages 61–76. Buttersworth's Scientific Publications, September 1956.

[28] D. Elkouss, A. Leverrier, R. Alleaume, and J. J. Boutros. Efficient reconciliation protocol for discrete-variable quantum key distribution. In *IEEE International Symposium on Information Theory*, pages 1879–1883, Jul. 2009.

[29] D. Elkouss, J. Martínez, D. Lancho, and V. Martín. Rate compatible protocol for information reconciliation: An application to qkd. In *IEEE Information Theory Workshop*, pages 145–149, Jan. 2010.

[30] D. Elkouss, J. Martinez-Mateo, and V. Martin. Secure rate-adaptive reconciliation. In *International Symposium on Information Theory and its Applications*, pages 179–184, Oct. 2010.

[31] D. Elkouss, J. Martinez-Mateo, and V. Martin. Information reconciliation for quantum key distribution. *Quantum Information & Computation*, 11(3):226–238, 2011.

[32] D. Elkouss, J. Martinez-Mateo, and V. Martin. Untainted puncturing for irregular low-density parity-check codes. *IEEE Wireless Communications Letters*, 1(6):585–588, 2012.

[33] David Elkouss, Daniel Lancho, Jesús Martínez Mateo, and Vicente Martín. Método y sistema de comunicaciones para la reconciliación de información en qkd mediante el uso de códigos ldpc adaptando la tasa de información, October 24 2012. ES Patent ES 2389217 B2.

[34] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh. Current status of the darpa quantum network. Technical Report quant-ph/0503058, Mar 2005.

[35] R. Fano. *Transmission of Information*. The MIT Press, 1961.

[36] A. Feinstein. *Foundations of Information Theory*. 1958.

[37] S. Fossier, E. Diamanti, T. Debuisschert, A. Villing, R. Tualle-Brouri, and P. Grangier. Field test of a continuous-variable quantum key distribution prototype. *New Journal of Physics*, 11 (4):045023, 2009.

[38] R. G. Gallager. *Low-density parity-check codes*. MIT Press, Cambridge, 1963.

[39] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., New York, NY, USA, 1968.

[40] R. G. Gallager. Claude E. Shannon: a retrospective on his life, work, and impact. *IEEE Transactions on Information Theory*, 47(7): 2681–2695, November 2001.

[41] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden. Quantum cryptography. *Reviews of Modern Physics*, 74(1):145–195, 2002.

[42] A. A. Gohari and V. Anantharam. Information-theoretic key agreement of multiple terminals: Part i. *IEEE Transactions on Information Theory*, 56(8):3973 –3996, aug. 2010.

[43] A. A. Gohari and V. Anantharam. Information-theoretic key agreement of multiple terminals: Part ii: Channel model. *IEEE Transactions on Information Theory*, 56(8):3997 –4010, aug. 2010.

[44] D. Gottesman and H. K. Lo. Proof of security of quantum key distribution with two-way classical communications. *IEEE Transactions on Information Theory*, 49(2):457 –475, 2003.

[45] R. M. Gray. *Entropy and information theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[46] F. Grosshans and P. Grangier. Continuous variable quantum cryptography using coherent states. *Physical Review Letters*, 88 (5):057902, Jan 2002.

[47] J. Ha, J. Kim, and S. W. McLaughlin. Rate-compatible puncturing of low-density parity-check codes. *IEEE Transactions on Information Theory*, 50(11):2824–2836, Nov. 2004.

[48] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin. Rate-compatible punctured low-density parity-check codes with short block lengths. *IEEE Transactions on Information Theory*, 52(2):728–738, Feb. 2006.

[49] J. Hagenauer. Rate-compatible punctured convolutional codes (rcpc codes) and their applications. *IEEE Transactions on Communications*, 36(4):389 –400, apr 1988.

[50] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, 42(2):429–445, Mar. 1996.

[51] J. Han and X. Qian. Auto-adaptive interval selection for quantum key distribution. *Quantum Information and Computation*, 9(7& 8): 693–700, Jul. 2009.

[52] R. Hill. *A First Course in Coding Theory*. Clarendon Press, 1986.

[53] C. H. Hsu and A. Anastasopoulos. Capacity achieving ldpc codes through puncturing. *IEEE Transactions on Information Theory*, 54(10):4698–4706, Oct. 2008.

[54] X. Y. Hu, E. Eleftheriou, and D. M. Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Transactions on Information Theory*, 51(1):386–398, Jan. 2005.

[55] S. J. Johnson. *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. Cambridge University Press, Jan. 2010.

[56] P. Jouguet, S. Kunz-Jacques, and A. Leverrier. Long-distance continuous-variable quantum key distribution with a gaussian modulation. *Phys. Rev. A*, 84:062317, Dec 2011.

[57] G. D. Forney Jr. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, 47(2):520 –548, feb 2001.

[58] D. Kahn. *The Codebreakers: The Story of Secret Writing*. Macmillan Publishing Co., 1967.

[59] Jaehong Kim, A. Ramamoorthy, and S. Mclaughlin. The design of efficiently-encodable rate-compatible LDPC codes. *IEEE Transactions on Communications*, 57(2):365–375, Feb. 2009.

[60] D. Lancho, J. Martinez, D. Elkouss, M. Soto, and V. Martin. QKD in standard optical telecommunications networks. In *Proceedings of Intenational ICST Conference on Quantum Communication and Quantum Networking (QuantumComm 2009)*, pages 142–149, Oct. 2009.

[61] A. Leverrier, R. Alléaume, J. Boutros, G. Zémor, and P. Grangier. Multidimensional reconciliation for continuous-variable quantum key distribution. *Physical Review A*, 77(4):042325–+, 2008.

[62] Y. Liang, H. V. Poor, and S. Shamai (Shitz). Information theoretic security. *Foundations and Trends in Communications and Information Theory*, 5:355–580, Apr. 2009.

[63] S. Liu. *Information-theoretic secret key agreement*. PhD thesis, Technische Universiteit Eindhoven, 2002. ISBN 90-386-1001-7.

[64] S. Liu, H. C. A. Van Tilborg, and M. Van Dijk. A practical protocol for advantage distillation and information reconciliation. *Designs Codes and Cryptography*, 30(1):39–62, 2003. ISSN 0925-1022.

[65] A. D. Liveris, Zixiang Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6(10):440–442, Oct. 2002.

[66] S. Lloyd. Seth lloyd lectures at ICQ. *New Bit*, (15):10, Winter 2011.

[67] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. Practical Loss-Resilient Codes. In *In Proceedings of the 29th annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.

[68] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *STOC 98 - Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 249–258, New York, NY, USA, 1998. ACM.

[69] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Transactions on Information Theory*, 47:585–598, 2001.

[70] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2): 399–431, Mar. 1999.

[71] D. J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Oct. 2003.

[72] D. J. C. MacKay and R. M. Neal. Good codes based on very sparse matrices. In *Cryptography and Coding. 5th IMA Conference, number 1025 in Lecture Notes in Computer Science*, pages 100–111. Springer, 1995.

[73] D. J. C. MacKay and R. M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32: 1645–1646, 1996.

[74] J. Martinez-Mateo, D. Elkouss, and V. Martin. Interactive reconciliation with low-density parity-check codes. In *6th Int. Symposium on Turbo Codes & Iterative Information Processing*, pages 270–274, Sep. 2010.

[75] J. Martinez-Mateo, D. Elkouss, and V. Martin. Improved construction of irregular progressive edge-growth tanner graphs. *IEEE Communications Letters*, 14(12):1155–1157, Dec. 2010.

[76] J. Martinez-Mateo, D. Elkouss, and V. Martin. Blind reconciliation. *Quantum Information and Computation*, 12(9&10):0791–0812, 2012.

[77] J. L. Massey. Shannon's 'proof' of the noisy coding theorem. In *IEEE International Symposium on Information Theory*, page 107, 1977.

[78] U. Maurer and S. Wolf. Secret-key agreement over unauthenticated public channels .i. definitions and a completeness result.

IEEE Transactions on Information Theory, 49(4):822 – 831, april 2003.

[79] U. Maurer and S. Wolf. Secret-key agreement over unauthenticated public channels-part ii: the simulatability condition. *IEEE Transactions on Information Theory*, 49(4):832 – 838, april 2003.

[80] U. Maurer and S. Wolf. Secret-key agreement over unauthenticated public channels .iii. privacy amplification. *IEEE Transactions on Information Theory*, 49(4):839 – 851, april 2003.

[81] Ueli Maurer and Stefan Wolf. Information-theoretic key agreement: From weak to strong secrecy for free. In *Lecture Notes in Computer Science*, pages 351–368. Springer-Verlag, 2000.

[82] U.M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, May 1993.

[83] R.J. McEliece. Achieving the shannon limit: A progress report. In *38th Annual Allerton Conference on Communication, Control and Computing*, 2000.

[84] A. Meier and S. Heimlicher. Information-theoretically secure key agreement from arbitrarily correlated information. 2004.

[85] K. C. Nguyen, Gilles Van Assche, and Nicolas J. Cerf. Side-information coding with turbo codes and its application to quantum key distribution. In *International Symposium on Information Theory and its Applications*, 2004.

[86] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information (Cambridge Series on Information and the Natural Sciences)*. Cambridge University Press, Jan. 2004.

[87] C. Pacher, A. Abidin, T. Lorünser, M. Peev, R. Ursin, A. Zeilinger, and J Å. Larsson. Hacking qkd protocols that employ non-its authentication. In *Annual Conference on Quantum Cryptography*, 2011.

[88] N. Papanikolaou and R. Nagarajan. Classical security protocols for qkd systems, 2006.

[89] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Fürst, J. D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hübel, G. Humer, T. Länger, M. Legré, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J. B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki,

M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden, and A. Zeilinger. The secoqc quantum key distribution network in vienna. *New Journal of Physics*, 11(7):075001, 2009.

[90] J. R. Pierce. *An Introduction to Information Theory*. Dover Publications, Nov. 1980.

[91] H. Pishro-Nik and F. Fekri. Results on punctured low-density parity-check codes and improved iterative decoding techniques. *IEEE Transactions on Information Theory*, 53(2):599–614, Feb. 2007.

[92] A. Politi, J. C. F. Matthews, and J. L. O'Brien. Shor's quantum factoring algorithm on a photonic chip. *Science*, 325(5945):1221, 2009.

[93] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, 1979.

[94] T. C. Ralph. Continuous variable quantum cryptography. *Physical Review A*, 61(1):010303, Dec 1999.

[95] R. Renner. *Security of Quantum Key Distribution*. PhD thesis, ETH Zurich, 2005. Diss. ETH No. 16242.

[96] R. Renner and S. Wolf. Smooth Renyi entropy and applications. In *International Symposium on Information Theory*, page 232. IEEE.

[97] R. Renner and S. Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *ASIACRYPT 2005 - Advances in Cryptology*, pages 199–216. Springer-Verlag, 2005.

[98] R. Renner, N. Gisin, and B. Kraus. Information-theoretic security proof for quantum-key-distribution protocols. *Physical Review A*, 72(1):12332, 2005.

[99] A. Rényi. On Measures Of Entropy And Information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1960.

[100] F. M. Reza. *An Introduction to Information Theory*. Dover Publications, Sep. 1994.

[101] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, Feb. 2001.

[102] T. J. Richardson and R. L. Urbanke. *Modern coding theory*. Cambridge University Press, 2008.

[103] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, Feb. 2001.

[104] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[105] V. Scarani and R. Renner. Quantum cryptography with finite resources: unconditional security bound for discrete-variable protocols with one-way post-processing. *Physical Review Letters*, 100:200501, 2008.

[106] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev. The security of practical quantum key distribution. *Review of Modern Physics*, 81:1301–1350, Sep. 2009.

[107] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27:379–423, Jul. 1948.

[108] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

[109] C. E. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8 –19, Sep. 1956.

[110] A. Shokrollahi. Ldpc codes: An introduction. *Digital Fountain, Inc., Tech. Rep*, page 2, 2003.

[111] A. Shokrollahi and R. Storn. Design of efficient erasure codes with differential evolution. *IEEE International Symposium on Information Theory*, pages 5–, 2000.

[112] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, Oct. 1997.

[113] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4): 471–480, Jul. 1973.

[114] D. Stinson. Universal hashing and authentication codes. In *Advances in cryptology*, volume 576 of *Lecture Notes in Computer Science*, pages 74–85, 1992.

[115] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, 1995.

[116] R. Storn and K. Price. Minimizing the real functions of the icec'96 contest by differential evolution. In *IEEE Conference on Evolutionary Computation*, pages 842–844, 1996.

[117] T. Sugimoto and K. Yamazaki. A study on secret key reconciliation protocol cascade. *IEICE Transactions on Fundamentals*, E83-A(10):1987–1991, Oct. 2000.

[118] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, Sep. 1981.

[119] S. ten Brink. Convergence of iterative decoding. *Electronics Letters*, 35(10):806 –808, May 1999.

[120] S. ten Brink, G. Kramer, and A. Ashikhmin. Design of low-density parity-check codes for modulation and detection. *IEEE Transactions on Communications*, 52(4):670 – 678, Apr. 2004.

[121] G. Van Assche. *Quantum Cryptography and Secret-Key Distillation*. Cambridge University Pres, 2006.

[122] G. Van Assche, J. Cardinal, and N. J. Cerf. Reconciliation of a quantum-distributed gaussian key. *IEEE Transactions on Information Theory*, 50:394, 2004.

[123] B. N. Vellambi and F. Fekri. Finite-Length Rate-Compatible LDPC Codes: A Novel Puncturing Scheme. *IEEE Transactions on Communications*, 57(2):297–301, Feb. 2009.

[124] S. Watanabe, R. Matsumoto, T. Uyematsu, and Y. Kawano. Key rate of quantum key distribution with hashed two-way classical communication. *Physical Review A*, 76:032312, Sep. 2007.

[125] S. Watanabe, R. Matsumoto, and T. Uyematsu. Tomography increases key rates of quantum-key-distribution protocols. *Physical Review A*, 78(4):042316, 2008.

[126] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, Jun. 1981.

[127] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, 1996. Diss. No. 440.

[128] N. Wiberg, H. A. Loeliger, and R. Kotter. Codes and iterative decoding on general graphs. *European Transactions on Telecommunications*, 1995.

[129] A. D. Wyner. Recent results in the shannon theory. *IEEE Transactions on Information Theory*, 20(1):2–10, 1974.

[130] K. Yamazaki and T. Sugimoto. On secret reconciliation protocol–modification of "cascade" protocol. In *IEEE International Symposium on Information Theory and Its Applications*, pages 223–226, Honolulu, Hawaii, Nov. 2000.

[131] K. Yamazaki, M. Osaki, and O. Hirota. On reconciliation of discrepant sequences shared through quantum mechanical channels. In *Proceedings of the First International Workshop on Information Security*, ISW '97, pages 345–356. Springer-Verlag, 1998.

[132] M. R. Yazdani and A. H. Banihashemi. On construction of rate-compatible low-density parity-check codes. *IEEE Communications Letters*, 8(3):159–161, Mar. 2004.

[133] R. W. Yeung. *A First Course in Information Theory (Information Technology: Transmission, Processing and Storage)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[134] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory*, 48(6):1250–1276, 2002.

VITAE

David Elkouss Coronas is a doctoral researcher at the Technical University of Madrid (UPM). He holds a degree of Ingeniero de Telecomunicaciones by the UPM and a Diplôme d'Ingénieur by Télécom ParisTech. Currently, he is member of the research group on Quantum Information and Computation and a former member of the group on Network and Communications Services. His interests are related to error correcting codes and the classical post-processing of quantum key distribution protocols.

COLOPHON

This thesis was typeset with LaTeX 2ε using Hermann Zapf's *Palatino* and *Euler* type faces (Type 1 PostScript fonts *URW Palladio L* and *FPL* were used). The listings are typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera". (Type 1 PostScript fonts were made available by Malte Rosenau and Ulrich Dirr.)

The typographic style is available for LaTeX via CTAN as "`classicthesis`".