University of Padova
Department of Information Engineering

Ph.D. School in Information Engineering
Information Science and Technology
XXVI Class

Classical processing algorithms for Quantum Information Security

Ph.D. candidate:
Matteo Canale

Supervisor:
Dr. Nicola Laurenti

Course coordinator: Prof. Carlo Ferrari

Ph.D. School director:
Prof. Matteo Bertocco

Academic Year 2013-2014

Indice

Abstract **5**

Sommario **7**

1 Introduction **9**

2 Randomness extraction **13**

2.1 Preliminary definitions and problem statement 14

2.2 Types of extractors 19

2.2.1 Deterministic extractors 19

2.2.2 Seeded extractors 21

2.2.3 Extractors from multiple independent sources 24

2.3 Practical constructions for seeded extractors 25

2.3.1 (Almost) 2-Universal hash functions 26

2.3.2 Trevisan's extractor 29

2.4 Design and implementation of a randomness extractor for the Quantis device 31

2.4.1 Hardware setup 32

2.4.2 Notation 33

2.4.3 General considerations 33

2.4.4 Extractor design 38

2.4.5 Software architecture 39

2.4.6 Experimental results 41

3 Quantum Key Distribution **43**

3.1 Information-theoretic secret key agreement: system model 44

3.2 Taxonomy of QKD protocols 47

3.3 Security of quantum key distribution protocols 48

3.3.1 Quantum principles for security 49

3.3.2 Attack models 50

3.3.3 Secrecy measures 51

3.4 Explicit QKD protocols 55

3.4.1 Bennett-Brassard 1984 protocol (BB84) 55

3.4.2 Efficient BB84 protocol 57

3.4.3 Bennett 1992 protocol 59

3.4.4 Remarks on the single photon assumption for practical QKD 61
3.5 Information reconciliation 61
3.5.1 Direct reconciliation: problem statement and coding approaches . . 62
3.5.2 Analysis of the Winnow protocol 68
3.5.3 LDPC codes for information reconciliation 78
3.5.4 Error verification 82
3.6 Privacy amplification 82
3.6.1 Asymptotic information leakage under selective individual attacks . 83
3.6.2 Finite-key analysis 89

4 Experimental free-space Quantum Key Distribution 93
4.1 Software implementation 94
4.1.1 FPGA software interface 95
4.1.2 Processing module 95
4.1.3 Networking module 97
4.2 Experimental B92 over 50 meters free-space link 104
4.2.1 Transmission setup and protocol 104
4.2.2 Attack model 105
4.2.3 Channel estimation 106
4.2.4 Classical processing 108
4.2.5 Authentication and transmission over the public channel 109
4.2.6 Experimental results 110
4.3 Efficient BB84 in free-space with finite-key analysis 110
4.3.1 Experimental setup 111
4.3.2 Classical post-processing 111
4.3.3 Finite-length secret key rate 113
4.3.4 Experimental results 114
4.3.5 Discussion 118
4.4 Long distance free-space quantum key distribution using turbulence as a resource 118
4.4.1 Experimental setup 119
4.4.2 Preliminary analysis 121
4.4.3 QBER and secret key rate prediction 123
4.4.4 Experimental results 124
4.4.5 Comment on security and conclusions 126

5 Conclusions 127

Acknowledgments 129

Publications 131

Abstract

In this thesis, we investigate how the combination of quantum physics and information theory could deliver solutions at the forefront of information security, and, in particular, we consider two focus applications: randomness extraction as applied to quantum random number generators and classical processing algorithms for quantum key distribution (QKD).

We concentrate on practical applications for such tools. We detail the implementation of a randomness extractor for a commercial quantum random number generator, and we evaluate its performance based on information theory. Then, we focus on QKD as applied to a specific experimental scenario, that is, the one of free-space quantum links. Commercial solutions with quantum links operating over optical fibers, in fact, already exist, but suffer from severe infrastructure complexity and cost overheads. Free-space QKD allows for a higher flexibility, for both terrestrial and satellite links, whilst experiencing higher attenuation and noise at the receiver. In this work, its feasibility is investigated and proven in multiple experiments over links of different length, and in various channel conditions. In particular, after a thorough analysis of information reconciliation protocols, we consider finite-key effects as applied to key distillation, and we propose a novel adaptive real-time selection algorithm which, by leveraging the turbulence of the channel as a resource, extends the feasibility of QKD to new noise thresholds. By using a full-fledged software for classical processing tailored for the considered application scenario, the obtained results are analyzed and validated, showing that quantum information security can be ensured in realistic conditions with free-space quantum links.

Sommario

In questa tesi si mostra come la combinazione tra la fisica quantistica e la teoria dell'informazione permetta di realizzare protocolli all'avanguardia per la sicurezza dell'informazione. Si considerano in particolare due specifiche applicazioni: la randomness extraction per generatori quantistici di numeri casuali e gli algoritmi di processing classici nel contesto della crittografia quantistica.

Focalizzando lo studio sugli sviluppi pratici delle menzionate applicazioni, si descrive anzitutto in dettaglio l'implementazione di un randomness extractor per un generatore quantistico di numeri casuali ad uso commerciale, e si valutano le sue prestazioni sulla base della teoria dell'informazione.

Quindi, ci si concentra sulla crittografia quantistica nello specifico scenario sperimentale dei canali quantistici in spazio libero. Ad oggi, infatti, sono disponibili soluzioni commerciali con canali quantistici in fibra ottica, che sono però condizionate da un'alta complessitá infrastrutturale e da un elevato costo economico. La crittografia quantistica in spazio libero, al contrario, permette una maggior flessibilitá, sia per link terrestri che per link satellitari, nonostante essa soffra di perdite e rumore più elevati al ricevitore. Attraverso la realizzazione di vari esperimenti su link di diversa lunghezza e con diverse condizioni di canale, se ne dimostra la fattibilitá. In particolare, dopo un'accurata analisi dei protocolli di correzione d'errore, si considerano gli effetti della lunghezza finita delle chiavi sul processo di distillazione. Inoltre, si propone un algoritmo innovativo di selezione adattiva ed in tempo reale dei dati che, sfruttando la turbolenza del canale come risorsa, permette di estendere l'applicabilitá della crittografia quantistica a nuovi livelli di rumore. Utilizzando un software per il processing classico ottimizzato per lo scenario considerato, i risultati ottenuti sono quindi analizzati e validati, dimostrando che la sicurezza quantistica dell'informazione può essere garantita in condizioni realistiche con link quantistici in spazio libero.

Capitolo 1

Introduction

Pervasive digital communication and its sensitive applications have made information security mechanisms a fundamental component of most communication networks, spanning from national security infrastructures to on-line shops, from social networks to banking platforms. Depending on the application, different security services may be required, such as confidentiality, integrity and authenticity of the exchanged data. Cryptographic mechanisms which provide these services base their security on a secret bit sequence, shared with legitimate communication parties but, in an ideal setting, completely unknown to potential adversaries. In order to establish the secrecy of this sequence, referred to as *secret key*, we have to make sure that an attacker cannot guess it with a probability higher than that of a uniformly random guess on the key space. This means that, on one hand, the secret key itself should be uniformly distributed on the key space, and that, on the other hand, eventual side information available to the eavesdropper should not provide her with a guessing advantage. These requirements establish the need for two main tools: a generator which produces uniformly random sequences and a way to estimate and compensate for the information leaked to the eavesdropper in the key generation process.

In this thesis, we investigate how the combination of quantum physics and information theory could deliver solutions at the forefront of information security, by considering two focus applications: randomness extraction as applied to quantum random number generators and classical processing algorithms for quantum key distribution (QKD). Whereas the first provides a provably secure way for producing uniformly random sequences, the second allow to establish an unconditionally secure key agreement scheme.

We concentrate on practical applications of such tools and, as for QKD, on a specific experimental scenario, that is, the one of free-space quantum links. Up to date, in fact, commercial solutions with quantum links operating over optical fibers already exist, but suffer from severe infrastructure complexity and cost overheads. On the other hand, free-space QKD would allow for higher flexibility, for both terrestrial and satellite links, whilst experiencing higher attenuation and noise at the receiver. In this work, its feasibility is investigated and proven in multiple experiments over links of different length and in different channel conditions. In particular, experimental values for the secret key rate are reported, depending on the channel losses, on the noise at the receiver and according to different security definitions. By using a full-fledged software for classical processing

tailored for this application scenario, the obtained results are analyzed and validated, showing that quantum information security can be ensured in realistic conditions with free-space quantum links.

Contributions

This thesis is mostly focused on practical applications of classical processing algorithms for quantum information security. In the following, we briefly describe its main contributions, in the order of presentation in the manuscript.

First, we implemented the randomness extractor proposed in [1] for a commercial quantum random number generator, the Quantis^{TM}, by ID Quantique. A critical discussion on its parameters and on the effects of some non-idealities is provided, together with a description of its software implementation and with the experimental evaluation of the obtained results.

Second, we provide a classification of information reconciliation protocols and a detailed analysis and performance comparison of some available solutions. The considered protocols are optimized for minimizing the error correction information leakage while still ensuring a target output bit error rate. Results of this analysis are used in [J1,C2,C3].

Third, privacy amplification against selective individual attacks is investigated. A tight bound on the information leakage is given while using multiplication by random matrices as universal hashing function [C1]. Also, the security proof with finite-key analysis of a possibly aborting protocol is provided, as an extension of the results published in [J1].

Last, the above contributions were jointly exploited in free-space quantum key distribution experiments under realistic conditions of noise and losses. A full-fledged software for experimental quantum key distribution has been implemented, including an FPGA software interface, classical processing algorithms and a network module for key distillation by communication over a classical channel. Either real-time or off-line key distillation has been performed, so that the feasibility of free-space quantum key distribution is proven in different scenarios. In particular, in [C3] we describe the results obtained with an experiment over a 50 meters free-space, indoor link, where keys were distilled in real time. In [J1], we provide the experimental results obtained via finite-key analysis for different noise levels and according to two distinct security definitions. Finally, in [P1] (to be submitted), we propose a new technique for enabling key distillation in free-space channels with harsh channel conditions by using the turbulence of the channel as a resource. The proposed technique is validated by the experimental data obtained over the 143 km free-space link between the Isles of Santa Cruz de la Palma and Tenerife.

Outline

In chapter 2, we present a family of mathematical tools, known as randomness extractors, which turn out to be of fundamental importance both for producing provably uniformly random numbers and for allowing the distillation of an unconditionally secure key in a secret key agreement scheme and, in particular, in quantum key distribution protocols. We recall a possible classification for such extractors and we concentrate on the most versatile

class, i.e., the family of *seeded* randomness extractors. By using a short uniformly random sequence, in fact, seeded extractors allow to extract a uniformly random sequence out of a weakly uniform randomness source, at the price of a compression which depends on the required uniformity level. We then consider a possible application scenario for randomness extractors, and we describe the design, the implementation and the experimental results obtained by applying such tools to a commercial quantum random number generator, the Quantis$^{\text{TM}}$, by ID Quantique.

In chapter 3, we delve into the topic of quantum key distribution. We start by giving the general system model for information-theoretic secret key agreement proposed by Maurer. Then, we classify QKD protocols according to two core components: the key distribution technique and the information coding scheme. Subsequently, two fundamental physical principles for quantum security are recalled, that is, an information-disturbance lemma and the quantum no-cloning theorem. After an overview of possible attack models, security notions covering different adversarial scenarios, spanning from purely classical to general quantum attacks, are provided. We then describe some quantum key distribution protocols, mainly BB84 and B92, and jointly introduce some practical attack strategies, such as intercept-and-resend and unambiguous state discrimination. We conclude this section by briefly reviewing photon number splitting attacks and the principles of decoy state protocols. A taxonomy of information reconciliation protocols is then proposed, and two possible practical constructions (namely, the Winnow protocol and a scheme based on LDPC codes) are detailed. Information reconciliation is, in fact, a crucial phase in the process of distilling a secret key and its application in the context of QKD undergoes different constraints with the respect to ordinary error correction protocols, as the redundancy bits are directly leaked to a potential adversary and should therefore be minimized. As a conclusion to this section, error verification is briefly described. A further application to randomness extractors is then introduced, i.e., privacy amplification in a quantum adversarial scenario. In particular, the impact of finite-key effects on the extractable secure key length is considered, as an extension to recent literature results.

In chapter 4, we describe the free-space QKD experiments that have been carried out as a consistent part of this work. We start by giving a description of the software implemented for performing the experiments. Three experiments are then detailed, covering different scenarios. We start from describing an experiment which was conducted in Palazzo della Ragione (Padova, Italy), where the B92 protocol was implemented on the top of a 50 meters, indoor, free-space quantum link. We then describe the results obtained with a free-space implementation of the efficient BB84 protocol, where finite-key analysis was used for deriving the secret key rate under different noise conditions. We conclude by detailing the results obtained by a further experiment, held between Santa Cruz de La Palma and Tenerife (Canary islands), which showed that daylight, free-space QKD is feasible even with a 143 Km free-space link, by using the turbulence of the channel as a resource.

In chapter 5, we finally draw the conclusions.

Let us conclude by remarking that, throughout this manuscript, we are going to recall

some fundamental notions of information theory and quantum mechanics, but, in general, we assume that the reader is familiar with them. We hence refer the reader to [2] and [3] for an accurate dissertation on these topics.

Capitolo 2

Randomness extraction

Random numbers are of crucial importance to several applications, spanning from cryptography to numerical simulations. Different sources may be used in order to produce random bit sequences. Nowadays, pseudo-random number generators (PseudoRNGs) are probably the most widespread technique used for accomplishing this task. Nevertheless, they are nothing but deterministic algorithms, which, based on an input random seed, produce bit sequences that appear to be random, but are in fact predictable. In fact, given the knowledge of this seed, the whole sequence is perfectly and easily reproducible: this may even be an advantage in some scenarios, such as numerical simulations, but, on the other hand, is not acceptable for cryptographic applications. Also, PseudoRNGs exhibit a periodicity, and the period is dependent on the size of the seed. Physical random number generators (PhyRNGs) overcome these limitations by exploiting the intrinsic randomness of a physical process in order to produce random bit sequences. For instance, a remarkable tool for physical random number generation is provided by quantum optics and, in particular, by the fact that a photon impinging onto an ideal semi-transparent mirror is reflected or transmitted with equal probability [4]. This phenomenon underlies, for example, the design of the Quantis device, a PhyRNG based on the laws of quantum mechanics, and henceforth referred to as a Quantum Random Number Generator (QRNG); a description of the Quantis setup can be found in §2.4.1.

Unfortunately, despite the postulated randomness of the underlying physical process and the soundness of its practical implementation, non-idealities of the hardware may translate into slight deviations from uniformity in the sequence which is output from a PhyRNG. These deviations are typically measured in terms of bias and correlations. A powerful tool to cope with these issues are the so-called *randomness extractors*, which, by leveraging information-theoretic proofs, allow one to distill from the raw PhyRNG output sequence a shorter random string with better statistical properties. In particular, the mathematical framework of randomness extraction allows for a direct, analytic proof, unlike traditional empirical assessments of randomness, that are based on the assumption that if a given sequence passes a series of statistical tests, then it is likely that the sequence itself is random (see, for instance, [5]).

While in this chapter we focus on randomness extractors for processing the output of a PhyRNG, their use can be extended to a quantum adversarial scenario, as for the

privacy amplification phase in a quantum key distribution scheme. In the following, we therefore mention quantum-resilient extractors (i.e., randomness extractors in the presence of quantum side information), but, for their application, we refer the reader to chapter 3.

In section 2.2 we provide a general classification of randomness extractors. Then, in section 2.3 we briefly describe and compare a few possible practical constructions. Finally, in section 2.4 we detail the practical implementation of an efficient randomness extractor as applied to a specific QRNG, i.e., the Quantis$^{\mathrm{TM}}$.

2.1 Preliminary definitions and problem statement

While considering the process of generating randomness, as a first step we need the definition of what a *source* of random values is. In this work we refer to binary sources, and we hence give the following definition.

Definition 1. (Binary source). A n-binary source $\bar{X}$ on $\{0,1\}^n$ is a discrete random variable taking values in $\{0,1\}^n$ according to the probability mass distribution $p_{\bar{X}}$.

Please note that a n-binary source can be seen as a collection of n random variables $\{X_i\}$, $i = 1, \ldots, n$, each one taking values in the binary alphabet $\{0,1\}$, and whose joint probability distribution is denoted by $p_{\bar{X}} = p_{X_1 X_2 \ldots X_n}$.

Now, since we are interested in evaluating how much "random" a source is, we need to define our target, i.e., a *perfectly random* n-binary source, and measure the distance of the actual source from this ideal, theoretical model. We say that a n-binary source $\bar{X}$ is perfectly random if it is made of a set of i.i.d. (independent and identically distributed) uniform random variables $X_1, \ldots, X_n$. More formally, we say that a source $\bar{X} = [X_1, \ldots, X_n]$ is perfectly random if it is uniform on $\{0,1\}^n$, that is

$$p_{\bar{X}}(\bar{x}) = 2^{-n}, \quad \forall\ \bar{x} \in \{0,1\}^n. \tag{2.1}$$

In particular, we denote the uniform probability mass distribution over $\{0,1\}^n$ by U_n. It should be noted that condition (2.1) implies that the single random variables $\{X_i\}_{i\in[1,n]}$ are uniform and jointly independent.

Now, we need a tool for evaluating the deviation of a given probability distribution with respect to the ideal uniform and independent one. An adequate measure of the deviation between two probability distributions is provided by the following function.

Definition 2. (Statistical distance). The statistical distance between two probability distributions $p_{\bar{X}}$ and $p_{\bar{X}'}$ over the same alphabet $\mathcal{X}$ is defined as

$$\delta(p_{\bar{X}}, p_{\bar{X}'}) \triangleq \frac{1}{2} \sum_{\bar{x}\in\mathcal{X}} \left| p_{\bar{X}}(\bar{x}) - p_{\bar{X}'}(\bar{x}) \right| .$$

We say that two distributions $p_{\bar{X}}$ and $p_{\bar{X}'}$ over the same domain are ε-close if $\delta(p_{\bar{X}}, p_{\bar{X}'}) \leq \varepsilon$. In particular, we say that the n-binary source $\bar{X}$ is ε-uniform if it is ε-close to U_n. The statistical distance also has an operational interpretation, that is, it can be seen as the

maximum probability by which two probability distributions can be distinguished, as formalized in the following lemma.

Lemma 1. *[6, Proposition 2.1.1] Let $p_{\bar{X}}$ and $p_{\bar{X}'}$ be two probability distributions on the same alphabet $\mathcal{X} = \{a_1, \ldots, a_N\}$ such that their statistical distance is $\varepsilon = \delta(p_{\bar{X}}, p_{\bar{X}'})$. Then there exists a joint distribution $p_{\bar{X}\bar{X}'}$, with marginal distributions $p_{\bar{X}}$ and $p_{\bar{X}'}$, such that the following inequality holds:*

$$P[\bar{X} \neq \bar{X}'] \leq \varepsilon, \tag{2.2}$$

where

$$P[\bar{X} \neq \bar{X}'] = \sum_{(\bar{x},\bar{x}') \in \mathcal{X}\times\mathcal{X}} p_{X\bar{X}'}(\bar{x}, \bar{x}')\chi(\bar{x}, \bar{x}'), \tag{2.3}$$

and $\chi(\bar{x}, \bar{x}') = \begin{cases} 1, & \text{if } \bar{x} \neq \bar{x}', \\ 0, & \text{if } \bar{x} = \bar{x}'. \end{cases}$

Dimostrazione. Let $p_i = p_{\bar{X}}(a_i)$ and $q_i = p_{\bar{X}'}(a_i)$ for every $i \in [1, N]$. Then, we can design a joint distribution such7 that

$$p_{\bar{X}\bar{X}'}(a_i, a_j) = \begin{cases} \min(p_i, q_i), & \text{if } i = j \\ p_{ij}, & \text{otherwise} \end{cases} \tag{2.4}$$

with p_{ij} chosen such that $\sum_{i=1}^{N} p_{ij} = q_j$ and $\sum_{j=1}^{N} p_{ij} = p_i$. The existence of such $\{p_{ij}\}$ is ensured by the balance condition $\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} q_i$, as explained in [7]. Then

$$\begin{aligned}
P[\bar{X} \neq \bar{X}'] &= 1 - \sum_{i=1}^{N} p_{\bar{X}\bar{X}'}(a_i, a_i) && (2.5)\\
&= 1 - \sum_{i=1}^{N} \min(p_i, q_i) && (2.6)\\
&= \frac{1}{2}\left(\sum_{i=1}^{N}[q_i - \min(p_i, q_i)] + \sum_{i=1}^{N}[p_i - \min(p_i, q_i)]\right) && (2.7)\\
&= \frac{1}{2}\left(\sum_{i:q_i>p_i} |q_i - p_i| + \sum_{i:p_i>q_i} |q_i - p_i|\right) && (2.8)\\
&\leq \frac{1}{2}\sum_{i=1}^{N} |q_i - p_i| && (2.9)\\
&= \delta(p_{\bar{X}}, p_{\bar{X}'}) \leq \varepsilon && (2.10)
\end{aligned}$$

□

The result of the previous lemma can be extended so that for any random variable $\bar{X}$ distributed according to $p_{\bar{X}}$ it is possible to define a random variable $\bar{X}'$ on the same probability space and distributed with $p_{\bar{X}'}$ such that (2.2) holds. This requires the additional

hypothesis that the probability space underlying the joint distribution of $\bar{X}$ and $\bar{X}'$ has a fine enough resolution.[1]

Lemma 2. *Let $p_{\bar{X}}$ and $p_{\bar{X}'}$ be two probability distributions on the same alphabet $\mathcal{X} = \{a_1, \ldots, a_N\}$ such that their statistical distance is $\varepsilon = \delta(p_{\bar{X}}, p_{\bar{X}'})$. Then, for every $\bar{X}$ defined on a sufficiently resolved probability space and distributed according to $p_{\bar{X}}$, there exists some $\bar{X}'$ distributed according to $p_{\bar{X}'}$ such that*

$$P[\bar{X} \neq \bar{X}'] \leq \varepsilon, \tag{2.11}$$

where $P[\bar{X} \neq \bar{X}']$ is defined as in (2.3).

Dimostrazione. As a first step, given $p_i = p_{\bar{X}}(a_i)$ and $q_i = p_{\bar{X}'}(a_i)$ for every $i \in [1, N]$, design the joint probability distribution $p_{\bar{X}\bar{X}'}(a_i, a_j)$ as in (2.4). Then, given the random variable $\bar{X}$, defined on the probability space[2] $(\Omega, \mathcal{F}, P)$, let $A_i = \bar{X}^{-1}(a_i) \in \mathcal{F}$, for each $i \in [1, N]$, that is, A_i is the event corresponding to the outcome a_i. We say that the probability space is sufficiently resolved if, for every i, there exist $A'_{i1}, \ldots, A'_{iN} \subseteq A_i$, with $A'_{ij} \cap A'_{jk} = \emptyset$ for $j \neq k$, such that $P[A'_{ij}] = p_{ij}$. Under this hypothesis, we can define a random variable $\bar{X}' : \Omega \to \mathcal{X}$, with

$$\bar{X}'(\omega) = \begin{cases} a_j, & \omega \in \bigcup_{i=1}^{N} A'_{ij} \\ \text{arbitrary}, & \omega \in \Omega / \bigcup_{i=1}^{N} A'_{ij}. \end{cases} \tag{2.12}$$

It is then easy to see that the joint probability distribution of $\bar{X}$ and $\bar{X}'$ is $p_{\bar{X}\bar{X}'}$, which, in the proof of lemma 1, was shown to imply condition (2.11). □

These lemmas have a relevant practical consequence, in that, if an application is proved to work well with random sequences produced by $\bar{X}$, then it is supposed to work well also with $\bar{X}'$, given that $\varepsilon = \delta(p_{\bar{X}}, p_{\bar{X}'})$ is sufficiently small.

Lemma 3. *[1] Let Π be a stochastic process that takes as input a random value $\bar{X}$ and may fail with probability $p_{\text{fail}}(\Pi(\bar{X}))$. If the input $\bar{X}$ is replaced by $\bar{X}'$, then the failure probability can increase by at most $\varepsilon = \delta(p_{\bar{X}}, p_{\bar{X}'})$, i.e.,*

$$p_{\text{fail}}(\Pi(\bar{X}')) \leq p_{\text{fail}}(\Pi(\bar{X})) + \varepsilon \ .$$

At this point, we have the definition of random source, the one of a perfectly random source and a mathematical metric, the statistical distance, to measure the deviation of a given source from the ideal and uniform one. We are then ready to formally define the operational problem of randomness extraction.

Definition 3. (Extraction problem). Let $\bar{X}$ be a n-binary, γ-uniform source. A randomness extractor Ext, which takes as input a single realization of $\bar{X}$ and possibly further inputs produced by a source $\bar{Z}$ is a function such that its ℓ-bit output $\bar{Y} = Ext(\bar{X}, \bar{Z})$ is ε-uniform, with $\ell < n$, $\varepsilon < \gamma$ and $\varepsilon \ll 1$ as small as required by further applications.

[1] the notion of fine enough resolution is provided in the proof of the lemma.

[2] we recall that Ω is the sample space, $\mathcal{F}$ the σ-algebra of events and P the probability measure.

The nature of the $\bar{Z}$ will be clarified in the following section; in particular, it could consist of some additional, external randomness (seeded extractor) or of multiple, independent randomness sources (multiple, independent sources extractor). Intuitively, given a weak randomness source $\bar{X}$, for each n-bit sample we want to produce an ℓ-bit output such that the distribution of the output is ε-uniform. With a different perspective, according to this definition the extractor output is such that a potential adversary using an optimal strategy[3] cannot guess it with an advantage greater than ε as compared with a uniformly random guess, that is, $p_{\text{guess}} \leq 2^{-\ell} + \varepsilon$.[4]

With this problem in mind, it is of crucial interest to evaluate the number ℓ of bits that can be extracted from a source so that the extractor output is ε-uniform. Intuitively, this quantity depends on the nature of the extractor and on the statistical description of the source. In particular, if the input source produces i.i.d. samples, as a consequence of the asymptotic equipartition property it can be proven (see e.g., [8]) that the amount of extractable randomness is asymptotically equal to the Shannon entropy, which is defined as follows:

Definition 4. (Shannon Entropy) Let $\bar{X}$ be a source with probability mass distribution $p_{\bar{X}}$. The Shannon entropy of $\bar{X}$ is defined as

$$H(\bar{X}) \triangleq \sum_{\bar{x}\in\{0,1\}^n} -p_{\bar{X}}(\bar{x}) \log_2(p_{\bar{X}}(\bar{x})). \tag{2.13}$$

Unfortunately, this entropy measure is not sensitive enough to deviations from uniformity. Let us consider the following clarifying example.

Example 1. *Consider the following class of distributions on $\mathcal{X} = \{0,1\}^n$:*

$$\mathcal{P} = \left\{ p : \mathcal{X} \to \mathbb{R}^+ \text{ such that } \exists\ a \in \mathcal{X} : p(a) = \frac{1}{2} \wedge\ p(a') = \frac{1}{2^n}, \forall a' \neq a \right\}, \tag{2.14}$$

that is, all $p \in \mathcal{P}$ take some value a with probability $\frac{1}{2}$ and any other value $a' \neq a$ with probability $\frac{1}{2^n}$. It is easy to see that, for every $\bar{X}$ distributed according to p,

$$H(\bar{X}) \quad = \quad -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - 2^{n-1}\left\{\frac{1}{2^n}\log_2\left(\frac{1}{2^n}\right)\right\} = \frac{1}{2} + \frac{n}{2}. \tag{2.15}$$

In the above example, for a random choice of $p \in \mathcal{P}$, $H(\bar{X})$ linearly increases with n and can get arbitrarily high, but, in fact, an adversary can always guess $\bar{X}$ with a probability at least 1/2. Hence, given that $\{X_i\}_{i\in[1,n]}$ are not i.i.d., $H(\bar{X})$ does not properly take into account the worst case, but rather considers the average behavior. On the contrary, this is exactly the rationale behind the definition of *min-entropy* of a source, which in fact captures the maximum guessing probability for an adversary, which is actually what we are interested in for the case of randomness extraction. Let us now give a formal definition.

[3] the optimal strategy consists in guessing the most probable event.

[4] please note that this bound may be loose, as the statistical distance takes into account *all* the deviations from uniformity, whereas the guessing probability (assuming the optimal attacker's strategy) considers just the one of the most probable element, if any.

Definition 5. (Min-entropy). Let $\bar{X}$ be a source with probability mass distribution $p_{\bar{X}}$. The min-entropy of $\bar{X}$ is defined as

$$H_{min}(\bar{X}) \triangleq -\log_2 \max_{\bar{x}\in\{0,1\}^n} \{p_{\bar{X}}(\bar{x})\}. \tag{2.16}$$

This definition implies that if $\bar{X}$ is a binary source, then $H_{min}(\bar{X})$ is equal to the maximum k such that for every $\bar{x} \in \bar{X}$, $p_{\bar{X}}(\bar{x}) \leq 2^{-k}$, that is, "it is directly related to the probability of correctly guessing the value of $\bar{X}$ using an optimal strategy" [9]. In particular, in the following sections we report some results on how the min-entropy provides a bound on the amount of random bits that can be extracted out of a randomness source.

If we go back to example 1, we get $H_{min}(\bar{X}) = 1$, which is exactly the maximum amount of randomness we can deterministically extract from $\bar{X}$ by just knowing that its distribution lies in $\mathcal{P}$. In fact, for this particular case, the attacker has no guessing advantage only when trying to guess one of the two events $\{a' = a\}$ and $\{a' \neq a\}$, which both have probability $\frac{1}{2}$ and can therefore be represented by just one bit of information.

Finally, we introduce the notion of collision entropy which provides a tighter bound on the amount of extractable randomness for some specific extractors.

Definition 6. (Collision entropy) Let $\bar{X}$ be a source with probability mass distribution $p_{\bar{X}}$. The collision entropy of $\bar{X}$ is defined as

$$H_2(\bar{X}) \triangleq -\log_2 \sum_{\bar{x}\in\{0,1\}^n} \{p_{\bar{X}}(\bar{x})\}^2. \tag{2.17}$$

It should be noted that all the entropy measures introduced above can be seen as particular cases of a more general one, that is, the Rényi entropy of order α, which is defined in the following.

Definition 7. (Rényi entropy of order α) For $\alpha > 0,\ \alpha \neq 1$, the Rényi entropy of order α of a n-binary source $\bar{X}$ is defined as

$$H_\alpha(\bar{X}) = \frac{1}{1-\alpha} \log_2 \sum_{\bar{x}\in\{0,1\}^n} [p_{\bar{X}}(\bar{x})]^\alpha. \tag{2.18}$$

In particular, as it can be proven (see, e.g., [10]), the Shannon entropy can be seen as the limit for $\alpha \to 1$ of H_α and the min-entropy as the limit for $\alpha \to \infty$ of H_α, whereas the collision entropy is the Rényi entropy of order 2.

The following proposition provides a comparison between the proposed entropy definitions, based on the fact that $H_\alpha(\bar{X}) \geq H_\beta(\bar{X})$ for $0 \leq \alpha \leq \beta$ (a proof can be found in [10, Proposition 2.4]).

Proposition 1. *For every source $\bar{X}$, the following inequalities holds*

$$H_{\min}(\bar{X}) \leq H_2(\bar{X}) \leq H(\bar{X}), \tag{2.19}$$

and the equality holds if and only if $\bar{X}$ is uniform or almost surely constant.

Let us conclude by noting that in order to compute $H_{\min}(\bar{X})$ and $H_2(\bar{X})$, the probability distribution $P_{\bar{X}}$ of the source $\bar{X}$ has to be known a priori. Unfortunately, in practical applications such probability distribution is not available, but it should be estimated relying on finite bit sequences. Hence, the reliability of such an estimate should be taken into account and, given an error probability μ of the estimate, we now define two quantities, called *smooth entropies*, which embed this statistical error in their definition.

Definition 8. [1] For any $\mu \geq 0$, the *smooth min-entropy* $H_{\min}^{\mu}$ and the *smooth collision entropy* H_2^{μ} of a random variable X are defined by

$$H_{\min}^{\mu}(\bar{X}) = \max_{P_{\bar{X}'} \in \mathcal{B}^{\mu}(P_{\bar{X}})} H_{\min}(\bar{X}') \tag{2.20}$$

$$H_2^{\mu}(\bar{X}) = \max_{P_{\bar{X}'} \in \mathcal{B}^{\mu}(P_{\bar{X}})} H_2(\bar{X}') \ . \tag{2.21}$$

where $\mathcal{B}^{\mu}(P_{\bar{X}})$ is the set of all distributions $P_{\bar{X}'}$ which have at most distance μ from $P_{\bar{X}}$.

It should be noted that these two entropy measures are essentially equivalent. This is stated more formally in the following lemma.

Lemma 4. *[1] For any $\mu \geq 0$ and $\mu' > 0$,*

$$H_{\min}^{\mu}(X) \leq H_2^{\mu}(X) \leq H_{\min}^{\mu+\mu'}(X) + \log_2 \frac{1}{\mu'} \ .$$

In the remainder of this chapter, in order to simplify the analysis and the notation, we are not explicitly considering entropy smoothing, though most cited results immediately generalize to smooth entropies. In §2.4.3, however, the impact of the entropy estimation error on the performance of the Quantis extractor is evaluated and, hence, the smooth collision entropy is considered.

2.2 Types of extractors

Randomness extractors can be distinguished in three main categories, depending on their structure: deterministic extractors, seeded extractors and extractors from multiple independent sources. In the following sections, we describe these categories while presenting some known results from the literature. In particular, we refer the reader to [11, 12].

2.2.1 Deterministic extractors

Deterministic extractors represent the simplest family of randomness extractors. In fact, they just process the weakly-random input sequence, without needing any additional input. Let us start from giving a definition of these mathematical tools.

Definition 9. (Deterministic extractor) Let $n, \ell \in \mathbb{N}$ be such that $\ell \leq n$ and let $\varepsilon \geq 0$ be a parameter. Let $Ext : \{0,1\}^n \to \{0,1\}^{\ell}$ be a function and $\bar{X}$ be a source over $\{0,1\}^n$. Then Ext is an ε-extractor for $\bar{X}$ if the distribution of $Ext(\bar{X})$ is ε-close to U_{ℓ}. Furthermore, given a class $\mathcal{C}$ of sources over $\{0,1\}^n$, Ext is an ε-extractor for $\mathcal{C}$ if the distribution of

$Ext(\bar{X})$ is ε-close to U_ℓ for every $\bar{X} \in \mathcal{C}$.

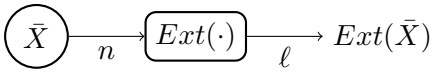


Figura 2.1: Block diagram for a deterministic extractor.

It can be proven that deterministic extractors are able to produce uniform randomness when the class $\mathcal{C}$ of possible sources has a specific structure. As an example, deterministic extractors exist for the following classes of sources:

1. *Von Neumann sources*

   These sources consist of a sequence of independent but uniformly biased boolean random variables $X_1, X_2, \ldots, X_n \in \{0, 1\}$, that is:

   (a) (Identical bias, $p \in]0, 1[$)

   $$p_{X_i}(1) = p, \quad \forall i \in [1, n]$$

   (b) (Independence)

   $$p_{X_1 X_2 \ldots X_n}(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_{X_i}(x_i)$$

   Please note that the well-known Von-Neumann algorithm, which divides the input string in pairs and maps the pair 01 to 0 and the pair 10 to 1, while discarding the pairs 00 and 11, is an example of deterministic extractor.

2. *Independent-bit sources*

   These sources are an extension of the previous ones. In particular, we now let different sources to have different biases. More specifically:

   (a) (Non-uniform bias, $p_i \in]0, 1[$)

   $$p_{X_i}(1) = p_i, \quad \forall i \in [1, n]$$

   (b) (Independence)

   $$p_{X_1 X_2 \ldots X_n}(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_{X_i}(x_i)$$

   It can be shown that when we take a parity of p bits from such an independent-bit source, the result approaches an unbiased coin flip exponentially fast in p. This means that we can design an extractor which asymptotically extracts a uniformly random sequence out of a weakly random one, but this comes at the price of a rate reduction of a factor p.

Nevertheless, as soon as we drop the *independence* assumption, deterministic extractors are no longer able to efficiently produce perfectly random sequences. For instance, the authors of [13] have shown that deterministic extractors *do not* exist for the following class of sources, that are considered as a generalization of Von Neumann sources, and are often referred to as *Santha-Vazirani* sources:

Definition 10. (Unpredictable-bit sources, $\text{UPB}_{n,\delta}$). Given $n \in \mathcal{N}$ and $\delta > 0$, a $\text{UPB}_{n,\delta}$ source is a collection of binary sources $\{X_i\}$ such that

$$\forall\ i,\ x_1, \ldots, x_n \in \{0,1\},\ \delta > 0 :$$
$$\delta \leq P[X_i = 1 | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \leq 1 - \delta$$

More specifically, the non existence of deterministic extractors for this kind of sources is stated in the following proposition.

Proposition 2. *[14] For every $n \in \mathbb{N}$, $\delta > 0$, and a fixed extraction function $Ext : \{0,1\}^n \to \{0,1\}$ there exists a $UPB_{n,\delta}$ source $\bar{X}$ such that either $P[Ext(\bar{X}) = 1] \leq \delta$ or $P[Ext(\bar{X}) = 1] \geq 1 - \delta$. Hence, there is no ε-extractor for the class $UPB_{n,\delta}$.*

The above example shows that "there are families of sources that are very structured and still do not allow deterministic extraction" [12]. Hence the need for non-deterministic extractors, that will be described in the following sections.

As a conclusion to this section, and for the sake of completeness, we list a few more general families of sources which are considered in the literature on deterministic extraction, as reported in [12].

- *Affine sources* are distributions that are uniform over some affine subspace of $\mathbb{F}_q^n$ (space composed by vectors of n elements in the finite field $\mathbb{F}_q$); the min-entropy of these sources coincides with the dimension of this subspace.
- *Feasibly generated sources* - family of sources that are specified by placing limitations on the process that generates the source (e.g., generated by a finite Markov chain).
- *Feasibly recognizable sources* - family of sources that are uniform over sets of the form $\{\bar{x} : f(\bar{x}) = \bar{v}\}$ for a fixed $\bar{v}$ and with f belonging to some specified class ([15]).

2.2.2 Seeded extractors

As was shown in the previous section, deterministic extraction is not suitable for all types of sources. We will now consider a much more general class of sources, which rely on a single constraint on the min-entropy. We will refer to this kind of sources as *k-sources*.

Definition 11. (k-source). A random variable $\bar{X}$ is a k-source if $H_{\min}(\bar{X}) \geq k$, i.e., if $p_{\bar{X}}(\bar{x}) \leq 2^{-k}$ for any $\bar{x} \in \{0,1\}^n$.

Examples of k-sources

- *oblivious bit-fixing sources* - k random independent uniform bits, $n-k$ fixed bits (in an arbitrary order).
- *adaptive bit-fixing sources* - k random independent uniform bits, $n-k$ bits that depend arbitrarily on the first k bits.
- $UPB_{n,\delta}$ *(see §2.2.1)* - k-sources with $k = \log(1/(1-\delta)^n) = \Theta(\delta n)$
- *flat k-sources* - sources with uniform distribution on a set $S \subset \{0,1\}^n$, with $|S| = 2^k$ (generalization of adaptive bit-fixing sources).

Given this definition, we are now ready to define a seeded randomness extractor.

Definition 12. (Seeded extractor) A function $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ is a (k,ε)-extractor if for every k-source $\bar{X}$, the distribution of $Ext(\bar{X},\bar{S})$ is ε-close to uniform, where $\bar{S} \sim U_d$ and is independent of $\bar{X}$. In particular, we say that Ext is a (k,ε)-strong extractor if $Ext'(\bar{x},\bar{s}) = (Ext(\bar{x},\bar{s}),\bar{s})$ is a (k,ε)-extractor.

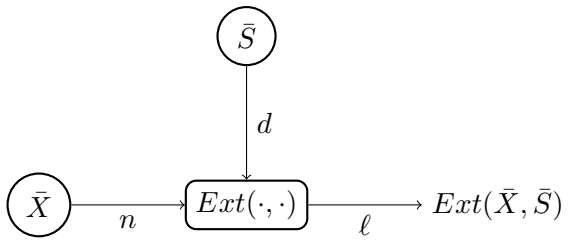


Figura 2.2: Block diagram for a seeded extractor.

It is of course interesting to evaluate what are the optimal parameters for this kind of extractors. The following theorem states a result in this sense.

Theorem 1. *[16] For every $n \in \mathbb{N}$, $k \in [0,n]$ and $\varepsilon > 0$, there exist (k,ε)-extractors with:*

- *seed length* $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$
- *output length* $\ell = k + d - 2\log(1/\varepsilon) - O(1)$

It is then natural to define the amount of randomness that a seeded extractor is able to produce given a k-source and a uniformly random seed, or, equivalently, how much randomness is lost in the extraction process; this gap is captured by the so-called *entropy loss*.

Definition 13. (Entropy Loss). The entropy loss of a seeded randomness extractor $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ is intuitively defined as $\Delta := k + d - \ell > 0$.

From the previous theorem, it follows that there exist extractors such that $\Delta = 2\log(1/\varepsilon) + O(1)$, but it should be noted that we cannot exclude the existence of better extractors with smaller Δ. Trivially, the lower is the entropy loss, the higher is the extraction efficiency. It should be noted that theorem 1 establishes an asymptotic lower bound on the minimum seed length and an asymptotic upper bound on the maximum output length, but does not provide a constructive method for designing an optimal extractor. Some close-to-optimal constructions are introduced in section 2.3.

On the nature of side information.

Up to this point, we have required that given a k-source and a uniformly random seed, the distribution of the output of a (k, ε)-extractor, $Ext(\bar{X}, U_d)$, is ε-close to uniform, that is

$$H_{\min}(\bar{X}) \geq k \Rightarrow Ext(\bar{X}, U_d) \overset{\varepsilon}{\sim} U_\ell. \tag{2.22}$$

For some applications, however, it is necessary to take into account also the *side information*, that is the information about the input which is accessible to a potential adversary. In particular, side information should be considered when using randomness extractors in the context of privacy amplification (as for Quantum Key Distribution, see chapter 3) or simply when applying two extractors in succession to the same input $\bar{X}$ (see §2.4.3 for a discussion on the impact of block-wise extraction). In this setting, criterion (2.22) should be rewritten as

$$H_{\min}(\bar{X}|E) \geq k \Rightarrow Ext(\bar{X}, U_d) \overset{\varepsilon}{\sim} U_\ell \text{ conditioned on E.} \tag{2.23}$$

As stated in [9], "the relationship between criterion (2.22) and criterion (2.23) depends on the physical nature of the side information E, i.e., whether E is represented by the state of a classical or a quantum system". In fact, in the case of purely classical side information, the two criteria are almost equivalent up to some small factor [17], while in the quantum case, criterion (2.23) is strictly stronger than (2.22), and extractors that fulfill (2.22) do not necessarily verify (2.23) (a counterexample can be found in [18]). This has the fundamental implication that, in the presence of a quantum adversary, which has access to some side quantum information, the adoption of criterion (2.23) is mandatory in order to ensure secure randomness extraction. In particular, an extractor fulfilling criterion (2.23) (with E being a quantum state), is referred to as a *quantum-resilient* extractor.

For defining this type of extractor, we first have to define the min-entropy of a classical random variable conditioned on the quantum side information available to the adversary.

Definition 14. [19] Let $\rho_{\bar{X}E}$ be a cq-state, that is, a bipartite density operator describing the joint state of the classical value $\bar{X}$ and of the quantum system E:

$$\rho_{\bar{X}E} = \sum_{\bar{x} \in \{0,1\}^n} p_{\bar{\bar{X}}}(\bar{x}) |\bar{x}\rangle\langle\bar{x}| \otimes \rho_{E|\bar{X}=\bar{x}}. \tag{2.24}$$

The min-entropy of $\bar{X}$ conditioned on E (evaluated for $\rho = \rho_{\bar{X}E}$) is defined as

$$H_{\min}(\bar{X}|E)_\rho = -\log_2 p_{\text{guess}}(\bar{X}|E) \tag{2.25}$$

where $p_{\text{guess}}(\bar{X}|E)$ is the average probability of guessing the value of $\bar{X}$ correctly using an optimal strategy with access to E.

We are now ready to define a quantum-resilient extractor:

Definition 15. [19] A seeded randomness extractor, $Ext(\bar{X}, \bar{S})$, is a (k, ε)-strong quantum resilient extractor if for all cq-states $\rho_{\bar{X}E}$ with $H_{\min}(\bar{X}|E)_\rho \geq k$ we have

$$\mathbb{E}_{\bar{S}}[\|\rho_{Ext(\bar{X},\bar{S})E} - \omega_{\bar{X}} \otimes \rho_E\|_1] \leq \varepsilon \tag{2.26}$$

where $\omega_{\bar{X}}$ is the state corresponding to a uniformly distributed $\bar{X}$ and $\mathbb{E}_{\bar{S}}[\cdot]$ denotes the expectation value over a uniform choice of the seed $\bar{S}$.

In this thesis, we will consider a specific instance of randomness extraction in a quantum adversarial scenario, namely, the privacy amplification phase of a quantum key distribution protocol. For further details, we therefore refer to §3.6 and, for a practical solution, to §4.2 and §4.3. In the remainder of this chapter, however, we restrict the analysis to the classical case, as we are concentrating on how to improve the quality of randomness of the classical output of a QRNG to which the adversary has no physical access.

2.2.3 Extractors from multiple independent sources

A further generalization of seeded extractors is the one of extractors from multiple independent sources. Here the idea is to replace the requirement that the seed is uniformly distributed by the weaker requirement that it has a sufficiently large min-entropy. This requirement could be achieved, for instance, in the case of two independent sources, one of which is regarded as a weakly random seed for the randomness extractor which receives as input a sequence originated by the other one. The extension to t independent sources is then straightforward.

Definition 16. (Extractors for independent sources). A function $Ext : (\{0,1\}^n)^t \to \{0,1\}^m$ is a (k, ε)-t-source extractor if for every t independent random sources $\bar{X}_1, \ldots, \bar{X}_t$ such that, $\forall\ i = [1, \ldots, t]$, $H_{min}(\bar{X}_i) \geq k$, the distribution of $Ext(\bar{X}_1, \ldots, \bar{X}_t)$ is ε-close to U_ℓ.

We stress that, as said in [12], extractors from multiple independent sources could be seen as particular cases of deterministic extractors as well, as they do not require external randomness, but just weakly random sources. We will not delve deeper into the description of this kind of extractors, but we just recall the current state-of-the-art[5] for 2-source extractors in terms of min-entropy threshold (i.e., the minimum min-entropy that each source should have); the result is due to Bourgain [20] and fixes this threshold at $k = (1/2 - \alpha)n$ for some small constant α.

[5] to the best of the author's knowledge.

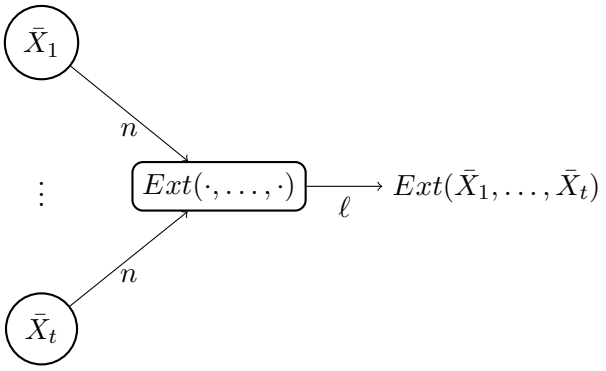


Figura 2.3: Block diagram for extractors from multiple independent sources.

Finally, let us recall the following result, which shows that a very simple extractor from multiple independent sources, that is, the extractor which outputs the XOR of all the input bits, reduces the distance from uniform of the input exponentially in the number of sources. In particular, this result will be used in §2.4.4 as applied to the generation of the seed for the Quantis$^{\text{TM}}$ extractor.

Proposition 3. *[1, Lemma 4] Let $X_1, \ldots X_t$ be t bits produced by t independent random sources such that, for some $\xi > 0$ and for all $i \in [1, t]$,*

$$\delta(X_i, U_1) \leq \xi.$$

Then the bit $W = \sum_i X_i \mod 2$ satisfies

$$\delta(W, U_1) \leq \frac{1}{2}(2\xi)^t \ .$$

Dimostrazione. A simple calculation shows that, for two independent bits X_1 and X_2 with $\xi_i = \delta(X_i, U_1)$, and with $W = X_1 + X_2 \mod 2$,

$$\delta(W, U_1) = 2\xi_1\xi_2 \ .$$

The claim then follows by recursive application of this rule. □

2.3 Practical constructions for seeded extractors

As mentioned in section 2.2, seeded extractors are the most versatile category of randomness extractors, since they can efficiently cope with both bias and correlation in the input sequence. In this section we therefore overview some fundamental results and practical constructions of such functions.

Given an n-bit input $\bar{x}$ produced by a source $\bar{X}$ and $Ext(\cdot, \bar{s})$ being the extraction function chosen uniformly at random from a class of possible functions according to an input seed $\bar{s}$ generated by a source $\bar{S}$ with distribution $P_{\bar{S}}$, we write the input-output relation of the seeded extractor as

$$\bar{y} = Ext(\bar{x}, \bar{s}). \tag{2.27}$$

Assuming that the source $\bar{X}$ has min-entropy per bit $H^b_{\min}(\bar{X})$ and given that the ℓ-bit output $\bar{y}$ is distributed according to $P_{\bar{Y}}$, we point out three desirable requirements:

1. *minimization of the distance from uniform of the output*: we want $\delta\big(P_{\bar{Y}\bar{S}}, U_\ell \times P_{\bar{S}}\big)$ to be arbitrarily low;

2. *maximization of the extraction efficiency*: $\eta_{\text{ext}} = \ell/n$ should be as close as possible to $H^b_{\min}(\bar{X})$;

3. *high computational efficiency*: the computational overhead due to the randomness extraction should be acceptable;

It is reasonably understood that, for a fixed n, the first two objectives are conflicting: the lower the required distance from uniform, the lower the corresponding extraction efficiency; this trade-off is shown in figure 2.4 for universal$_2$ hash functions (see §2.3.1).

In this section we introduce some available constructions for seeded randomness extraction, and, in particular, we focus on two significant schemes: almost 2-universal hashing and Trevisan's extractor. In order to provide an overview of possible solutions, we also report two tables: the first, taken from [19], provides some references for different randomness extraction schemes, either with classical or with quantum resiliency; the second one, taken from a presentation of Thomas Vidick (QCrypt2011[6]), schematically shows the length of the seed and the length of the corresponding output for the some randomness extractors with quantum resiliency (please note that these results extend smooth entropies as well).

	Resiliency	
	classical	quantum
2-universal hashing	[21, 22]	[23, 24]
Almost 2-universal hashing	[25]	[26]
δ-biased masking	[27]	[28]
Trevisan's extractor	[29]	[9, 30, 31]
Sample-then-extract	[32]	[33]

Tabella 2.1: Practical randomness extractors: classical and quantum resiliency.

2.3.1 (Almost) 2-Universal hash functions

The notion of universal$_2$ hash function[7] was first introduced by Carter and Wegman in their seminal work [34], dating back to 1979. Since then, universal hash functions have found many applications (e.g., in databases, in message authentication schemes, etc.) and, in particular, they are extensively used in the privacy amplification phase of a secret key agreement protocol (see, e.g., [35, 21]).

[6] available at `http://www.qcrypt2011.ethz.ch/programme/slides/vidick.ppsx`

[7] for conciseness we drop the prefix 2- from now on.

Construction	d	ℓ	Ref.
2-Universal hashing	n	$H_{\min}(\bar{X}\|E)$	[23]
Almost 2-Universal hashing	ℓ	$H_{\min}(\bar{X}\|E)$	[26]
δ-biased masking	n	$H_{\min}(\bar{X}\|E)$	[28]
1-bit extractors	$\log(n)$	1	[17]
Trevisan's extractor	$\log^3(n)$	$H_{\min}(\bar{X}\|E)$	[9, 30, 31]

Tabella 2.2: Quantum-resilient randomness extractors: seed and output length given an n-bit input.

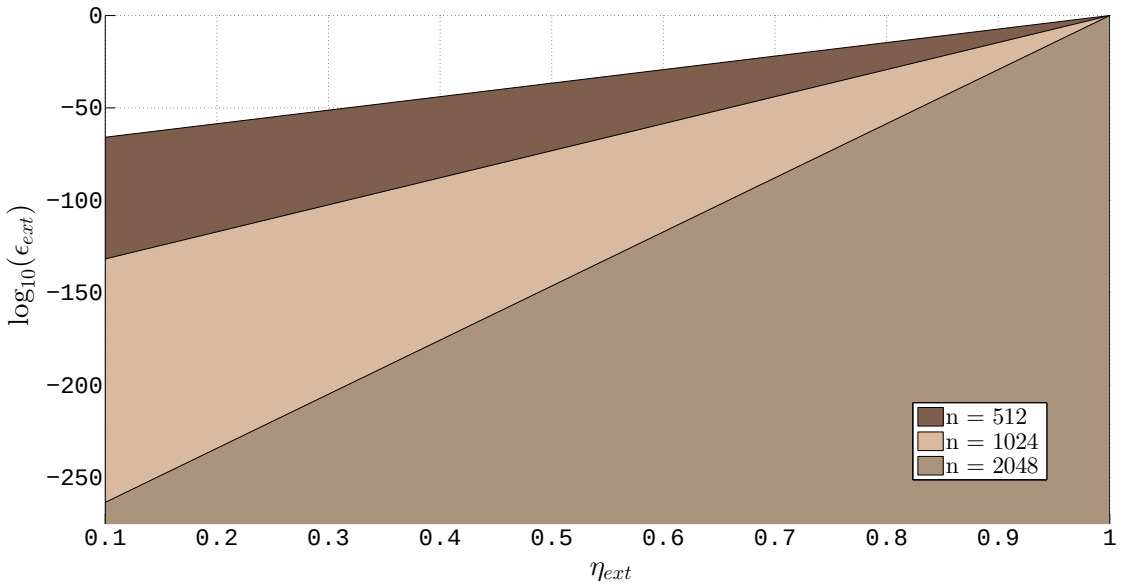


Figura 2.4: Admissible regions for the distance from uniform of the output to an extractor based on 2-universal hash functions, given the upper-bound provided by the leftover hashing lemma (Lemma 5, see §2.3.1); the regions are plotted as a function of the extractor efficiency η_{ext}, for different values of the input block size n and given $H^b_{\min}(\bar{X}) = 0.95$.

Intuitively, a class $\mathcal{H}$ of hash functions from $\mathcal{X}$ to $\mathcal{Y}$ is universal if no pair of distinct inputs is mapped to the same element by more than $1/|\mathcal{Y}|$-th of the functions. This constraint can be relaxed, by requiring, for instance, that no more than a fraction Δ of the functions maps distinct inputs into the same output; in that case we say that $\mathcal{H}$ is a Δ-almost universal class of hash functions.

Definition 17. [34]: Let $\mathcal{H}$ be a class of hash functions from $\mathcal{X}$ to $\mathcal{Y}$. We say that $\mathcal{H}$ is Δ-universal if

$$\forall\ x, x' \in \mathcal{X},\ x \neq x', \quad |\{h \in \mathcal{H} : h(x) = h(x')\}| \leq \Delta|\mathcal{H}|.$$

In particular, we say that a class $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is *perfectly* universal, that is, such that $\Delta = 1/|\mathcal{Y}|$, if no pair of distinct inputs is mapped to the same element by more that $1/|\mathcal{Y}|$-th of the functions. Let us consider some examples.

Example 2. *All the linear functions from $\{0,1\}^n$ to $\{0,1\}^\ell$ form a universal hash class [34], $\mathcal{H}_3$. These functions can be described by $n \times \ell$ matrices over GF(2), i.e., by $n\ell$*

bits, and are particularly interesting due to their simple and convenient implementation (traditional tools for efficient matrix multiplication can be exploited straightforwardly).

Example 3. *The class of Toeplitz matrices in $\{0,1\}^{n\times\ell}$ is universal [36]. This class turns out to be particularly convenient, both for the compactness of its description (only $\ell+n-1$ bits are needed to represent each matrix) and for the efficiency of matrix multiplication algorithms that the Toeplitz structure allows [37].*

Example 4. *The last example of universal class of hash functions, $\mathcal{H}_4$, is the one proposed in [34] and revisited in [21]. Let a be an element of GF(2^n) and also interpret $\mathbf{x}$ as an element of GF(2^n). Then, let us consider the function from $\{0,1\}^n$ to $\{0,1\}^\ell$ assigning to an argument $\mathbf{x}$ the first ℓ bits of the element $a\mathbf{x} \in GF(2^n)$. It can be proven that $\mathcal{H}_4$, for $a \in GF(2^n)$, is universal for $1 \leq \ell \leq n$.*

We now briefly report some fundamental results recalled or proven in [24] and in [26]. In particular, we recall the classical version of the *Leftover hashing lemma*, which is the fundamental result that states why universal hash functions can be used for randomness extraction.

Lemma 5. *(Leftover hashing lemma)[26]. Given a n-binary source $\bar{X}$, by averaging over the choice of a function f from a class of universal hash functions $\mathcal{H}$, the distribution of the ℓ-bit output $Ext(\bar{x}, f) = f(\bar{x})$[8] is at most ε_{ext}-far from uniform conditioned on E, where*

$$\varepsilon_{\text{ext}} = \frac{1}{2}2^{\sqrt{\ell - H_2(\bar{X}|E)}}. \tag{2.28}$$

As stated in [26], this "immediately implies that for a fixed joint distribution of $\bar{X}$ and E, there is a fixed function f that extracts almost uniform randomness. In fact, for any $\varepsilon_{\text{ext}} > 0$, there exists a function f which produces

$$\ell = \left\lfloor H_2(\bar{X}|E) - 2\log\left(\frac{1}{\varepsilon_{\text{ext}}}\right) + 2 \right\rfloor \tag{2.29}$$

bits that are ε_{ext}-close to a bit string which is both uniform and independent of E".

Similar results can be proven in the case of Δ-almost universal hash functions, that is, the distance from uniform is bounded by

$$\varepsilon_{\text{ext}} = \frac{1}{2}\sqrt{(2^\ell\Delta - 1) + 2^{\ell - H_2(\bar{X}|E)}}, \tag{2.30}$$

and Eq.(2.28) can be obtained with $\Delta = 2^{-\ell}$. It should be stressed that the relaxation given by setting $\Delta > 2^{-\ell}$ allows for a smaller family of universal hash functions $\mathcal{H}$, that is, less bits are required for specifying a function $f \in \mathcal{H}$ (and the random bits that specify a randomly chosen function f are in fact the *seed* of the extractor).

From a practical point of view, a widely used family of universal hash functions is the one of random matrices. In particular, as mentioned in example 2, the class of Toeplitz matrices takes advantage on one hand of a compact representation and on the other hand

[8] $Ext(\bar{x}, f)$ represents the extractor which corresponds to the random choice of the function $f \in \mathcal{H}$; the seed $\bar{S}$ would in fact be the random bit string which uniquely determines f.

of an efficient software and hardware implementation, based on the Fast Fourier Transform (FFT). A solution based on universal hash functions and tailored for randomness extraction is the one proposed in [38].

2.3.2 Trevisan's extractor

In this section we introduce the notion of Trevisan's extractor. Rather than giving a complete overview, however, we simply sketch the intuition behind this extractor and state some fundamental results. As shown in table 2.2, in fact, Trevisan's extractors allow for the shortest seed length as compared with other solutions (such as universal hashing), and still they output all the extractable randomness of the source; in this sense they are the optimal solution for seeded randomness extraction. For the specific applications we here consider (post-processing of QRNG output and privacy amplification), however, universal hash functions turn out to be a more suitable choice, since they provide both a reasonable seed length and a simple, efficient implementation (for more details, see §2.4).

The main idea behind Trevisan's extractor is that of applying ℓ times a 1-bit extractor to a string of n bits in order to finally get an ℓ-bit string as output. Obviously, a naive application of this paradigm [17], which uses a fresh t-bit seed for each of the ℓ extractions, would require $d = \ell \cdot t$ uniformly random bits to be used as seed, and this is of course not practical. Hence, given an overall seed of $d < \ell \cdot t$ bits, Trevisan's idea is to define minimally overlapping sets $D_1, \ldots, D_\ell \subset \{1, \ldots, d\}$, corresponding to the indexes of the overall seed to be picked up for each of the ℓ single seeds, so that d is poly-logarithmic in the input length; in particular he resorts to the definition of *design* introduced by Nisan and Widgerson [39]. This idea can in fact be improved by relaxing some requirements on the definition of these designs (which are nothing but a partitioning of the set $\{1, \ldots, d\}$ which allows overlappings); the following is the relaxation proposed by Raz, Reingold and Vadhan [40]:

Definition 18. (Weak design). Given $r \geq 1$ and $t \in \mathbb{N}$, the sets $D_1, \ldots, D_\ell \subset \{1, \ldots, d\}$ are said to form a weak (t, r)-design if

1. $\forall i, \quad |D_i| = t$
2. $\forall i, \quad \sum_{j=1}^{i-1} 2^{|D_j \cap D_i|} \leq r(\ell - 1)$.

where the parameter r should be chosen so that $r \approx H_{\min}(\bar{X}|E)/\ell$; in particular, "$1/r$ is essentially the fraction of source min-entropy that is extracted, so ideally r should be as close to 1 as possible" [40]. Furthermore, the authors of [40] prove that by using a weak design a shorter seed can be used in order to extract the same fraction of the source min-entropy, and propose a constructive probabilistic method, based on the method of conditional expectations.

Let us now give a formal definition Trevisan's extractor.

Definition 19. (Trevisan's extractor) For a 1-bit extractor $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}$, which uses a (not necessarily uniform) seed of length t and for a weak (t, r)-design

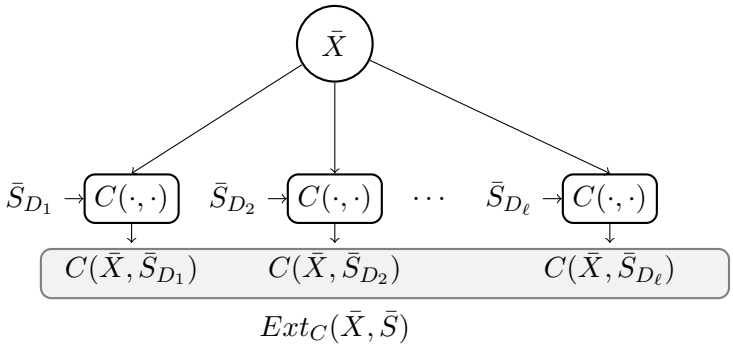


Figura 2.5: Block diagram for a Trevisan's extractor.

$D_1, \ldots, D_\ell \subset \{1, \ldots, d\}$, we define the ℓ-bit extractor $Ext_C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ as

$$Ext_C(\bar{x}, \bar{s}) := C(\bar{x}, \bar{s}_{D_1}) \ldots C(\bar{x}, \bar{s}_{D_\ell})$$

being $\bar{s}_{D_i}$ the subset of the overall seed $\bar{s}$ corresponding to the indexes specified by D_i.

Given this construction and that C is a strong extractor itself, the authors of [9] prove that Ext_C is a randomness extractor resilient to quantum side information[9] both for perfectly uniform seeds and for weakly random seeds (as in the case of extractors from multiple independent sources).

Theorem 2. *[9, Theorem 4.6] (Uniformly random seed) Let $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}$ be a (k,ε)-strong extractor with uniform seed and $D_1, \ldots, D_\ell \in \{1, \ldots, d\}$ a weak (t,r)-design. Then $Ext_C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ is a quantum-proof (k', ε')-strong extractor, with $k' = k + r\ell + \log(1/\varepsilon)$ and $\varepsilon' = 3\ell\sqrt{\varepsilon}$.*

Theorem 3. *[9, Theorem 4.7] (Weakly random seed) Let $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}$ be a (k,ε)-strong extractor with an s-bit seed - i.e., the seed needs at least s bits of min-entropy - and $D_1, \ldots, D_\ell \in \{1, \ldots, d\}$ a weak (t,r)-design. Then $Ext_C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ is a quantum-proof (k'', ε'')-strong extractor, with $k'' = k + r\ell + \log(1/\varepsilon)$ and $\varepsilon'' = 6\ell\sqrt{\varepsilon}$, for any seed with min-entropy $d - (t - s - \log(1/3\sqrt{\varepsilon}))$.*

As we have seen, a fundamental ingredient for a Trevisan's extractors is the underlying one-bit extractor. In order to identify a possible solution for choosing one-bit extractors, let us first give the following definition.

Definition 20. (List-decodable code [30]). A code $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ is said to be an (ε, L)-list-decodable code if every Hamming ball of relative radius $1/2 - \varepsilon$ in $\{0,1\}^{\bar{n}}$ contains at most L codewords.

As the authors of [30] state, both Trevisan [29] and Raz et al. [40] implicitly prove the following result.

[9] it should be noted that, being classical resiliency a sub-case of quantum resiliency, a quantum-resilient extractor is also a classical-resilient extractor.

Proposition 4. *If* $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ *is an* (ε, L)*-list-decodable code, then*

$$Ext : \{0,1\}^n \times \{1,\ldots,\bar{n}\} \to \{0,1\}$$

such that $Ext(\bar{x},\bar{s}) = C(\bar{x})_{\bar{s}}$*, is a* $(\log(L) + \log(1/2\varepsilon), 2\varepsilon)$*-strong extractor.*

To conclude, let us introduce the following simple one-bit extractor:

Definition 21. Given an input $\bar{X} \in \{0,1\}^n$ and a seed $\bar{S} \in \{0,1\}^{k\cdot\log(n)}, \bar{S} = (i_1,\ldots,i_k)$,

$$Ext_k(\bar{X},\bar{S}) = X_{i_1} \oplus \ldots \oplus X_{i_k}$$

is a one-bit extractor.

More specifically, the following theorem can be proven.

Theorem 4. *[9]* Ext_k *is a* $(k, 3\sqrt{\varepsilon})$*-strong extractor for any* $k > H\left(\frac{1}{k}\ln\left(\frac{2}{\varepsilon}\right)\right) n + O\left(\ln\left(\frac{1}{\varepsilon}\right)\right)$.

While thinking of a randomness extractor, one could distinguish different goals, namely: maximize the output length ℓ, minimize the seed length d, optimize computational efficiency. A trade-off between these three targets should necessarily be found, as the optimal solutions for these problems are not likely to coincide. In the case of Trevisan's extractors, their achievement can be pursued by exploiting the two degrees of freedom that its construction provides, that is the choice of the weak design and the choice of the one-bit extractor. We refer to [30] for the description of some possible constructions, namely: near optimal entropy loss extractor, extractor with seed of logarithmic size, locally computable extractor, extractor with weakly random seed. Also, details on a practical implementation of Trevisan's extractor can be found in [41].

2.4 Design and implementation of a randomness extractor for the Quantis device

In this section, we describe a *case study* for the application of randomness extractors to real physical random number generators. In particular, we describe the randomness extractor tailored for the ID Quantique Quantis QRNG proposed in [1], which, more in general, applies to any randomness source which suffers from both bias and correlation in the generated bit sequence. In this scenario, seeded extractors turn out to be the best choice, as they can cope with such kind of sources by providing good extraction and computational efficiency.[10] Among this category of extractors, a solution which fulfils all the requirements described at the beginning of section 2.3, and the additional one of a low-complexity implementation[11] is a particular class of universal$_2$ hash functions, that is, the multiplication by a binary random matrix. In that scenario, we can use Lemma

[10] deterministic extractors would not efficiently remove correlations, whereas multiple independent source extractors would require additional randomness sources which would increase the system complexity and cost.

[11] in the perspective hardware implementation of the extractor, a simple architecture which can be easily deployed in electronics is preferred.

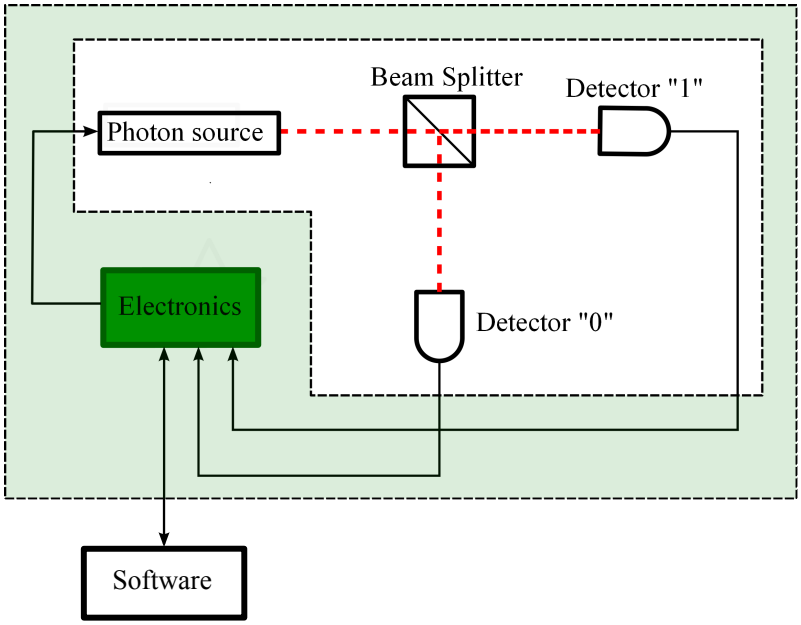


Figura 2.6: Hardware setup of the Quantis device.

5 in order to bound the distance from uniform of the extractor output and the entropy measure to be used is $H_2(\cdot)$.

In the following, we start by recalling the hardware setup of the Quantis device, we introduce some notation, we make some general considerations on the implementation of a randomness extractor base on universal$_2$ hash functions and we then detail the extractor proposed in [1]. Subsequently, we describe its software implementation, which was developed as part of this thesis, and we finally comment the obtained experimental results.

2.4.1 Hardware setup

Let us start by briefly describing the hardware setup of the Quantis. The interested reader can find more details in [42].

As mentioned in the introduction to this chapter, Quantis bases its randomness on the fact that a photon impinging onto an ideal semi-transparent mirror is reflected or transmitted with equal probability [4]. As shown in figure 2.6, a photon source, implemented by a light emitting diode, emits photons that impinge on a 50/50 beam splitter. The transmitted and the reflected photons are then measured by two detectors with single photon resolution, each associated to a bit outcome. This optical subsystem is controlled by a synchronization and acquisition electronic circuit, which then interacts with the Quantis software package. Furthermore, a continuous status monitoring is performed on the device, so that eventual hardware failures are signalled at the output.

Despite the intrinsic randomness of this generator, however, the following non-idealities of the hardware may affect the distance from uniform of its output [43]:

- the beam splitter may not be perfectly unbiased, i.e., it may either transmit or reflect with a slightly higher probability;

- the detectors may have slightly different efficiencies, thus resulting in a small bias of the output sequence;
- the detectors may suffer from after-pulsing. [44], that is, a detection may be more likely to occur after the detector already clicked;
- the detectors may exhibit slightly different dark count rates.

Aware of these effects, ID Quantique tests all of the components for each Quantis device, so that they are minimized. Also, the Quantis periodically switches the bit associated to each detector. Nevertheless, some residual contributions potentially introduce some bias and correlation in the output sequence. The extractor [1, 43] described in the following provides an effective solution for compensating these small imperfections.

2.4.2 Notation

The extraction is performed block-wise via multiplication by a matrix $\mathbf{M}_S \in \{0,1\}^{\ell \times n}$ randomly chosen according to an input seed, S. We denote the random vector representing the i-th n-bit block produced by Quantis which is input to the extractor by $\mathbf{X}^{(i)} = [X_{(i-1)n+1}, \ldots, X_{in}]$, where X_j is the random variable representing the j-th bit produced by Quantis; similarly, we denote the corresponding ℓ-bit block which is output from the extractor by

$$\mathbf{Y}^{(i)} = [Y_{(i-1)\ell+1}, \ldots, Y_{i\ell}] = \mathbf{M}_S \mathbf{X}^{(i)}, \tag{2.31}$$

where operations are performed modulo 2. Also, we denote by $X_j^{(i)}$ and by $Y_j^{(i)}$ the random variables representing the j-th bit of the i-th input block and the j-th bit of the i-th output block, respectively, i.e., $\mathbf{X}^{(i)} = [X_1^{(i)}, \ldots, X_n^{(i)}]$ and $\mathbf{Y}^{(i)} = [Y_1^{(i)}, \ldots, Y_\ell^{(i)}]$. Finally, with lower case letters we denote the specific realization of a given random variable; for instance, the realization of the i-th input block will be denoted by $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_n^{(i)}]$.

2.4.3 General considerations

Before describing the specific extractor for the Quantis, we make some preliminary remarks which apply, in general, to any randomness extractor for the post-processing of the output to a PhyRNG. Despite the straightforward efficient implementation that a solution such as the one proposed in the following ensures, in real world applications one should also take into account non-idealities of the system. Let us consider some crucial aspects. Their joint contribution to the overall performance of the extractor are then assessed in proposition 6.

- **Entropy estimation.** The impact of the reliability of entropy estimation on the raw data should be considered when evaluating the overall statistical distance from uniform of the output. In particular, the entropy is estimated starting from the empirical distribution of the source, which is derived from a finite number of samples. If we denote by $\tilde{H}_2^b(\bar{X})$ the estimated collision entropy per bit of the raw source and by $H_2^b(\bar{X})$ the actual one, we bound the probability of an estimation error as follows

$$P\left[\tilde{H}_2^b(\bar{X}) > H_2^b(\bar{X})\right] \leq \varepsilon_{\text{est}}. \tag{2.32}$$

In [1] a method for estimating $H_2^b(\bar{X})$ with bounded error is provided and we describe it in the following. Given a sequence of random input bits $\{X_i\}$, let us denote by $X_{[1,m]}$ the random vector representing the sequence of the first m bits, that is, $X_{[1,m]} = [X_1, \ldots, X_m]$, and, similarly, let $x_{[1,m]}$ be its realization. Then, we define the entropy gain obtained by adding 1 bit to an $(m-1)$-bit sequence as:

$$H_2^m(\bar{X}) \triangleq H_2^{\varepsilon_{\text{est}}}(X_{[1,m]}) - H_2^{\varepsilon_{\text{est}}}(X_{[1,m-1]}), \tag{2.33}$$

where the smoothing parameter ε_{est} is the tolerated estimation error.[12] Then, $\tilde{H}_2^b(\bar{X})$ should be taken as the asymptotic limit of (2.33), that is,

$$\tilde{H}_2^b(X) = \lim_{m\to\infty} H_2^m(X). \tag{2.34}$$

The collision entropy $H_2(X_{[1,m]})$ can be estimated by the normalized histogram $h^{(m)}(\cdot)$, that is, the empirical probability distribution of m-bit strings. More formally,

$$\tilde{H}_2(X_{[1,m]}) = -\log_2 \sum_{x_{[1,m]}\in\{0,1\}^m} [h^{(m)}(x_{[1,m]})]^2 \ . \tag{2.35}$$

As observed in [1], higher values of m require exponentially more memory and sampling time; therefore, the possibility of increasing m is also limited by its computational feasibility. Fortunately, as for the Quantis device, the authors of [1] observed a very fast convergence to the asymptotic value, and choosing $m \leq 16$ is sufficient. Then, the histograms for smaller values of m can be derived from $h^{(m)}(\cdot)$ by partial sums, that is,

$$h^{(m-1)}(x_{[1,m-1]}) = \sum_{x_m=0}^{1} h^{(m)}(x_{[1,m]}). \tag{2.36}$$

Intuitively, this procedure should be repeated several times for evaluating the estimation error. In particular, the estimates for the mean value, $\bar{H}_2^b(\bar{X})$, and for the statistical error, $\Delta H_2^b(\bar{X})$, have to be derived by standard statistical methods. Then, given the tolerance on the estimation error ε_{est} and a sufficiently high number of samples (so that the Gaussian approximation can be invoked by the central limit theorem), we get

$$\tilde{H}_2^b(\bar{X}) = \bar{H}_2^b(\bar{X}) - \alpha\Delta H_2^b(\bar{X}), \tag{2.37}$$

where the value α is chosen so that $\text{erfc}(\alpha) < \varepsilon_{\text{est}}$, being erfc the complementary error function.

[12] assuming an error probability ε_{est} in the estimate for the collision entropy corresponds, in fact, to estimating the ε_{est}-smooth collision entropy.

Finally, the estimate for the collision entropy per bit to be considered when applying the left-over hash lemma has to be chosen as

$$\tilde{H}_2^b(X) = \min_m H_2^m(X). \tag{2.38}$$

- **Non-uniformity of the seed.** Also the d-bit seed to the extractor, $\bar{S}$, may not be perfectly uniform (this is actually the case in any realistic application), i.e.,

$$\delta\big(P_{\bar{S}}, U_d\big) \leq \varepsilon_{\text{seed}}. \tag{2.39}$$

As shown in §2.4.4, the value of $\varepsilon_{\text{seed}}$ can be made arbitrarily small, but, still, it has to be considered while assessing the overall distance from uniform of the output.

- **Block-wise extraction effect.** Extraction is performed block-wise, as the raw output bit stream is partitioned in n-bit input blocks that produce ℓ-bit output blocks, with $\ell < n$. Besides the trade-off between computational overhead and extraction efficiency, in general also the correlation between subsequent blocks has to be taken into account. This means that the conditioned entropy should be used while applying Lemma 5, that is, the distance from uniform of the extractor output, ε_{ext}, is such that

$$\varepsilon_{\text{ext}} = \frac{1}{2} 2^{\sqrt{\ell - H_2^c(\mathbf{X})}}, \tag{2.40}$$

being $H_2^c(\bar{\mathbf{X}})$ the average collision entropy per block conditioned on previous blocks. For the sake of simplicity, we here assume that this quantity is stationary and, without loss of generality, in what follows we temporarily do not consider the statistical error ε_{est} in the collision entropy estimation. In that setting, the following proposition (which is a re-adaptation of the result shown in [45, Appendix A]) is proven for the extractor described in §2.4.4.

Proposition 5. *Let $\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(N)}\}$ be a set of n-bit blocks which are input to the extractor and let $\{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(N)}\}$ be the corresponding ℓ-bit output blocks for a random S, independent of the input blocks. If the input to the extractor satisfies a Markov condition of length m, that is,*

$$P[\mathbf{X}^{(i)} | \mathbf{X}^{(i-1)}, \ldots, \mathbf{X}^{(1)}] = P[\mathbf{X}^{(i)} | X_{(i-1)n}, \ldots, X_{(i-1)n-m}] \tag{2.41}$$

and the conditional collision entropy is stationary for all blocks and equal to $H_2^c(\bar{\mathbf{X}})$, then the distance from uniform per block is bounded as

$$\delta\big(P_{\mathbf{Y}^{(i)}S}, U_\ell \times U_d\big) \leq \varepsilon_{\text{seed}} + \varepsilon_{\text{ext}}, \tag{2.42}$$

where $\varepsilon_{\text{seed}} = \delta\big(P_S, U_d\big)$ and $\varepsilon_{\text{ext}} = \frac{1}{2} 2^{\sqrt{\ell - H_2^c(\mathbf{X})}}$.

Dimostrazione. Let us first assume that the choice of the seed is performed uniformly at random, namely, $\delta\big(P_S, U_d\big) = 0$, and consider the effects of correlations between blocks. The first block which is output from the extractor does not depend on previous blocks, that is

$$\delta\big(P_{\mathbf{Y}^{(1)}}, U_\ell\big) \leq \varepsilon_1, \tag{2.43}$$

being $\varepsilon_1 \triangleq \frac{1}{2} 2^{\sqrt{\ell - H_2(\mathbf{X}^{(1)})}}$. The second block, on the other hand, may depend on the previous one, thus yielding

$$\delta\big(P_{\mathbf{Y}^{(2)}\mathbf{X}^{(1)}}, U_\ell \times P_{\mathbf{X}^{(1)}}\big) \leq \varepsilon_2, \tag{2.44}$$

being $\varepsilon_2 \triangleq \frac{1}{2} 2^{\sqrt{\ell - H_2(\mathbf{X}^{(2)}|\mathbf{X}^{(1)})}}$. Therefore, the distance from uniform of the joint probability distribution of the first two output blocks can be bounded as follows:

$$\begin{aligned}
\delta\big(P_{\mathbf{Y}^{(2)}\mathbf{Y}^{(1)}}, U_\ell \times U_\ell\big) &\leq \delta\big(P_{\mathbf{Y}^{(2)}\mathbf{Y}^{(1)}}, U_\ell \times P_{\mathbf{Y}^{(1)}}\big) && (2.45)\\
&+ \delta\big(U_\ell \times P_{\mathbf{Y}^{(1)}}, U_\ell \times U_\ell\big) \\
&\leq \delta\big(P_{\mathbf{Y}^{(2)}\mathbf{X}^{(1)}}, U_\ell \times P_{\mathbf{X}^{(1)}}\big) + \varepsilon_1 && (2.46)\\
&\leq \varepsilon_2 + \varepsilon_1, && (2.47)
\end{aligned}$$

where in (2.45) we used the triangle inequality, in (2.46) the data processing inequality and (2.43), and in (2.47) we used (2.44). Hence, by extending the previous argument to N blocks, we get

$$\delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}}, U_\ell \times \ldots \times U_\ell\big) \leq \sum_{i=1}^{N} \varepsilon_i \triangleq \varepsilon(N). \tag{2.48}$$

Given (by hypothesis) that the input to the extractor satisfies a Markov condition of length m, i.e.,

$$H_2(\mathbf{X}^{(i)}|\mathbf{X}^{(i-1)}, \ldots, \mathbf{X}^{(1)}) = H_2(\mathbf{X}^{(i)}|X_{(i-1)n}, \ldots, X_{(i-1)n-m}) \tag{2.49}$$

and that the conditional collision entropy is stationary for all blocks and equal to $H_2^c(\mathbf{X})$, i.e.,

$$\begin{aligned}
H_2^c(\mathbf{X}) &\triangleq H_2(\mathbf{X}^{(i)}|X_{(i-1)n}, \ldots, X_{(i-1)n-m}) && (2.50)\\
&= H_2(\mathbf{X}^{(j)}|X_{(j-1)n}, \ldots, X_{(j-1)n-m}), \quad \forall i, j
\end{aligned}$$

we get

$$\varepsilon(N) = \varepsilon_1 + (N-1)\varepsilon_{\text{ext}}, \tag{2.51}$$

where $\varepsilon_{\text{ext}} \triangleq \frac{1}{2} 2^{\sqrt{\ell - H_2^c(\mathbf{X})}}$. Now, since $H_2^c(\mathbf{X}) \leq n H_2^b(X)$ (entropy cannot increase by conditioning), we upper bound ε_1 by ε_{ext} and we finally obtain

$$\delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}}, U_\ell \times \ldots \times U_\ell\big) \leq N\varepsilon_{\text{ext}}. \tag{2.52}$$

If we now take into account also the seed, S, we get

$$\begin{aligned}\delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}S}, U_\ell \times \ldots \times U_\ell \times U_d\big) \;&\leq\; \delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}S}, P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}} \times U_d\big) \\ &+\; \delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}} \times U_d, U_\ell \times \ldots \times U_\ell \times U_d\big)\end{aligned} \tag{2.53}$$

Now, given (by hypothesis) that the input blocks $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(N)}$ are independent of S, we get

$$\delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}S}, P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}} \times U_d\big) = \delta\big(P_S, U_d\big) = \varepsilon_{\text{seed}}. \tag{2.54}$$

Eventually, by plugging (2.52) and (2.54) into (2.53), we get the final bound on the overall distance from uniform of the concatenated output sequence:

$$\delta\big(P_{\mathbf{Y}^{(1)}\ldots\mathbf{Y}^{(N)}S}, U_\ell \times \ldots \times U_\ell \times P_S\big) \leq \varepsilon_{\text{seed}} + N\varepsilon_{\text{ext}}. \tag{2.55}$$

Therefore, the distance *per block*, $\delta\big(P_{\mathbf{Y}^{(i)}S}, U_\ell \times U_d\big)$, is bounded by $\varepsilon_{\text{seed}} + \varepsilon_{\text{ext}}$.

□

Proposition 5 states that the distance from uniform per block depends neither on the number of concatenated blocks N nor on the fact that the same seed is reused several times (provided that the seed is independent of the input to the extractor), but rather relies on the quality of the seed as well as on the quality of the raw input bits, measured here in terms of conditional collision entropy. Therefore, the impact of block-wise hashing is essentially captured by the fact that $H_2^c(\mathbf{X}) \leq nH_2^b(X)$. We note, however, that an appropriate choice of n can easily ensure the conditional entropy per bit is arbitrarily close to $H_2^b(X)$.

As a conclusion to this section, we summarize the overall distance from uniform per block due to the three above-mentioned contributions, ε_{est}, $\varepsilon_{\text{seed}}$, ε_{ext}. More formally, the following result is proven.

Proposition 6. *Given that* $P\left[\tilde{H}_2^c(\bar{X}) > H_2^c(\mathbf{X})\right] \leq \varepsilon_{\text{est}}$*, the overall distance from uniform of each block which is output to the extractor can be bounded as*

$$\delta(P_{\mathbf{Y}^{(i)}\bar{S}}, U_\ell \times U_d) \leq \varepsilon_{\text{est}} + \varepsilon_{\text{ext}} + \varepsilon_{\text{seed}}. \tag{2.56}$$

Dimostrazione. Let Q be the event $\{\tilde{H}_2^c(\bar{X}) > H_2^c(\bar{\mathbf{X}})\}$ and $\bar{Q}$ its complementary, where $P[Q] \leq \varepsilon_{\text{est}}$.

$$\delta(P_{\mathbf{Y}^{(i)}S}, U_\ell \times U_d)$$
$$\leq \frac{1}{2}\sum_{(\bar{y},\bar{s})}\left\{P[Q]\left|p_{\mathbf{Y}^{(i)}S|Q}(\bar{y},\bar{s}) - \frac{1}{2^n}\frac{1}{2^\ell}\right| + P[\bar{Q}]\left|p_{\mathbf{Y}^{(i)}S|\bar{Q}}(\bar{y},\bar{s}) - \frac{1}{2^n}\frac{1}{2^d}\right|\right\} \quad (2.57)$$
$$\leq P[Q] + P[\bar{Q}]\,(\varepsilon_{\text{ext}} + \varepsilon_{\text{seed}}) \quad (2.58)$$
$$\leq \varepsilon_{\text{est}} + (1-\varepsilon_{\text{est}})(\varepsilon_{\text{ext}} + \varepsilon_{\text{seed}}) \quad (2.59)$$
$$\leq \varepsilon_{\text{est}} + \varepsilon_{\text{ext}} + \varepsilon_{\text{seed}}. \quad (2.60)$$

Eq.(2.57 follows from the definition of statistical distance and from the total probability theorem, Eq.(2.58) from the fact that the statistical distance is always smaller than 1 and from eq.(2.42) (conditioning on $\bar{Q}$ is actually the hypothesis under which eq.(2.42) has been derived), Eq.(2.59) from the fact that $P[Q] \leq \varepsilon_{\text{est}}$. □

Summarizing, the contributions of the entropy estimation error, of the distance from uniform of the seed and of the block-wise extraction effects have to be carefully taken into account while assessing the uniformity of the extractor output, and should reasonably be of the same order of magnitude.

2.4.4 Extractor design

The randomness extractor we describe is the one proposed in [1], which, based on known literature results, is tailored for the Quantis device and oriented to its software implementation.

Extractor parameters. In the framework of universal$_2$ hash functions, the distance from uniform of the extractor output, ε_{ext}, can be bounded via the leftover hashing lemma (lemma 5), so that

$$\varepsilon_{\text{ext}} = \frac{1}{2}2^{\sqrt{\ell - nH_2^b(\bar{X})}}, \quad (2.61)$$

where $H_2^b(\bar{X})$ is the collision entropy per bit of the source $\bar{X}$. Given ε_{ext} and n, we then define the extraction efficiency as

$$\eta_{\text{ext}} \triangleq \frac{\ell}{n} = H_2^b(\bar{X}) - \frac{2\log_2(1/\varepsilon_{\text{ext}})}{n}. \quad (2.62)$$

Hence, as stated in [1], the choice of the extractor parameters, i.e., n and ℓ, undergoes two conflicting requirements: on one hand, η_{ext} can be raised by increasing the value of n; on the other hand, the computational complexity of the matrix-vector multiplication (2.31) is $O(n\ell)$, thus requiring $O(n)$ operations per output bit.

In [1], the authors propose $(n_1, \ell_1) = (1024, 768)$ and $(n_2, \ell_2) = (2048, 1792)$ as parameters which provide a good trade-off between extraction efficiency and computational overhead. These parameters ensure a distance from uniform of the output $\varepsilon_{\text{ext}} < 2^{-100}$ as long as $H_2^b(\bar{X}) > 0.946$ for (n_1, ℓ_1) and $H_2^b(\bar{X}) > 0.973$ for (n_2, ℓ_2). Empirical tests showed that these conditions are always matched by Quantis, thanks to the continuous

monitoring of the output to the device (see implementation details [42]). Finally, as for the Quantis extractor, the choice of n and ℓ has been restricted to multiples of 64 for performance reasons (see section 2.4.5).

Extractor matrix generation. The distance from uniform $\varepsilon_{\text{seed}}$ of the extractor matrix $\mathbf{M}_S$ is a fundamental parameter for the described construction, as shown in proposition 5. The creation of $\mathbf{M}_S$ is therefore of crucial importance. Let us also note that, provided that $\mathbf{M}_S$ is independent of the extractor input, it can be reused several times without affecting the overall distance from uniform of the extractor (see, again, proposition 5). For the parameters $(n_1, \ell_1) = (1024, 768)$ and $(n_2, \ell_2) = (2048, 1792)$, a seed of 768 Kbits and 3584 Kbits is required, respectively.

An effective method for producing $\mathbf{M}_S$ is that of exploiting the procedure and the result of proposition 3, that is, XORing the output of t independent, weakly-random sources. This method is not efficient, as it requires t bits from t distinct sources for producing 1 output bit, but, since the seed has to be determined only once and can be used for any Quantis device, this does not represent a problem. The value of $\varepsilon_{\text{seed}}$ decreases exponentially with t, which, therefore, should be chosen sufficiently large to satisfy the required overall distance from uniform of the extractor output.

In addition, in order to minimize the distance from uniform of the random sequences produced by each source, the authors of [1] suggest to take some further precautions: first, to use only bits separated by an interval which is much longer than the auto-correlation time, in order to get rid of correlations between subsequent bits generated by the same source (for a Quantis source, a distance of 100 to 1000 bits turns out to be an appropriate choice, since only short-time correlation are measured for this device); second, to use the Von Neumann de-biasing algorithm [46] in order to compensate for a bias in the probability of having either a 0 or 1. From now on, we define as *elementary matrix* the output of these sampling and de-biasing steps, while we denote as *extractor matrix* the random sequence resulting from XORing t elementary matrices obtained from independent sources.

2.4.5 Software architecture

The randomness extraction algorithm detailed in section 2.4.4 has been finally implemented and included into the Quantis software library. In particular, a new library, called `libQuantisExtensions`, has been developed on the top of the existing `libQuantis` (see figure 2.7); while designing its architecture, the following principles were observed:

- *seamless integration with* `libQuantis`: the newly introduced functions use the functions of `libQuantis` for accessing the device and reading raw random bits sequences and are consistent with its syntax.

- *backward compatibility*: existing applications which use the `libQuantis` do not have to be modified, unless the user wants to take advantage of the randomness extraction.

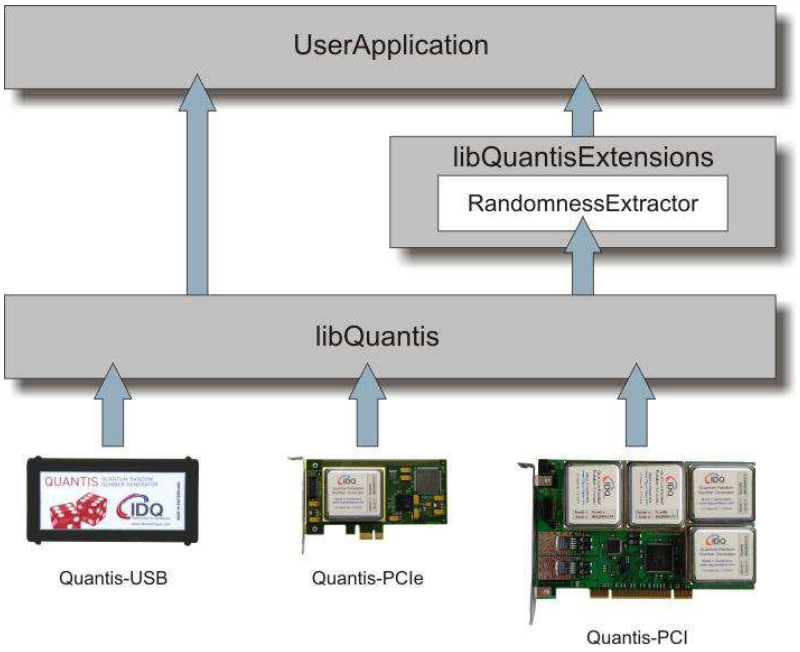


Figura 2.7: Quantis software libraries architecture [43].

- *computational efficiency*: in order to maximize the generation bit-rate of extracted bit sequences, the minimization of the computational overhead of the randomness extraction algorithm has been a leading design priority.

- *modular structure*: the structure of `libQuantisExtensions` is such that other extraction algorithms could be easily integrated into the software package.

The library has two main set of functions, one devoted to the extraction procedure and the other one to the generation of the extraction matrix. The first set of functions allows either to read extracted random data directly from Quantis, or to process an input file; the user can specify the extraction parameters n and ℓ depending on the collision entropy per bit of the raw input sequence (see §2.4.3 for a discussion on the entropy estimation) and on the required distance from uniform of the output, according to equation (2.61). The second set of functions allows the creation of the extractor matrix, according to the procedure described in §2.4.4. More specifically a function is available for creating an elementary matrix out of a single Quantis source, while another function performs the XOR of the t input elementary matrices, possibly obtained by means of multiple Quantis devices. Furthermore, the user who wants to apply the elementary matrix creation procedure to a source different from Quantis, may use the down-sampling and the Von Neumann de-biasing functions which are made available in the software package.

As the creation of an extractor matrix requires several independent devices, a default extractor matrix has been hard-coded in a file delivered within the Quantis software package. The embedded extractor matrix has been generated by means of 10 independent Quantis devices, whose output has been down-sampled so that 1 bit out of 100 has been used for the elementary matrix generation; since the measured bias is in the order of 10^{-4}, according to lemma 3 we get $\varepsilon_{\text{seed}} \simeq 5 \cdot 10^{-40}$, which is generously compliant with the

(n, ℓ)	Bit-rate [Mbit/s]		
	TH	PC-1	PC-2
$(1024, 768)$	3.00	2.80	2.90
$(2048, 1792)$	3.50	3.00	3.30
$(4096, 3840)$	3.75	3.20	3.40

Tabella 2.3: Theoretical vs. experimental read-and-extract bit-rates with Quantis USB for different extraction parameters.

required overall distance from uniform. If the user wants to use his own extractor matrix, however, the default one can easily be replaced.

For further details, the interested reader may refer directly to the source code, available at `http://www.idquantique.com/random-number-generators/resources.html`.

2.4.6 Experimental results

We now briefly analyze the performance of the extractor, both in terms of efficiency and in terms of quality of the produced randomness.

In order to evaluate the efficiency of the extractor, we first recall that a single Quantis module (such as the one installed on a Quantis USB) produces raw random sequences at 4 Mbit/s; furthermore, we here assume that the module is operating in the so-called *standard mode*, thus meaning that the bit values associated with each detector are periodically switched. Table 2.4.6 shows, for a single Quantis module, the highest theoretical read-and-extract bit-rate (column TH) that is achievable for three different pairs of extraction parameters (n, ℓ), together with the experimental results obtained on two different testing platform. The first one, denoted in table 2.4.6 as PC-1, is a personal computer running Ubuntu 12.04 64-bit on an Intel Core 2 Duo E7400 CPU (2.8 GHz) with 4 GB of RAM, while the second one, denoted in table 2.4.6 as PC-2, is a personal computer running Ubuntu 12.04 64-bit on an Intel i7-2600 CPU (3.4GHz) with 4 GB of RAM. The experimental bit-rate is affected both by the extractor efficiency, $\eta_{\text{ext}} = n/\ell$, and by the computational overhead of the extraction algorithm, which depends on (n, ℓ) as well as on the available computational resources. We obtained the highest bit-rate on PC-2 (which is in fact the most performing platform) while using $(n, \ell) = (4096, 3840)$; more specifically, an average bit-rate of 3.4 Mbit/s has been measured, corresponding to a read-and-extract efficiency $\eta_{\text{tot}} = 0.85$ (i.e., the read-and-extract bit-rate is 85% of the raw data reading bit-rate), whereas the theoretical extractor efficiency is $\eta_{\text{ext}} = 3840/4096 = 0.9375\%$. The gap between η_{tot} and η_{ext} is due to the computational overhead, and might be reduced by deploying more computational resources and by taking advantage of parallel computing (e.g., by using one thread for reading raw data from Quantis and multiple threads to concurrently process distinct blocks). It should also be noted that this gap changes according to the extractor parameters (n, ℓ); for instance, we observe that for $(n, \ell) = (1024, 768)$ the computational overhead has an almost negligible impact on the read-and-extract efficiency, with the overall efficiency achieving $\eta_{\text{tot}} = 0.725$, which only slightly deviates from the theoretical optimum, $\eta_{\text{ext}} = 0.75$.

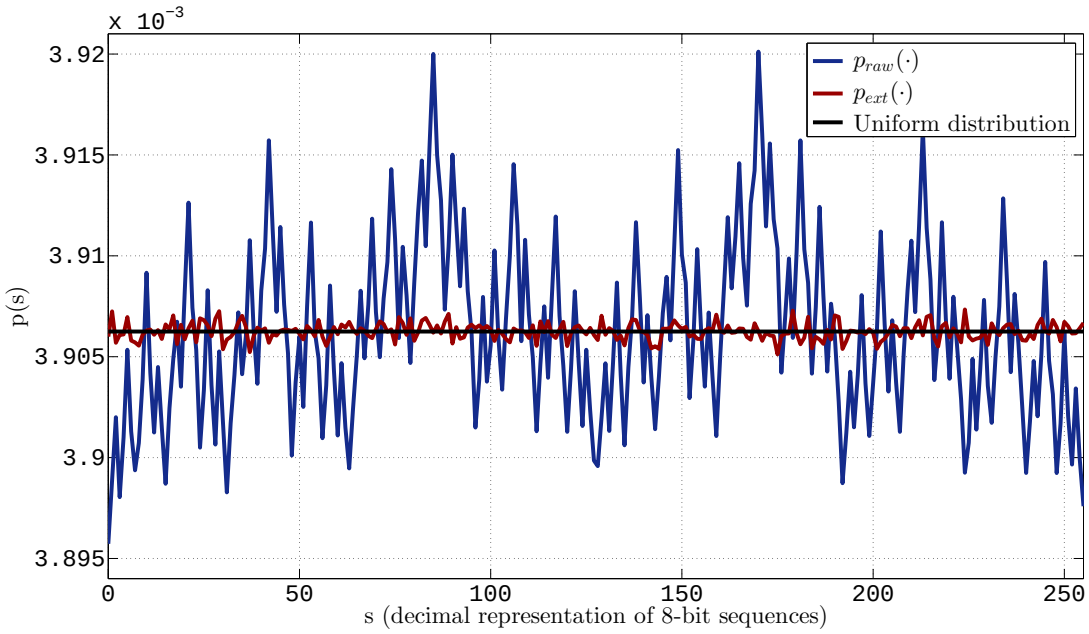


Figura 2.8: Empirical distribution of 8-bit sequences of 30 gigabytes of raw data (blue solid line) and extracted data (red solid line), respectively, and corresponding mean value (black solid line). The source is a Quantis USB (S/N 060010A410) and the used extractor parameters are $(n, k) = (1024, 768)$.

Finally, in order to provide a visual example of the improvement of the quality of randomness ensured by the extractor, in figure 2.8 we compare the empirical probability distribution of the raw random data, $p_{\text{raw}}(\cdot)$ (blue line), with the one of the extracted random data (red line), $p_{\text{ext}}(\cdot)$, produced by Quantis. In particular, this distribution is computed for 8-bit sequences on a data-set of 30 gigabytes.[13] We see that $p_{\text{ext}}(\cdot)$ has a significantly lower variance as compared with $p_{\text{raw}}(\cdot)$; also, the peaks of $p_{\text{raw}}(\cdot)$ are effectively smoothed by the extraction algorithm, and $p_{\text{ext}}(\cdot)$ slightly oscillates around the perfectly uniform distribution (black line).

[13]It should be noted that the deviation from uniform of the Quantis output is noticeable only when considering a sample space of at least 1 Gigabyte, as otherwise, the observed deviations in the bytes distribution can be ascribed to the statistical fluctuations.

Capitolo 3

Quantum Key Distribution

The problem of generating a shared secret key between two distant parties, known as secret key agreement, is of utmost importance to most security applications. Several protocols exist for addressing this problem, but, to date, most practical solutions base their security on computational assumptions [47]. According to this approach, a key is assumed to be secret if the adversary is not able to guess it in a feasible amount of time; typically, this infeasibility result is established on the assumed, yet unproven, hardness of a mathematical problem. This is the case, for instance, of the Diffie-Hellman scheme [48], a popular key agreement protocol whose security relies on the non-polynomial complexity of the discrete logarithm problem. Such infeasibility assumptions, however, are not proven, and a mathematical or technological breakthrough may, in the near future, open the way to an efficient solution for such problems.

It is then natural to look for a more general definition of security, that is, one which does not rely on any assumption on the attacker computing power. This is the rationale behind the notion of *information-theoretic security*, a new approach to cryptography pioneered by Claude Shannon [49]. More precisely, in the context of secret key agreement, Shannon proposed to establish the security of a cryptographic key based on the maximum amount of information that an adversary has on the key itself, regardless of her attack strategy or of her computational power.

While looking towards information-theoretic security, the laws of quantum physics provide significant advantages over purely classical systems, and allow to effectively bound the information that the adversary may have on a sequence which is output to the key distillation procedure. In a quantum channel, in fact, "the leakage of information is quantitatively related to the degradation of the communication" [50].

In this chapter, we start by reviewing the information-theoretic secret key agreement scheme that was proposed by Maurer (§3.1), and we extend it to the quantum scenario. We classify quantum key distribution protocols according to the distribution technique and to the coding scheme (§3.2), and we then recall some fundamental results for security (§3.3): the leveraged quantum laws, the considered attack models, and different secrecy measures. After a description of some explicit QKD protocols (§3.4), i.e., the BB84, the efficient BB84 and the B92 protocols, we detail the information reconciliation phase (§3.5), by first providing a classification of different approaches and by then focusing on the the

analysis of some practical solutions; the described results are used for the experiments described in chapter 4. Finally, we describe two results obtained in the framework of privacy amplification against selective individual attacks and presented in [C1] and in [J1].

3.1 Information-theoretic secret key agreement: system model

A novel approach to information-theoretic secret key agreement has been proposed by Ueli Maurer in his seminal paper [35]. In this work, the author provides a practical scheme for generating a shared secret key in an adversarial scenario with noisy channels, and proves that it is secure against attackers with unlimited computing power, that is, according to an information-theoretic approach. This scheme, which, in fact, does not rely on the laws of quantum mechanics, finds probably its most notable application in quantum key distribution. In this section, we provide an overview on the general scheme, whereas in the following sections we detail its building blocks as applied to quantum key distribution.

Let us start by describing the proposed system model, depicted in figure 3.1.

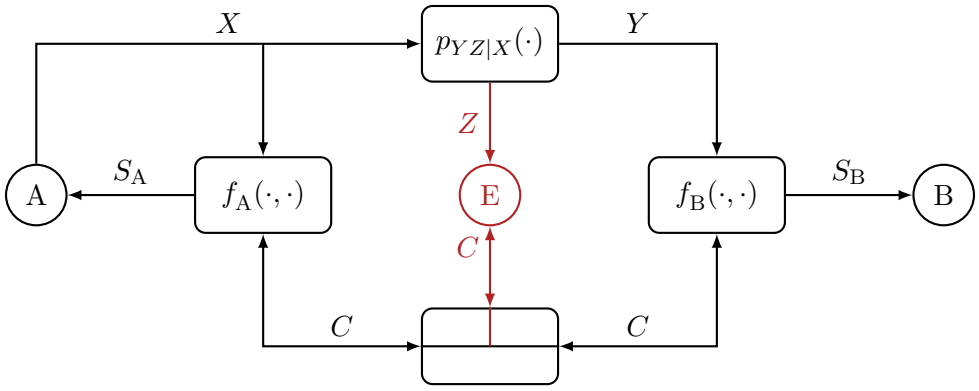


Figura 3.1: Information-theoretic secret key agreement system model.

Two legitimate parties, Alice and Bob, aim at creating a pair of shared secret keys, represented by the random variables S_A and S_B, so that $S_A = S_B = S$ and a potential eavesdropper, Eve, has negligible information on them.

In order to do that, Alice transmits a random sequence, X, over an insecure, noisy channel with two outputs: the first one, Y, is the noisier version of X received by Bob, whereas the second one, Z, represents the eavesdropped sequence. X and Y are called *raw keys*, in that they are obtained by a raw physical transmission with no post-processing.

In Maurer's scheme, the first step is in fact the transmission over an insecure channel, with the aim of sharing some correlated information between Alice and Bob, but, unfortunately, introducing some correlation also with the eavesdropper. In principle, Eve may even share more correlation than Bob with Alice. Formally, this can be written as follows:

$$\mathbb{I}(X;Z) > \mathbb{I}(X;Y), \tag{3.1}$$

where $\mathbb{I}(\cdot)$ is a generic information measure which depends on the assumed attack scenario (explicit definitions are presented in §3.3.3). However, even in this disadvantageous condition, it can be shown [35] that Alice and Bob can extract a secret key pair by jointly post-processing their data. In particular, they need to share some further information, so that the following conditions may be satisfied:

$$(\textit{correctness}) \qquad P[S_{\mathrm{A}} \neq S_{\mathrm{B}}] < \varepsilon_{\mathrm{cor}} \tag{3.2}$$

$$(\textit{secrecy}) \qquad \mathbb{I}(S_{\mathrm{A}}, S_{\mathrm{B}}; Z, C) < \varepsilon_{\mathrm{sec}} \tag{3.3}$$

where $\varepsilon_{\mathrm{cor}}$ and $\varepsilon_{\mathrm{sec}}$ are required to be negligibly small. Then, Alice and Bob communicate over a public authenticated channel, and the exchanged information is summarized by the random variable $C = [C_{\mathrm{A}}, C_{\mathrm{B}}]$, being C_{A} and C_{B} the random variables representing the messages sent by Alice and Bob, respectively. Eve has complete access to this channel, but cannot tamper with the sent messages, or forge new messages and pretend to be either Alice or Bob. Requiring public channel authentication in a secret key agreement scheme seems contradictory, as for creating a secret key pair, $(S_{\mathrm{A}}, S_{\mathrm{B}})$, Alice and Bob first need to share some further key material. However, this pre-shared secret is needed only at the first protocol run, as subsequent iterations can use part of the fresh-new generated key. Hence, from a practical point of view, this issue is easily circumvented by preliminarily loading a short secret in Alice and Bob's devices before deploying the agreement scheme . The correlation shared thanks to the transmission over the insecure channel, together with the information exchanged over the public channel, allows Alice and Bob, by means of a set of post-processing functions summarized by $f_{\mathrm{A}}(\cdot)$ at Alice's side and by $f_{\mathrm{B}}(\cdot)$ at Bob's side, to extract the secret key pair, $(S_{\mathrm{A}}, S_{\mathrm{B}})$.

Let us now take an overview of the steps of the practical key distillation scheme proposed by Maurer, as depicted in figure 3.2; further details are provided in the following sections.

As already mentioned, the first step is the physical transmission over the insecure channel, which allows Bob to establish some correlated information with Alice. Intuitively, since the adversary may share even more correlation with Alice as compared with Bob (see Eq.(3.1)), a post-selection of the transmitted and received sequence, respectively, is required. This phase, whose output are the *sifted keys*, X_{S} at Alice's side and Y_{S} at Bob's side, is called *advantage distillation* and aims at fulfilling the following condition:

$$\mathbb{I}(X_{\mathrm{S}}; Y_{\mathrm{S}}) > \mathbb{I}(X_{\mathrm{S}}; Z, C'), \tag{3.4}$$

that is, at getting an advantage over the eavesdropper. This task is accomplished by means of a pair of post-selection functions, $(f'_{\mathrm{A}}(\cdot,\cdot), f'_{\mathrm{B}}(\cdot,\cdot))$, which leverage the public communication for jointly choosing a subset of the transmitted and received bits at Alice's and Bob's side, respectively. In quantum key distribution, this phase is referred to as *sifting*, as it will be detailed in section 3.2.

The next step, called *information reconciliation*, has the objective of correcting the errors that the channel may have introduced in the transmission. Again, error correction

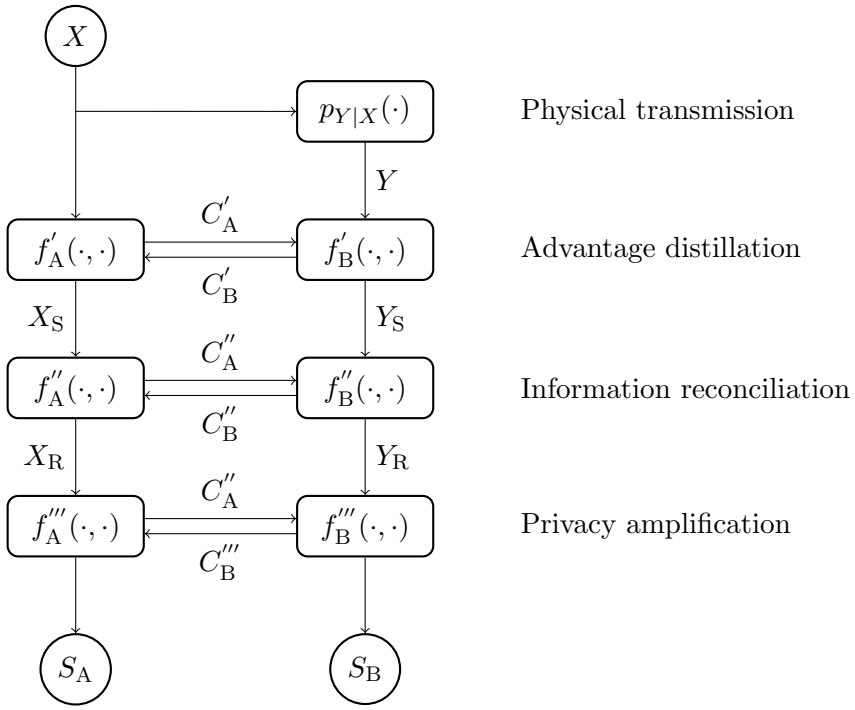


Figura 3.2: Information-theoretic secret key agreement procedure.

is jointly performed by Alice and Bob by exploiting the public channel. In particular, the information reconciliation functions are denoted by $f''_{\mathrm{A}}(\cdot,\cdot)$ and $f''_{\mathrm{B}}(\cdot,\cdot)$, they take as input the sifted keys and the exchanged public communication, and they output the reconciled sequences pair, $(X_{\mathrm{R}}, Y_{\mathrm{R}})$, so that

$$P[X_{\mathrm{R}} \neq Y_{\mathrm{R}}] < \varepsilon_{\mathrm{cor}}, \tag{3.5}$$

with $\varepsilon_{\mathrm{cor}}$ arbitrarily small. Condition (3.5) implies, in fact, the *correctness* constraint (3.3), since the final key is essentially derived by randomly choosing a compression function which is then applied to both reconciled keys (see section 3.6). We stress that there exist different approaches to information reconciliation, which are going to be detailed in section 3.5.

As a final step, the so-called *privacy amplification* takes place. Its objective is to produce a key pair, $(S_{\mathrm{A}}, S_{\mathrm{B}})$ such that the attacker has negligible information on it, that is, such that the secrecy constraint (3.3) is fulfilled. This task is accomplished by means of the privacy amplification functions $f'''_{\mathrm{A}}(\cdot,\cdot)$ and $f'''_{\mathrm{B}}(\cdot,\cdot)$, which take as input both the reconciled strings and the exchanged public communication, and output the final key. In the hypothesis that the two final keys are identical, we denote them by $S = S_{\mathrm{A}} = S_{\mathrm{B}}$.

An intuitive plot, showing how the crucial quantities involved in the different phases of the described secret key agreement scheme change, is shown in figure 3.3. In the following sections, we are going to describe in more detail these three phases as applied to our focus scenario, that is, quantum key distribution.

The performance of a secret key agreement scheme are quantitatively described by the secret key rate. In particular, this rate can be defined as the ratio of the number of final

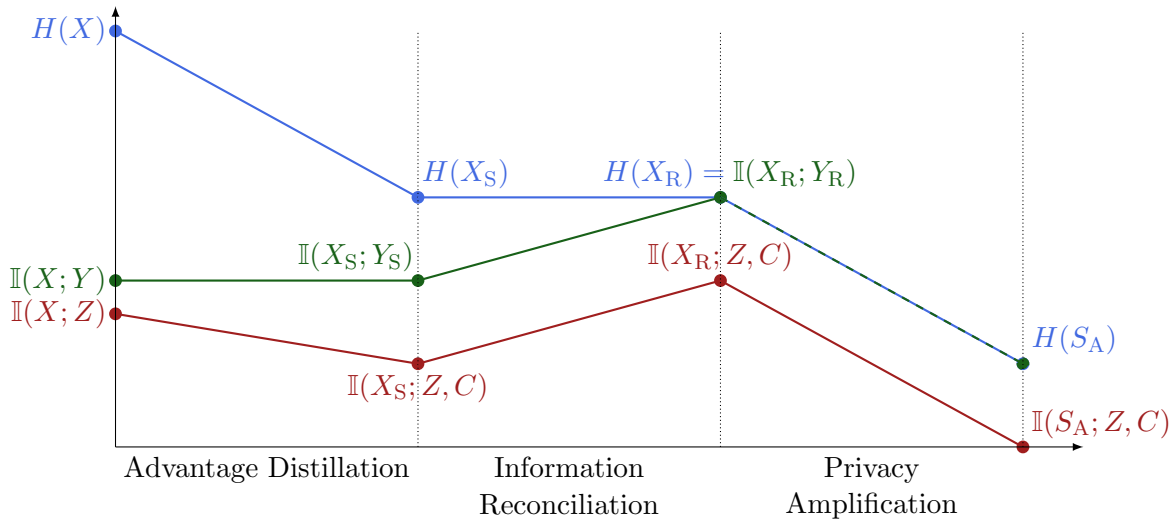


Figura 3.3: Evolution of information-theoretic measures involved in a secret key agreement protocol.

key bits to the number of sent raw key bits or to the number of sifted bits; in the first case, we denote this quantity as r_{raw}, in the second as r_{sift}. Hence, if we denote by $\ell(\cdot)$ the operator length of a string, we can define these rates as

$$r_{\text{raw}} \triangleq \frac{\ell(S)}{\ell(X)}, \quad r_{\text{sift}} \triangleq \frac{\ell(S)}{\ell(X_S)}. \tag{3.6}$$

Secret key rates can be derived in the asymptotic limit, that is, by assuming keys of infinite length, or by taking into account non-idealities of practical solutions, such as finite-length effects and inefficiencies of the protocol. Asymptotic secret key rates will be discussed for the specific QKD protocols in §3.4, whereas finite-length effects will be treated in §3.6.2.

3.2 Taxonomy of QKD protocols

Quantum key distribution protocols are nothing but secret key agreement schemes, where the insecure channel consists of a quantum channel. Depending on the key distribution technique and on the information coding scheme [51], QKD schemes can be distinguished into different families. In general, the system model for secret key agreement proposed in figure 3.1 is not suitable for describing all QKD protocols, but is appropriate for the sub-class of schemes we hereby consider, as explained at the end of this section.

As for the distribution technique, there exist two types of QKD protocols: *prepare-and-measure* (PM) and *entanglement-based* (EB) schemes. In PM protocols (see, e.g., [52, 53]), Alice prepares a sequence of quantum signals and sends them through a quantum channel to the receiver, Bob, who measures them. On the contrary, in EB schemes, pairs of entangled signals (see, e.g., [54]) are emitted by an entanglement source and then measured by Alice and Bob, who then act as two separate receivers. As shown in [55], in some scenarios EB protocols are essentially equivalent to PM schemes. Nevertheless, they

allow for a relaxation of the security assumptions typically made in QKD security proofs, opening the way to the so-called *device independent security* (see e.g., [56, 57, 58].

The two families of QKD schemes described above can be further distinguished into three main categories, which differ in the information coding technique: *discrete-variables coding* (DV-C), *continuous-variables coding* (CV-C) and *distributed-phase-reference coding* (DPR-C). DV-C is the first technique which has been proposed for QKD and, to date, is probably the most used. According to this approach, information is encoded in a discrete quantum degree of freedom of photons; in particular, widely used solutions are photon polarization for free-space implementations and phase coding for fiber-based implementations. The receiver uses a photon detector and only the events which resulted in a detection are taken into account for key distillation. DV-C protocols have the main advantage that, if the quantum channel is error-free and no eavesdropper is tampering with it, the legitimate parties immediately share a perfect secret key. On the other hand, they suffer from a low efficiency of photon detectors, high dark count rates and rather long dead-times, thus resulting in high overall losses [51]. These limitations were the driving reasons for the introduction of CV-C protocols, which are based on the measurement of quadrature components of light by means of *homodyne detection* (see, e.g., [59, 60]). Despite getting rid of hardware limitations of photon detectors, however, in CV-C implementations losses translate into noise,[1] resulting in a decrease of the signal-to-noise ratio. This entails an overhead in the information reconciliation procedure, which is now required to deal with noisier signals. In order to overcome the drawbacks of both the DV-C and the CV-C protocols, some experimental groups proposed a new approach to QKD, where the raw key bits are encoded into discrete quantum states (as for DV-C) and the quantum channel is monitored by observing the phase coherence of subsequent pulses. In that way, the communication efficiency can be improved with respect to DV-C and CV-C schemes. As previously mentioned, this class of protocols is referred to as DPR-C; examples of such protocols can be found in [61] and in [62].

In this thesis, we concentrate on discrete-variables, prepare-and-measure schemes. Please note, however, that part of the described classical algorithms can be straightforwardly extended for EB protocols. As for PM protocols, we may reformulate the system model shown in figure 3.1 for quantum key distribution as in figure 3.4. Here, $|\psi_X\rangle$ denotes the qubit prepared and sent by Alice and $|\psi_Y\rangle$ denotes the qubit received by Bob; also, the quantum operator $\mathcal{F}[\cdot]$ represents the interaction of the eavesdropper with the quantum channel. The nature of such interaction will be clarified in section 3.3.2.

3.3 Security of quantum key distribution protocols

In this section, we formalize the security of quantum key distribution protocols, starting from recalling the fundamental laws of quantum mechanics which enable this secret key agreement scheme. We then describe the attack models as applied to such protocols and we

[1] as said in [51], "because of the uncertainty principle, the measurement of complementary quadratures is intrinsically noisy".

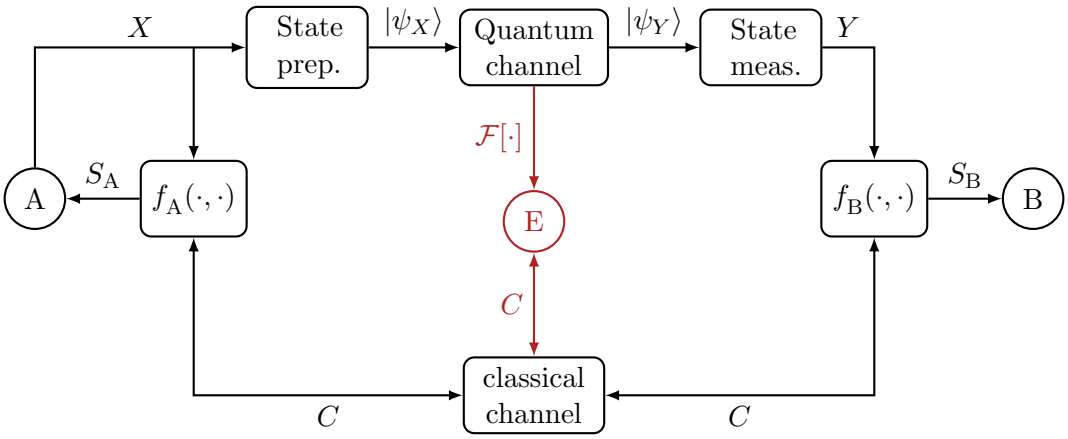


Figura 3.4: System model for prepare-and-measure QKD protocols.

finally provide some appropriate secrecy measures, which depend on the assumed attack scenario.

3.3.1 Quantum principles for security

As for the quantum key distribution protocols hereby considered (i.e., DV-C, PM), there are two fundamental results stated in the theory of quantum mechanics which are exploited in the construction of the secret key agreement scheme.

The first one is formalized in the following lemma.

Lemma 6. *(Information-disturbance lemma [63]). In a quantum system, one cannot take a measurement without perturbing the measured system itself.*

The lemma essentially follows from the third postulate of Quantum Mechanics [64]: the act of measuring an unknown quantum state produces a probabilistic outcome and, after the measure, the state itself collapses into a specific state,[2] so that further measurements always produce the same result. This lemma has the fundamental consequence that in a quantum scenario, differently from a classical setting, passive attacks may degrade the quality of communications, i.e., increase bit error rate and/or losses at the receiver. It should be stressed, however, that "Alice and Bob would be mistaken if they were to take a low error rate as evidence that a particular transmission is secure. [...] A combination of a low error rate and high information leakage is unlikely no matter what strategy the eavesdropper uses - as distinct from the (false) assertion that high information leakage is unlikely given a low error rate" [65].

The second fundamental result is the so-called *no-cloning theorem*:

Theorem 5. *(No-cloning theorem [66]). A quantum state cannot be cloned while keeping the original state unmodified.*

Dimostrazione. We here report the proof provided in [64]. Let us consider a quantum system $\mathcal{H}$ in a state $|\psi\rangle$. We assume that an eavesdropper wants to copy it to another

[2] the state is not perturbed if and only if it is an eigenstate of the measurement operator.

system $\mathcal{H}_E$, which is in a generic initial state $|\rho_E\rangle$. Hence, we want the composite system $\mathcal{H} \otimes \mathcal{H}_E$ to evolve from the initial state $|\psi\rangle \otimes |\rho_E\rangle$ to the state $|\psi\rangle \otimes |\psi\rangle$. It should therefore exist a unitary operator U on $\mathcal{H} \otimes \mathcal{H}_E$ such that, for any $|\psi\rangle \in \mathcal{H}$,

$$U\left(|\psi\rangle \otimes |\rho_E\rangle\right) = |\psi\rangle \otimes |\psi\rangle. \tag{3.7}$$

Since (3.7) holds for an arbitrary $|\psi\rangle$, we can also write, for any $|\phi\rangle \neq |\psi\rangle$,

$$U\left(|\phi\rangle \otimes |\rho_E\rangle\right) = |\phi\rangle \otimes |\phi\rangle. \tag{3.8}$$

Now, given the linearity of the tensor product, we get

$$\begin{aligned} U\left[(\alpha|\psi\rangle + \beta|\phi\rangle) \otimes |\rho_E\rangle\right] &= \alpha|\psi\rangle \otimes |\psi\rangle + \alpha|\phi\rangle \otimes |\phi\rangle & (3.9)\\ &\neq (\alpha|\psi\rangle + \beta|\phi\rangle) \otimes (\alpha|\psi\rangle + \beta|\phi\rangle), & (3.10) \end{aligned}$$

thus showing that no unitary transformation exists allowing the copy of a quantum state according to (3.7). □

Again, this result provides a significant advantage over classical systems, where information can be easily copied without being noticed (e.g., think of an arbitrary wireless channel, where information is readily available to any party in the transmission range).

3.3.2 Attack models

Quantum key distribution has the objective of providing an unconditionally secure cryptographic keys. In particular, bounds on the secrecy level can be derived depending on the assumed attacker model. According to the traditional classification, three attacks categories, with increasing generality, can be distinguished:

Individual attacks. Individual attacks (IAs) [67] are the most constrained attacks on a quantum key distribution system, in that they assume that [50]:

(IA.1) the eavesdropper attacks each qubit independently of all others and by using the same strategy.

(IA.2) quantum measurements on eavesdropped qubits are perfomed *before* the classical post-processing.

A remarkable example of individual attacks are the so-called *intercept-and-resend* (IR) attacks. IR attacks are, in fact, a subset of individual attacks, where Eve independently intercepts the transmitted qubits, performs a measurement on them and, according to the obtained result, she prepares a new quantum state which she then sends to Bob. A practical example of such attacks is provided in section 3.4.1

Collective attacks. Collective attacks (CAs) [68] are a generalization of IAs, and are defined by the following properties:

(CA.1) the eavesdropper attacks each qubit independently of all others and by using the same strategy (as (IA.1)).

(CA.2) Eve can take advantage of a quantum memories to store intercepted qubits and postpone the measurement to any later time convenient to her.

General attacks. General attacks (GAs) are the most general family of attacks on QKD systems, which drop the assumption that the eavesdropper interacts with each quantum signal independently (i.e., conditions (IA.1) and (CA.1)), and are then also known as *joint* or *coherent* attacks. GAs, in fact, do not impose any restriction on the attack strategy, nor on the use of quantum memories.

3.3.3 Secrecy measures

As mentioned above, the security of a QKD system can be defined according to the assumed eavesdropping model. In particular, the more general the attacker model, the more strict are the conditions to be verified by the final key. In this section, we briefly recall some security definitions proposed in the literature as applied to the quantum key distribution problem.

We start by describing a general framework for defining the security of a key S given an eavesdropper who has access to a quantum system E. More specifically, we here consider a generic information measure $\mathbb{I}(\cdot;\cdot)$, whose nature depends on the assumed attacker model and we define a secrecy and a uniformity constraint:

$$(\varepsilon_{\text{sec}}\text{-secrecy}) \qquad \mathbb{I}(S;E) \leq \varepsilon_{\text{sec}}, \tag{3.11}$$

$$(\varepsilon_{\text{u}}\text{-uniformity}) \qquad H(S) \geq \ell(S) - \varepsilon_{\text{u}}. \tag{3.12}$$

Intuitively, (3.11) bounds the information that the eavesdropper has on the final key, whereas (3.12) bounds the key distribution to be ε_{u}-close to uniform. Also, we require that the key which is output to a quantum key distribution system and, more in general, to any secret key agreement scheme, is *composable* with further cryptographic protocols, that is, the final key should be securely usable in any distinct cryptographic application.[3]

In the history of cryptosystems, the first secrecy measure that has been proposed was the one based on classical mutual information, defined by Claude Shannon in his seminal work [49]. More specifically, given the random variable representing the key, S, and the random variable summarizing the classical information available to the eavesdropper, Z, Shannon proves that S is perfectly secret ($\varepsilon_{\text{sec}} = 0$) if

$$I(S;Z) = 0, \tag{3.13}$$

thus meaning that S and Z are statistically independent.

In practical applications of classical information-theoretic cryptography, condition (3.13) is typically relaxed so that the correlation between S and Z is bounded by a small ε_{sec},

[3] we are not delving into the composability issue; the interested reader can find more details, e.g., in [69]

that is,

$$I(S;Z) \leq \varepsilon_{\text{sec}}. \tag{3.14}$$

This definition, however, assumes that the information available to the eavesdropper is classical and, therefore, it should be extended for being applied to a quantum adversarial scenario. This is the motivation behind the introduction of a new secrecy measure, based on the *accessible information* [70]. Given the quantum system of the eavesdropper, E, the accessible information is defined as the maximum mutual information that the attacker shares on the key by using her optimal strategy, that is,

$$I_{\text{acc}}(S;E) = \max_{\mathcal{M}} I(S;\mathcal{M}[E]), \tag{3.15}$$

where "the maximum is taken over all local measurements given by a positive operator-valued measure $\mathcal{M}$ on E and where $I(S;\mathcal{M}[E])$ denotes the mutual information between S and the measured outcome $\mathcal{M}[E]$" [70]. According to this definition, condition (3.14) can be therefore rewritten as

$$I_{\text{acc}}(S;E) \leq \varepsilon_{\text{sec}} \tag{3.16}$$

It has been proven that this secrecy definition is sound if hypothesis (IA.2) holds, that is, if we only consider attack scenarios where the eavesdropper cannot take advantage of a quantum memory, as for individual attacks. On the other hand, if collective or general attacks are considered, condition (3.16) no longer ensures the secrecy of the final key. In [70, Proposition 1], in fact, the authors prove that the accessible information is lockable, that is, "one additional bit of information can increase the accessible information by more than one bit". More formally:

[70, Proposition 1]. For any $\varepsilon > 0$ there exists a quantum state on a system E which depends on the classical random variable S and an m-bit string W, where m is linear in $\log(1/\varepsilon_{\text{sec}})$, such that the following holds:

1. $I_{\text{acc}}(S;E) \leq \varepsilon_{\text{sec}}$
2. $I_{\text{acc}}(S;W,E) \geq H(W) + 1$

In this sense, the security definition based on $I_{\text{acc}}(\cdot)$ is not composable when general attacks are considered [69], since, if the locking property is exploited, it does not allow to bound the actual information that the eavesdropper has on the final key. However, while in a long-term perspective (more than 50 years) security against general attacks is the goal, in the near future (5-10 years), we know that an ideal intercept-and-resend (IR) attack is the best option that an eavesdropper can choose because the quantum memory needed for a general or coherent attack is not yet available. In §4.3.4, we will show that there are situations in which no key can be extracted if general security is required, while a pragmatically secure secret key can be obtained. In these cases, requiring general security, a protection far above actual possibilities of an eavesdropper, prevents key generation.

We therefore define the notion of *pragmatic secrecy*, that is, providing security against IR attacks, by extending condition (3.16).

Definition 22. [J1] A key S is δ_{sec}-PS (pragmatic secret) if, for any IR attack strategy,

$$H(\mathbf{U_S}) - H(S|V) \leq \delta_{\text{sec}} \tag{3.17}$$

being $\mathbf{U_S}$ the uniform key with the same length as S, V the classical random variable which summarizes all the information available to the eavesdropper and $H(S|V)$ the equivocation (conditional entropy) of S given V.

The above definition of pragmatic secrecy implies both the uniformity and the secrecy of the key, as stated in the following proposition [J1].

Proposition 7. *The pragmatic security definition (3.17) implies the following bounds:*

$$\begin{cases} H(S) \geq H(\mathbf{U_S}) - \delta_{\text{sec}} & \textit{(uniformity)} \\ I_{\text{acc}}(S;E) \leq \delta_{\text{sec}} & \textit{(secrecy)} \end{cases} \tag{3.18}$$

Dimostrazione. The uniformity condition trivially derives from the fact that $H(S|V) \leq H(S)$. Also, from basic information theory, we know that

$$I(S;V) = H(S) - H(S|V) \leq H(\mathbf{U_S}) - H(S|V), \tag{3.19}$$

since S has maximal entropy if and only if it is uniformly distributed. Now, since condition (3.17) is verified for any IR attack strategy, and therefore for any outcome V of the eavesdropper measurement on the quantum system E, the security condition directly follows. □

We stress that, as for incoherent individual attacks, Eq. (3.17) guarantees composable security, as the eavesdropper, without a quantum memory, cannot exploit the "locking property" of the accessible information. Hence, pragmatic secrecy, unlike computational secrecy, offers forward security: if a key is produced today with pragmatic secrecy (without quantum memory available for Eve), the key or a message encrypted with it will be secure for any future use.

A definition which ensures composable security against general attacks is the one proposed in [24, 71], which is based on the trace distance $\|\cdot\|_1$. More specifically, given the quantum state of the adversary, ρ_E, and the classical-quantum state describing the classical key S together with the quantum knowledge of the adversary, defined as

$$\rho_{SE} = \sum_{s\in\mathcal{S}} P_S(s)|s\rangle\langle s| \otimes \rho_{E|S=s}, \tag{3.20}$$

where $\{|s\rangle\}_{s\in\mathcal{S}}$ are the orthonormal states representing the value of S, we define general secrecy as follows.

Definition 23. The key S is said to be ε_{sec}-GS (general secret) with respect to ρ_E if

$$\frac{1}{2}\|\rho_{SE} - \rho_U \otimes \rho_E\|_1 \leq \varepsilon_{\text{sec}}. \tag{3.21}$$

The composability of this criterion follows from two properties of the trace distance [24], namely:

- $\|\cdot\|_1$ is sub-additive with respect to tensor products:

$$\forall\ \rho,\sigma \in \mathcal{H}_S,\ \rho',\sigma' \in \mathcal{H}_E: \quad \frac{1}{2}\|\rho\otimes\rho' - \sigma\otimes\sigma'\|_1 \leq \frac{1}{2}\left(\|\rho-\sigma\|_1 - \|\rho'-\sigma'\|_1\right) \tag{3.22}$$

- $\|\cdot\|_1$ cannot increase when the same quantum operator $\xi[\cdot]$ is applied to both arguments:

$$\frac{1}{2}\|\xi[\rho] - \xi[\sigma]\|_1 \leq \frac{1}{2}\|\rho-\sigma\|_1 \tag{3.23}$$

These two results, in fact, ensure that no locking property can be exploited even for general quantum attacks: "since the trace distance does not increase when appending an additional quantum system (eq (3.22)) or when applying any arbitrary quantum operation (Eq. (3.23)), this also hold for any further evolution of the system" [24]. The authors of [24] also provide an operational meaning for condition (3.21), by claiming that if a key S is ε_{sec}-secret according to (3.21) then S cannot be distinguished from a perfectly secure key U, uniformly distributed on the key space and independent of the adversary's information, with a probability higher than $1-\varepsilon_{\text{sec}}$.[4]

To conclude, let us prove the following relationship between *pragmatic* and *general* secrecy.

Proposition 8. *For non-coherent attacks, δ_{sec}-PS secrecy implies ε_{sec}-GS secrecy for $\delta_{\text{sec}} = \frac{2}{\ln 2}\varepsilon_{\text{sec}}^2$.*

Dimostrazione. The Pinsker inequality (see section 11.6 in [2] and [79]) ensures that

$$\frac{1}{2}\|p_{SV} - u_S q_V\|_1 \leq \sqrt{\frac{\ln 2}{2}\mathbb{D}(p_{SV}||u_S p_V)} \tag{3.24}$$

$$= \sqrt{\frac{\ln 2}{2}(H(\mathbf{U_S}) - H(S|V))} \tag{3.25}$$

where u_S is the uniform distribution on S and $\mathbb{D}(p||q)$ is the relative entropy between the p and q distributions.

It is then straightforward to see that

$$H(\mathbf{U_S}) - H(S|V) \leq \frac{2}{\ln 2}\varepsilon_{\text{sec}}^2 \Rightarrow \frac{1}{2}\|p_{SV} - u_S p_V\|_1 \leq \varepsilon_{\text{sec}}. \tag{3.26}$$

□

[4] security definition (3.21) has been widely accepted and is currently used in most QKD security proofs (see, e.g., [72, 73, 74]). In a recent debate in the QKD community [75, 76, 77, 78], however, it was argued that, unless ε_{sec} is chosen exponentially small in the key size, (3.21) does not capture the actual distinguishing advantage of the adversary. The debate is still opened and, as a guideline, the choice of the parameter ε_{sec} should be tailored depending on the required level of secrecy and on the final key length.

3.4 Explicit QKD protocols

In this section, we describe some remarkable examples of QKD protocols, which were implemented in the experiments that were performed as part of this work: BB84 [53], efficient BB84 [74] and B92 [52]. Besides the described solutions, there exist some other schemes, such as the six-states [80] and the SARG [81] protocols; for a complete overview, the interested reader could refer to [82] and [51].

3.4.1 Bennett-Brassard 1984 protocol (BB84)

The *Bennett-Brassard 1984* protocol [53] (whence the acronym BB84) was the first QKD scheme that has been proposed. BB84 is a DV-C, PM protocol, where information is encoded into single polarized photons. More specifically, there exist two polarization bases, hereby denoted by $\mathbb{X}$ and $\mathbb{Z}$, each one specified by a pair of orthogonal polarization states. $\mathbb{X}$ is called *horizontal-vertical basis* and is specified by the quantum states $|\leftrightarrow\rangle$ and $|\updownarrow\rangle$, whereas $\mathbb{Z}$ is referred to as the *diagonal basis* and is specified by the quantum states $|\nearrow\rangle$ and $|\nwarrow\rangle$. For each basis, a bit-qubit map is defined and Alice, who prepares the state to be sent throughout the quantum channel, chooses in which basis to encode each bit. A possible map is shown in table 3.1.

Bit	$\mathbb{X}$-basis	$\mathbb{Z}$-basis
0	$\leftrightarrow$	$\nwarrow$
1	$\updownarrow$	$\nearrow$

Tabella 3.1: Map bit-qubit for the BB84 protocol.

Given this coding map, the BB84 protocol is specified by the following procedure, which can be reiterated according to the desired key length:

1. **Quantum transmission**

   (a) Alice randomly generates:

      i. a sequence of bits, $\{x_m\}$, i.i.d. in $\{0, 1\}$;

      ii. a sequence of state-preparation bases, $\{A_m\}$, i.i.d. in $\{\mathbb{X}, \mathbb{Z}\}$.

   (b) For each random bit x_i, Alice prepares a qubit $|\psi_{x_i}\rangle$, in the form of a single photon polarized in the A_i basis, and sends it through the quantum channel.

   (c) Bob randomly generates a sequence of state-measurement bases, $\{B_m\}$, i.i.d. in $\{\mathbb{X}, \mathbb{Z}\}$.

   (d) Bob measures each received qubit, $|\psi_{y_i}\rangle$, in the B_i basis and gets the measured bit y_i.

2. **Sifting** (*advantage distillation*)

   Alice and Bob exchange, through the public channel, the state-preparation and the state-measurement bases, $\{A_m\}$ and $\{B_m\}$, and discard the bits for which their choice differs, i.e., the sifted bits at Alice and Bob are, respectively,

$$Y_{\mathrm{S}} \quad = \quad \{y_i : A_i = B_i\}. \tag{3.27}$$

3. **Error estimation**

Bob sends to Alice (or viceversa) a random subset $\mathcal{C}$ of sifted key bits for estimating the bit error rate Q in the quantum channel (QBER). More specifically, the estimated qber $\hat{Q}$ is computed as follows:

$$\hat{Q} = \frac{\sum_{c\in\mathcal{C}} x_{\mathrm{S,c}} \oplus y_{\mathrm{S,c}}}{|\mathcal{C}|}. \tag{3.28}$$

The estimated QBER is a crucial parameter for the classical post-processing phase, that is, for information reconciliation and for privacy amplification, and affects the secret key rate as shown in (3.30).

x_m	0	1	1	0	0	1	1	1
A_m	$\mathbb{X}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{X}$	$\mathbb{Z}$	$\mathbb{X}$	$\mathbb{X}$	$\mathbb{Z}$
$\lvert\psi_{x_m}\rangle$	$\lvert\leftrightarrow\rangle$	$\lvert\nearrow\rangle$	$\lvert\swarrow\rangle$	$\lvert\leftrightarrow\rangle$	$\lvert\searrow\rangle$	$\lvert\updownarrow\rangle$	$\lvert\updownarrow\rangle$	$\lvert\swarrow\rangle$
B_m	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{X}$	$\mathbb{X}$	$\mathbb{X}$	$\mathbb{Z}$	$\mathbb{X}$	$\mathbb{Z}$
y_m	1	1	0	0	1	1	1	1
$y_{\mathrm{S,m}}$	-	1	-	0	-	-	1	1

Tabella 3.2: Example of quantum key distribution (up to the sifting phase) according to the BB84 protocol.

It is straightforward to see that if the quantum channel is ideal and if no adversary is attacking the system, the BB84 protocol directly produces a secret key. On the contrary, if an attacker is present, she is forced to interact with the quantum channel *before* she knows the chosen state-preparation and state-measurement bases. Hence, the sifting procedure, which is in fact a specific instance of advantage distillation (see §3.1) as applied to QKD, enables the legitimate parties to get an advantageous position with respect to the eavesdropper. This, however, comes at the price of a rate reduction, since, on average, only 1/2 of the times Alice's and Bob's basis choice will match. More specifically, we say that the raw-sifted efficiency of the BB84 protocol, in the absence of losses, is

$$\eta_{\mathrm{BB84}} = \frac{1}{2}. \tag{3.29}$$

Furthermore, the asymptotic (secret to sifted) key rate for the BB84 protocol has been derived in [83], and reads as follows:

$$r_{\mathrm{BB84}} = 1 - 2h_2(Q). \tag{3.30}$$

Let us now describe what happens if an eavesdropper independently attacks each sent qubit with probability p, by first measuring and by then resending it to Bob; this attack is

known as *intercept-and-resend* (IR) (§3.3.2). In this scenario, Eve has to choose a state-measurement basis for extracting the information out of an intercepted qubit.It was shown [84] that the best she can do is to mimic a legitimate receiver, i.e., to choose uniformly at random the measurement basis, E_i. Hence, she guesses the state preparation basis with probability 1/2, and she takes the wrong basis with probability 1/2. When she picks the correct basis, she also gets the right result, whereas when she picks the wrong basis, she gets a uniformly random result, thanks to the complete non-orthogonality of the states in the $\mathbb{X}$ and in the $\mathbb{Z}$ bases. She then retransmits the measured bit to Bob, by preparing it with the same basis she used for the measurement. In the hypothesis of a fixed QBER Q and of an ideal attack setup, that is, with no errors or losses introduced by the eavesdropper's devices, we derive the bit error rate in the received sequence under IR attack rate q as

$$Q_{\mathrm{IR}}(q) = (1-q)Q + q\left(\frac{Q}{2} + \frac{1}{4}\right) = \left(1 - \frac{q}{2}\right)Q + \frac{q}{4}, \tag{3.31}$$

whereas losses will not be affected. Therefore, the QBER measured at Bob linearly increases with the attack rate q, up to its maximum value,

$$Q_{\mathrm{IR}}(1) = Q + \frac{1}{4}. \tag{3.32}$$

Hence, Eve has to find a trade-off for getting as much information as possible without been noticed. At the same time, Alice and Bob aim at estimating as precisely as possible the attack rate q, in order to compensate for the eavesdropped information during the privacy amplification phase (see §3.6 and §4.2).

3.4.2 Efficient BB84 protocol

As seen in section 3.4.1, the BB84 protocol has a raw-sifted efficiency $\eta_{\mathrm{BB84}} = 1/2$, that is, in the absence of losses, only half of the raw bits sent by Alice yields a sifted sequence at Bob. With the aim of increasing this efficiency, a variant of the BB84 protocol, that we refer to as *efficient BB84* (e-BB84), has been proposed in [74].

In standard BB84, both polarization bases are used for raw key transmission and for attack estimation, and their choice is unbiased. In e-BB84, instead, one basis (say $\mathbb{X}$) carries the raw key sequence, whereas the other basis (say $\mathbb{Z}$) is used for eavesdropping detection. Also, the choice of the two bases at Alice and Bob is biased: intuitively, the basis carrying the raw key is chosen with higher probability, while the detection basis is chosen less frequently. Let us describe it in more detail.

The e-BB84 protocol is characterized by the sifted key length n and by the number of bits used for parameter estimation k; both parameters can be chosen according to the required secret key length and channel conditions as described below. Also, the choice of n and k yields the probability of picking each of the two bases, namely,

$$p_{\mathbb{X}} = \frac{1}{1 + \sqrt{k/n}}, \quad p_{\mathbb{Z}} = 1 - p_{\mathbb{X}}. \tag{3.33}$$

The protocol consists of the following subsequent steps:

1. **Quantum transmission**

   (a) Alice randomly generates:

      i. a sequence of bits, $\{x_m\}_{m\in[1,M]}$, i.i.d. in $\{0,1\}$;

      ii. a biased sequence of state-preparation bases $\{A_m\}_{m\in[1,M]}$ in $\{\mathbb{X},\mathbb{Z}\}$, chosen with probabilities $p_{\mathbb{X}}$ and $p_{\mathbb{Z}}$, respectively.

   where M is such that the condition in the sifting phase is met.

   (b) For each random bit x_i, Alice prepares a qubit $|\psi_{x_i}\rangle$, in the form of a single photon polarized in the A_i basis, and sends it through the quantum channel.

   (c) Bob randomly generates a biased sequence of state-measurement bases, $\{B_m\}$ in $\{\mathbb{X},\mathbb{Z}\}$, chosen with probabilities $p_{\mathbb{X}}$ and $p_{\mathbb{Z}}$, respectively.

   (d) Bob measures each received qubit, $|\psi_{y_i}\rangle$, in the B_i basis and gets the measured bit y_i.

2. **Sifting** (*advantage distillation*)

   Alice and Bob exchange, through the public channel, the state-preparation and the state-measurement bases, $\{A_m\}$ and $\{B_m\}$, and discard the bits for which their choice differs. Also, they distinguish the bits measured in the $\mathbb{X}$ basis and the ones measured in the $\mathbb{Z}$ basis, thus defining the following subsets:

$$\mathcal{X} = \{i : A_i = \mathbb{X}, B_i = \mathbb{X}\} \tag{3.34}$$

$$\mathcal{Z} = \{i : A_i = \mathbb{Z}, B_i = \mathbb{Z}\}. \tag{3.35}$$

   The quantum communication is repeated as long as either $|\mathcal{X}| < n$ or $|\mathcal{Z}| < k$. Then, Alice and Bob pick the same n and k indexes, randomly chosen, in $\mathcal{X}$ and in $\mathcal{Z}$, respectively, thus defining the subsets $\mathcal{X}_n$ and in $\mathcal{Z}_k$. Finally, the following sifted sequences are defined:

$$\mathbf{X}_{\mathbb{X}} = \{X_{\mathbb{X},i}\} = \{x_i : i \in \mathcal{X}_n\}, \quad \textit{(sifted key at A)} \tag{3.36}$$

$$\mathbf{Y}_{\mathbb{X}} = \{Y_{\mathbb{X},i}\} = \{y_i : i \in \mathcal{X}_n\}, \quad \textit{(sifted key at B)} \tag{3.37}$$

$$\mathbf{X}_{\mathbb{Z}} = \{X_{\mathbb{Z},i}\} = \{x_i : i \in \mathcal{Z}_k\}, \quad \textit{(estimation bits at A)} \tag{3.38}$$

$$\mathbf{Y}_{\mathbb{Z}} = \{Y_{\mathbb{Z},i}\} = \{y_i : i \in \mathcal{Z}_k\}, \quad \textit{(estimation bits at B)} \tag{3.39}$$

3. **Error estimation**

   Bob sends to Alice the whole sequence of estimation bits $\{Y_{\mathbb{Z},m}\}$ for computing the bit error rate $Q_{\mathbb{Z}}$ on the eavesdropping detection basis, yielding:

$$Q_{\mathbb{Z}} = \frac{\sum_{c\in\mathcal{Z}} X_{\mathbb{Z},c} \oplus Y_{\mathbb{Z},c}}{|\mathcal{Z}|} \tag{3.40}$$

Again, this QBER is a crucial design parameter for the classical post-processing phase, and, in particular, for the privacy amplification phase. On the other hand, the QBER on the $\mathbb{X}$-basis, $Q_{\mathbb{X}}$, is the main design parameter for the information reconciliation phase, and should be known to the legitimate parties. In section 4.3 we will describe in detail how these parameters influence the key distillation procedure.

As it can easily be seen, the efficiency of e-BB84 is

$$\eta_{\mathrm{e-BB84}} = p_{\mathbb{X}}^2, \tag{3.41}$$

and is therefore higher than η_{BB84} (see Eq. (3.29)) as soon as $p_{\mathbb{X}} > 1/\sqrt{2}$. Also, the asymptotic (secret to sifted) key rate for the efficient BB84 directly follows from the one of BB84, that is:

$$r_{\mathrm{e-BB84}} = 1 - h_2(Q_{\mathbb{X}}) - h_2(Q_{\mathbb{Z}}). \tag{3.42}$$

3.4.3 Bennett 1992 protocol

The last QKD protocol we describe is the *Bennett 1992* (B92) scheme [52]. B92 is a DV-C, PM protocol, where information is encoded in two non-orthogonal quantum states. In particular, a state-preparation basis, $\mathbb{P}$, and a state measurement basis, $\mathbb{M}$, are defined, so that the following map is defined:

Bit	$\mathbb{P}$-basis	$\mathbb{M}$-basis
0	$\updownarrow$	$\searrow$
1	$\swarrow$	$\leftrightarrow$

Tabella 3.3: State-preparation and state-measurement basis for the B92 protocol.

The protocol works as follows:

1. **Quantum transmission**

   (a) Alice randomly generates a sequence of bits, $\{x_m\}$, i.i.d. in $\{0, 1\}$

   (b) For each random bit x_i, Alice prepares a qubit $|\psi_{x_i}\rangle$, in the form of a single photon polarized in the corresponding $\mathbb{P}$-basis state, and sends it through the quantum channel. Please note that the choice of the $\mathbb{P}$-basis state is now deterministic, whereas in the BB84 protocol it was random.

   (c) Bob randomly generates a sequence of $\mathbb{M}$-basis states, $\{B_m\}$, i.i.d. in $\{\mathbb{P}, \mathbb{M}\}$.

   (d) Bob projects each received qubit, $|\psi_{y_i}\rangle$, onto the B_i polarization, and gets the measured bit y_i. If $|\psi_{y_i}\rangle$ is orthogonal to B_i, then no detector clicks; otherwise, the right detector clicks with probability 1/2 and does not click with probability 1/2.

2. **Sifting** (*advantage distillation*)

Bob sends to Alice, through the public channel, the indexes $\mathcal{D}$ of the the qubits that produced a click at the receiver. The sifted bits at Alice and Bob are, respectively,

$$X_{\mathrm{S}} = \{x_i : i \in \mathcal{D}\}; \tag{3.43}$$

$$Y_{\mathrm{S}} = \{y_i : i \in \mathcal{D}\}. \tag{3.44}$$

3. **Error estimation**

Bob sends to Alice (or viceversa) a random subset $\mathcal{C}$ of sifted key bits for estimating the bit error rate Q in the quantum channel (QBER). More specifically, the estimated qber $\hat{Q}$ is computed as follows:

$$\hat{Q} = \frac{\sum_{c\in\mathcal{C}} x_{\mathrm{S,c}} \oplus y_{\mathrm{S,c}}}{|\mathcal{C}|} \tag{3.45}$$

x_m	0	1	1	0	0	1	1	1
$\lvert\psi_{x_m}\rangle$	$\lvert\leftrightarrow\rangle$	$\lvert\nearrow\rangle$	$\lvert\swarrow\rangle$	$\lvert\leftrightarrow\rangle$	$\lvert\leftrightarrow\rangle$	$\lvert\swarrow\rangle$	$\lvert\nearrow\rangle$	$\lvert\swarrow\rangle$
B_m	$\lvert\updownarrow\rangle$	$\lvert\updownarrow\rangle$	$\lvert\nwarrow\rangle$	$\lvert\nwarrow\rangle$	$\lvert\updownarrow\rangle$	$\lvert\nwarrow\rangle$	$\lvert\updownarrow\rangle$	$\lvert\updownarrow\rangle$
y_m	-	-	-	0	-	-	1	-

Tabella 3.4: Example of quantum key distribution (up to the sifting phase) according to the B92 protocol.

The efficiency of the B92 protocol (in the absence of losses) immediately follows from the described scheme

$$\eta_{\mathrm{B92}} = \frac{1}{2}\left(P[x_i = 0, B_i = |\nwarrow\rangle] + P[x_i = 1, B_i = |\nearrow\rangle]\right) = \frac{1}{4}. \tag{3.46}$$

Hence, B92 has a consistently lower efficiency as compared with BB84, and a fortiori, with e-BB84. On the other hand, B92 relies on a simplified setup, which requires just two non orthogonal states at both the transmitter and the receiver side, that is, half of the complexity of BB84. Unfortunately, the described setup also comes with a significant security threat. Due to the deterministic coding of qubits, in fact, an eavesdropper who plays the man-in-the-middle can mimic Bob's receiver, and re-transmit the qubits which produced a click. This attack, known as *unambiguous state discrimination* (USD) [85], introduces significant losses, yielding an overall efficiency of $(1/4)^2$ if each qubit is attacked, but does not affect the measured QBER at the receiver. In the original paper by Bennett [52], the use of a strong reference was suggested for avoiding this problem, but this enhanced protocol becomes insecure as soon as the channel losses get higher than a given threshold which depends on the non-orthogonality of the signal states [86]. A further solution for making the B92 protocol more robust against losses and noise is the one presented in [87], where the decoy-states principle (see §3.4.4) is extended to B92 by using additional uninformative states.

Finally, the asymptotic (secret to sifted) key rate for the B92 protocol is the same as for BB84, that is,

$$r_{\mathrm{B92}} = 1 - 2h_2(Q). \tag{3.47}$$

3.4.4 Remarks on the single photon assumption for practical QKD

All the protocols presented above assume that each qubit consists of a single photon, so that the eavesdropper cannot take advantage of multiple measurements for getting information on the sent state without perturbing the one which is delivered to Bob (this attack is called *Photon Number Splitting* (PNS)). This assumption, in fact, is often not strictly verified in experimental scenarios, as, until recently, faint laser pulses were used for the sake of generating single photons [82]. In that scenario, the photon generation follows a poissonian statistics, and there always exists a non-zero probability that more than one photon is generated, that is

$$P[n_{\mathrm{ph}} > 1 | n_{\mathrm{ph}} > 0] = \frac{1 - e^{-\mu}(1+\mu)}{1 - e^{-\mu}}. \tag{3.48}$$

where n_{ph} is the number of photons for encoding a generic bit at the transmitter output and is Poisson distributed with mean μ. In order to reduce the probability (3.48), one could decrease μ, at the price of a lower bit-rate; typical values are $\mu \in [0.1, 0.4]$.

As the corresponding rate reduction has a significant impact on the system performance, the scientific community looked for a solution to overcome this limitation while coping with the PNS attack. Besides the proposal of protocols which allowed to slightly increase μ while ensuring the same security level (SARG04, [88]), a solution suitable for generic protocols was devised in [89] and first adapted for realistic implementations in [90]. This solution is based on the use of *decoy states*: in addition to the qubits which encode the raw key bits, Alice occasionally sends pulses with different intensities which do not carry information, but are rather used for testing the presence of the eavesdropper. In particular, if Eve, who ignores the intensity of each signal, performs a PNS attack, she alters the statistics of photon detections at the receiver, thus allowing Alice and Bob to effectively bound the number of actual single photon pulses. There exist several works demonstrating decoy states as applied to QKD protocols, both in free-space (e.g., [91, 92]) and in fiber optics (e.g., [93]), and eventually including the analysis with finite statistics [94, 95]. In this thesis, we do not include decoy states neither in the analysis of the secret key rate nor in the experimental implementations, but existing literature results could be effectively integrated in the proposed analysis for extending the security of the considered systems to PNS attacks.

3.5 Information reconciliation

Information reconciliation is the second step of a secret key agreement protocol which allows to correct the errors between Alice and Bob, so that the final keys are equal with

high probability and the correctness constraint (3.2) is ensured.

Information reconciliation protocols can be classified according to two main features: the direction of the reconciliation and the error correction strategy. As for the first feature, three different approaches can be distinguished:

- *direct reconciliation* [96, 97, 98]: Alice's string is considered to be correct, i.e., $X_{\mathrm{R}} = X_{\mathrm{S}}$, and Bob's string is reconciled accordingly, i.e., $Y_{\mathrm{R}} = \hat{X}_{\mathrm{S}}$.
- *reverse reconciliation* [99, 60]: Bob's string is considered to be correct, i.e., $Y_{\mathrm{R}} = Y_{\mathrm{S}}$, and Alice's string is reconciled accordingly, i.e., $X_{\mathrm{R}} = \hat{Y}_{\mathrm{S}}$.
- *two-way reconciliation* [100, 101]: Alice's and Bob's sifted string are dynamically modified so that $X_{\mathrm{R}} \neq X_{\mathrm{S}}$, $Y_{\mathrm{R}} \neq Y_{\mathrm{S}}$ and $X_{\mathrm{R}} = Y_{\mathrm{R}}$ with high probability ($1 - \varepsilon_{\mathrm{cor}}$, according to constraint (3.2)).

As detailed in [102, Chapter 8], the choice of the best approach depends on the application scenario, being reverse reconciliation more suitable for continuous-variable QKD with high channel losses [60]. Two-way reconciliation, on the other hand, allows to cope with high bit error rates on the quantum channel; while its correction capability enables key distillation even in very noisy scenarios (up to $Q = 20\%$), its efficiency for $Q \lesssim 10\%$ is typically much lower than that of one-way protocols [100]. As in this work we concentrate on discrete-variables coding QKD protocols (see §3.2 for a classification of QKD schemes), we decided to focus on direct reconciliation schemes; in particular, we are going to describe some practical constructions at the end of this section.

3.5.1 Direct reconciliation: problem statement and coding approaches

Let $\mathbf{x}_{\mathrm{S}}$ and $\mathbf{y}_{\mathrm{S}}$ be the random vectors representing the n-bit sifted keys to be reconciled at the transmitter and at the receiver, respectively; furthermore, let $\hat{\mathbf{x}}_{\mathrm{S}}$ be the random vector representing the n-bit reconciled key at the receiver. Namely,

$$\mathbf{x}_{\mathrm{S}} = [x_{\mathrm{S},1}, x_{\mathrm{S},2}, \ldots, x_{\mathrm{S,n}}]^T \in \{0,1\}^n, \tag{3.49}$$

$$\mathbf{y}_{\mathrm{S}} = [y_{\mathrm{S},1}, y_{\mathrm{S},2}, \ldots, y_{\mathrm{S,n}}]^T \in \{0,1\}^n, \tag{3.50}$$

$$\hat{\mathbf{x}}_{\mathrm{S}} = [y_{\mathrm{R},1}, y_{\mathrm{R},2}, \ldots, y_{\mathrm{R,n}}]^T \in \{0,1\}^n, \tag{3.51}$$

where the superscript T represents the transposition operator. Information reconciliation has the aim of reliably reconstructing $\hat{\mathbf{x}}_{\mathrm{S}}$, given the received sifted key $\mathbf{y}_{\mathrm{S}}$ and the public communication, summarized by $\mathbf{c}$, so that the least possible amount of information is disclosed on the public channel. More formally, we can define the following objectives:

$$(\textit{reliability}) \qquad P[\mathbf{x}_{\mathrm{S}} \neq \hat{\mathbf{x}}_{\mathrm{S}}] < \varepsilon_{\mathrm{cor}}, \tag{3.52}$$

$$(\textit{min. information leakage}) \qquad L_{\mathrm{EC}} \triangleq \ell(\mathbf{c}) \approx \ell(\mathbf{x}_{\mathrm{S}}) H(X_{\mathrm{S}}|Y_{\mathrm{S}}), \tag{3.53}$$

where $\ell(\cdot)$ is the length operator and $H(\cdot|\cdot)$ is the conditional Shannon entropy.

In particular, the performance of an information reconciliation protocol can be expressed through the so-called *reconciliation efficiency* [98], i.e.,

$$\eta_{\mathrm{EC}} = \frac{\ell(\mathbf{c})}{\ell(\mathbf{x}_{\mathrm{S}})H(X_{\mathrm{S}}|Y_{\mathrm{S}})} \geq 1. \tag{3.54}$$

Therefore, the closest is η_{EC} to 1, the more efficient is the considered reconciliation protocol. Furthermore, since we consider the quantum channel to be a binary symmetric channel with transition probability Q, we get $H(X_{\mathrm{S}}|Y_{\mathrm{S}}) = h_2(Q)$, and (3.54) can be rewritten as

$$\eta_{\mathrm{EC}} = \frac{\ell(\mathbf{c})}{\ell(\mathbf{x}_{\mathrm{S}})h_2(Q)}. \tag{3.55}$$

The reconciliation efficiency is a crucial parameter for the performance and for the feasibility of quantum key distribution, as the information disclosed on the public channel must be compensated during privacy amplification (see section 3.6): the L_{EC} bits that are leaked for error correction, in fact, will decrease the final secret key length by the same amount. In this sense, lower values of η_{EC} allow to obtain longer secret keys or, in the limit, to distil a key when it would not be possible with less efficient protocols. In the following, reconciliation efficiencies will be evaluated and discussed for some practical protocols.

As mentioned above, different error correction strategies can be chosen and, again, the best strategy depends on the considered scenario. More precisely, we can distinguish three possible approaches for error correction in an information reconciliation protocol: *interactive*, *systematic encoding* and *hashing*. In the following we formalize their construction for direct reconciliation protocols.

Interactive approach

According to this approach, $\mathbf{y}_{\mathrm{S}}$ is interactively reconciled with $\mathbf{x}_{\mathrm{S}}$ by means of multiple, subsequent public communications.

To the best of the author's knowledge, the only examples of natively[5] interactive information reconciliation are the ones proposed by Bennett and Brassard, known respectively as *Binary* and *Cascade*, which are both described in the seminal paper by Brassard and Salvail [96] dated 1993. The two protocols share a common structure, which is described in the following:

[5] some protocols use interactivity in order to enhance their performance, but error correction is not intrinsically interactive.

$\forall$ pair of blocks $(\mathbf{x}_S^{(i)}, \mathbf{y}_S^{(i)})$, compute the reconciled block $\hat{\mathbf{x}}_S^{(i)}$ at the receiver according to the following algorithm:

1. if $\bigoplus(\mathbf{x}_S^{(i)}) = \bigoplus(\mathbf{y}_S^{(i)}) \Rightarrow \hat{\mathbf{x}}_S^{(i)} = \mathbf{y}_S^{(i)}$
2. if $\bigoplus(\mathbf{x}_S^{(i)}) \neq \bigoplus(\mathbf{y}_S^{(i)}) \Rightarrow$ split $\bigoplus(\mathbf{x}_S^{(i)})$ and $\bigoplus(\mathbf{y}_S^{(i)})$ in two halves, i.e., $\mathbf{x}_S^{(i)} = [\mathbf{x}_S^{(i,1)}|\mathbf{x}_S^{(i,2)}]$, $\mathbf{y}_S^{(i)} = [\mathbf{y}_S^{(i,1)}|\mathbf{y}_S^{(i,2)}]$:
   (a) if $\bigoplus(\mathbf{x}_S^{(i,1)}) = \bigoplus(\mathbf{y}_S^{(i,1)})$, set $\hat{\mathbf{x}}_S^{(i,1)} = \mathbf{y}_S^{(i,1)}$ and fed $\mathbf{x}_S^{(i,2)}$ and $\mathbf{y}_S^{(i,2)}$ as input to step (2a).
   (b) if $\bigoplus(\mathbf{x}_S^{(i,1)}) \neq \bigoplus(\mathbf{y}_S^{(i,1)})$, fed $\mathbf{x}_S^{(i,1)}$ and $\mathbf{y}_S^{(i,1)}$ as input to step (2a) and set $\hat{\mathbf{x}}_S^{(i,2)} = \mathbf{y}_S^{(i,2)}$.

The *Binary* protocol actually consists only of the algorithm described above, while the *Cascade* protocol, in addition, keeps track of block parities in subsequent iterations, so that when a new error is corrected, another one can be found in blocks of previous iterations with even parity in which the corresponding bit was located and so on.

The main advantage of *Binary* and *Cascade* is that of being natively usable with any input bit error rate, since the procedure does not need to know the bit error rate in advance, except for the optimal choice of block sizes. Its major drawback, on the other hand, lies in its interactivity: given a block with odd parity and length b, it takes $\log_2(b)$ interactions to find the (single) error to correct.

Systematic encoding approach

The systematic encoding approach leverages classical channel codes for correcting the errors that the quantum and the classical channel may have introduced, and, hence, it could be convenient for those scenarios where also the classical channel is prone to errors.[6]

Let us now describe this approach more formally. Consider a $(b+p, b)$ code $\mathcal{C}$ with generator matrix $\mathbf{G} \in \{0,1\}^{(b+p)\times b}$; we assume $\mathbf{G}$ to be in systematic form, i.e., $\mathbf{G} = [\mathbf{I}_b|\mathbf{P}]^T$, being $\mathbf{I}_b$ the identity matrix of size b and $\mathbf{P} \in \{0,1\}^{p\times b}$. Now, $\mathbf{x}_S$ is considered as a sequence of b-bit *information words*, $\mathbf{x}_S^{(i)}$, to be fed as input to an $(b+p, b)$ encoder for error correction; $\hat{\mathbf{x}}_S$ is then obtained as the concatenation of the output of the decoding of $\mathbf{y}_S^{(i)}$ with side information granted by channel coding. More formally:

[6] this could be the case if the classical channel model is at the physical or at the data-link layer.

$\forall$ $(\mathbf{x}_{\mathrm{S}}^{(i)}, \mathbf{y}_{\mathrm{S}}^{(i)})$, compute the reconciled block $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$ at the receiver according to the following algorithm:

1. Alice computes the i-th parity sequence as

$$\mathbf{c}^{(i)} = \mathbf{P}\mathbf{x}_{\mathrm{S}}^{(i)}. \tag{3.56}$$

2. Alice sends $\mathbf{c}^{(i)}$ to Bob over the classical channel.

3. Given the received parity bits $\mathbf{d}^{(i)}$ (i.e., the possibly corrupted version of $\mathbf{c}^{(i)}$), Bob concatenates them with $\mathbf{y}_{\mathrm{S}}^{(i)}$ and decodes the received sequence according to a decoding map μ_d[a] so that

$$[\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}|\hat{\mathbf{d}}^{(i)}] = \mu_d([\mathbf{y}_{\mathrm{S}}^{(i)}|\mathbf{d}^{(i)}]). \tag{3.57}$$

4. Bob outputs $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$.

[a] the choice of such decoding map is discussed at the end of this section.

The overall block diagram is depicted in figure 3.5.

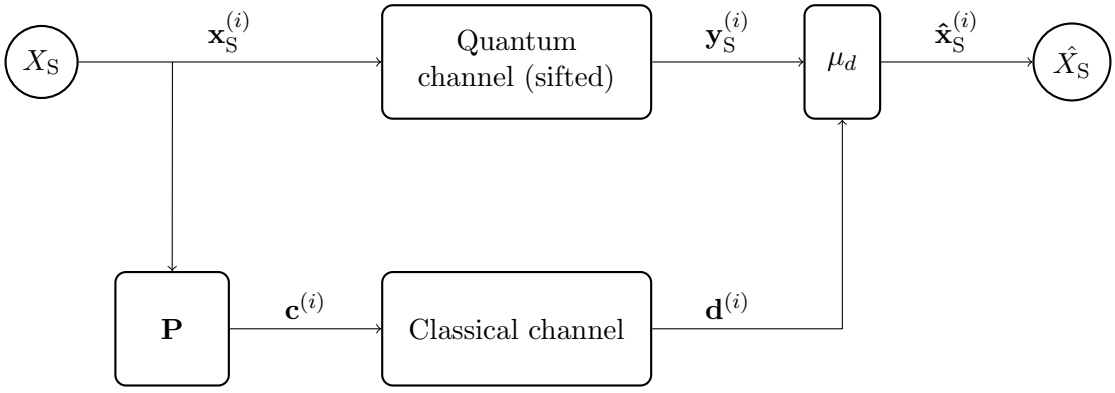


Figura 3.5: Direct reconciliation, systematic encoding scheme.

The choice of the decoding map depends on the model for the classical channel that we consider [103]. More specifically, if we access the channel at the physical layer, we could take advantage of the available bit reliability values for performing a soft-decoding (e.g., *maximum-likelihood*), whereas if we consider a channel at the data-link layer we are forced to choose a hard-decoding algorithm, e.g., minimum distance.

Examples of information reconciliation protocols using the systematic encoding approach can be found in [104], where the adoption of LDPC codes is proposed, and in [105], where BCH codes are preferred.

Hashing approach

The hashing approach takes advantage of channel codes, but, as compared with the systematic encoding one, it uses them in a different fashion.

First, we give an intuitive description of such scheme. More precisely, $\mathbf{x}_{\mathrm{S}}$ is considered as a sequence of b-ples, $\mathbf{x}_{\mathrm{S}}^{(i)}$, for each of which, by means of a hashing function, a hash $\mathbf{c}^{(i)}$ is computed. The hash is sent over the public channel, which is now supposed to be error-free,[7] and is jointly used with the received bits in order to get $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$; in particular, each reconciled b-ple is chosen as the one with hash $\mathbf{c}^{(i)}$ and with minimum Hamming distance from the $\mathbf{y}_{\mathrm{S}}^{(i)}$. Incidentally, the hashing function can be chosen as the parity check matrix of a linear block $(b, b-p)$ code, with parity-check matrix $\mathbf{H} \in \{0,1\}^{p \times b}$. The reconciled key $\hat{\mathbf{x}}_{\mathrm{S}}$ is then obtained as the concatenation of subsequent $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$.

Let us now define this approach more formally.

$\forall$ $(\mathbf{x}_{\mathrm{S}}^{(i)}, \mathbf{y}_{\mathrm{S}}^{(i)})$, compute the reconciled block $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$ at the receiver according to the following algorithm:

1. Alice computes the i-th syndrome (i.e., the i-th hash) as
$$\mathbf{c}^{(i)} = \mathbf{H}\mathbf{x}_{\mathrm{S}}^{(i)}. \tag{3.58}$$
2. Alice sends $\mathbf{c}^{(i)}$ to Bob over the classical channel, assumed to be error-free.
3. Bob performs a *minimum-distance* decoding of the received sequence so that
$$\hat{\mathbf{x}}_{\mathrm{S}}^{(i)} = \mu_d^*(\mathbf{y}_{\mathrm{S}}^{(i)}, \mathbf{c}^{(i)}) = \underset{\alpha:\mathbf{H}\alpha=\mathbf{c}^{(i)}}{\arg\min} \{d_{\mathrm{H}}(\alpha, \mathbf{y}_{\mathrm{S}}^{(i)})\}. \tag{3.59}$$
4. Bob outputs $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$.

In figure 3.6 the block diagram for this approach is depicted, where μ_d^* represents the decoding procedure described by Eq. (3.59).

To the best of the author's knowledge, the first information reconciliation protocol to use this approach was Winnow [97], a protocol based on Hamming codes. Another significant, recent example is that of [106, 107], where the authors introduce rate-adaptive LDPC codes for Quantum Key Distribution.

Remarks

The interactive approach takes advantage of its flexibility at the price of a very high interactivity. It exhibits an unequalled efficiency in scenarios where the bit error rate in the strings to be reconciled rapidly changes (e.g., in quantum key distribution), but it suffers a high latency due to interactive communications.

[7] if the public channel introduces errors, in fact, the decoding procedure is likely to introduce further errors, as the whole procedure works given that the received hash is correct.

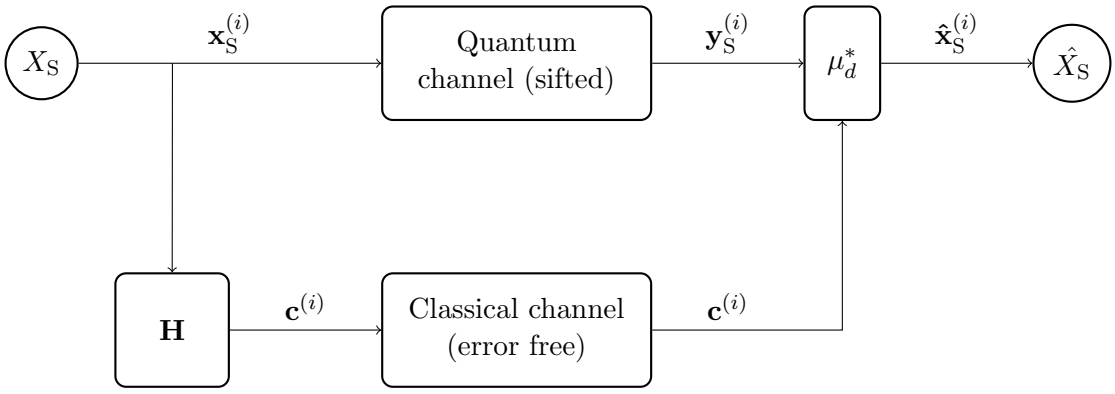


Figura 3.6: Direct reconciliation, hashing scheme.

The systematic encoding approach and the hashing approach try to overcome this severe limit through the use of channel codes; while looking in this direction, the main problem is that of rate-adaptability in case the protocol has to be applied to a channel with dynamic characteristics. Significant improvements towards this direction were pursued in [106, 107].
Even though the systematic encoding approach and the hashing approach present a similar structure, a significant difference has to be stressed: in the systematic encoding approach redundancy is used to correct errors both in the quantum channel *and* in the public channel, while, in the hashing approach, it is committed to correct errors only in the quantum channel. As a consequence, the systematic encoding approach is better suited to scenarios where the public channel is affected by non-negligible transmission errors (physical and data-link layers), while the hashing approach fits environments where the public channel is assumed to be error-free (network, transport or upper layers). It should be noted, however, that the hashing approach is more efficient in terms of information disclosed on the public channel and is thus generally preferable in QKD scenarios, as the correctness of the information sent on the public channel can be ensured without affecting the final secret key rate.

In table 3.5 we show some examples of how the choice of the coding approach depends on the considered classical channel model.

Classical channel (layer)	Channel type	Condition	Delays	Coding approach
Physical	AWGN	high SNR	none	systematic (soft)
Data link	binary	low BER	low	systematic (hard)
Net & up	packet	error free	long	interactive, hashing

Tabella 3.5: Examples of coding approaches for different classical channel models.

3.5.2 Analysis of the Winnow protocol

The *Winnow* protocol [97] mixes the *hashing* and the *interactive* approach in order to build a flexible, still lowly interactive solution. Winnow is based on two fundamental ingredients, Hamming codes and syndrome decoding, and it leverages these tools over multiple iterations in order to ensure the correctness constraint (3.2).

In this subsection, we first describe the protocol and we then propose its analysis in a way that we think is more intuitive with respect to the one proposed in [97].

Winnow error correction algorithm.

Given two n-bit sifted sequences $\mathbf{x}_\mathrm{S}$ and $\mathbf{y}_\mathrm{S}$, let us define the block size $b = 2^p$, for any arbitrary p. Furthermore, we define with $\mathcal{H}(p)$ the $(2^p - 1, 2^p - 1 - p)$ Hamming code and we denote its parity check matrix by $\mathbf{H}$. A single iteration of the Winnow error correction algorithm works as follows:

1. split $\mathbf{x}_\mathrm{S}$ and $\mathbf{y}_\mathrm{S}$ in $B = \lceil n/b \rceil$ blocks of b bits; let us denote with $\mathbf{x}_\mathrm{S}^{(i)}$ and $\mathbf{y}_\mathrm{S}^{(i)}$ the pair of the i-th blocks.

2. $\forall$ $(\mathbf{x}_\mathrm{S}^{(i)}, \mathbf{y}_\mathrm{S}^{(i)})$, compute the corrected block $\hat{\mathbf{x}}_\mathrm{S}^{(i)}$ at the receiver according to the following algorithm:

   (a) if $\bigoplus(\mathbf{x}_\mathrm{S}^{(i)}) = \bigoplus(\mathbf{y}_\mathrm{S}^{(i)}) \Rightarrow \hat{\mathbf{x}}_\mathrm{S}^{(i)} = \mathbf{y}_\mathrm{S}^{(i)}$

   (b) if $\bigoplus(\mathbf{x}_\mathrm{S}^{(i)}) \neq \bigoplus(\mathbf{y}_\mathrm{S}^{(i)})$:

      i. define the $(b-1)$-bit strings obtained by removing the last bit from $\mathbf{x}_\mathrm{S}^{(i)}$ and $\mathbf{y}_\mathrm{S}^{(i)}$:

$$\mathbf{p}_\mathrm{S}^{(i)} \triangleq [\mathbf{x}_\mathrm{S}^{(i)}(1), \mathbf{x}_\mathrm{S}^{(i)}(2), \ldots, \mathbf{x}_\mathrm{S}^{(i)}(b-1)] \tag{3.60}$$
$$\mathbf{q}_\mathrm{S}^{(i)} \triangleq [\mathbf{y}_\mathrm{S}^{(i)}(1), \mathbf{y}_\mathrm{S}^{(i)}(2), \ldots, \mathbf{y}_\mathrm{S}^{(i)}(b-1)] \tag{3.61}$$

      ii. compute the syndromes

$$s(\mathbf{p}_\mathrm{S}^{(i)}) = \mathbf{H}\mathbf{p}_\mathrm{S}^{(i)} \tag{3.62}$$
$$s(\mathbf{q}_\mathrm{S}^{(i)}) = \mathbf{H}\mathbf{q}_\mathrm{S}^{(i)} \tag{3.63}$$

      and their difference

$$s_\mathrm{d}^{(i)} = s(\mathbf{p}_\mathrm{S}^{(i)}) - s(\mathbf{q}_\mathrm{S}^{(i)}) \tag{3.64}$$

      A. if the computed syndromes match ($s_\mathrm{d}^{(i)} = \mathbf{0}$):

$$\hat{\mathbf{p}}_\mathrm{S}^{(i)} = \mathbf{q}_\mathrm{S}^{(i)}, \tag{3.65}$$
$$\hat{\mathbf{x}}_\mathrm{S}^{(i)} = [\hat{\mathbf{p}}_\mathrm{S}^{(i)}, \mathbf{y}_\mathrm{S}^{(i)}(b) \oplus 1] \tag{3.66}$$

B. if the computed syndromes match ($s(\mathbf{p}_{\mathrm{S}}^{(i)}) \neq s(\mathbf{q}_{\mathrm{S}}^{(i)})$):

$$\hat{\mathbf{p}}_{\mathrm{S}}^{(i)} = \underset{\gamma:\mathbf{H}\gamma=s_{\mathrm{d}}^{(i)}}{\arg\min} \{d_{\mathrm{H}}(\gamma, \mathbf{q}_{\mathrm{S}}^{(i)})\} \tag{3.67}$$

$$\hat{\mathbf{x}}_{\mathrm{S}}^{(i)} = [\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}, \mathbf{y}_{\mathrm{S}}^{(i)}(b)] \tag{3.68}$$

3. output $\hat{\mathbf{x}}_{\mathrm{S}} = [\hat{\mathbf{x}}_{\mathrm{S}}^{(1)}, \ldots, \hat{\mathbf{x}}_{\mathrm{S}}^{(B)}]$.

The Winnow correction algorithm is typically used over multiple iterations with increasing block size b; Hamming codes, in fact, are capable of correcting just one error per block, and in order to make the residual error probability sufficiently lower it may be required to perform a few iterations. In particular, the block sizes, which are constrained to be powers of 2 by construction, should be chosen so that the event of having more than one error per block is unlikely.

For evaluating the performance of the Winnow protocol, we want to compute P_{out}, i.e., the residual bit error rate on the b-bit string after the execution of a single pass of the protocol, and L_{EC}, i.e., the number of bits disclosed in the process. Let us define the following random variables:

- number of errors in the b-bit string received by Bob, $\mathbf{y}_{\mathrm{S}}^{(i)}$:

$$N \triangleq \left(\sum_{j=1}^{b} \mathbf{x}_{\mathrm{S}}^{(i)}(j) \oplus \mathbf{y}_{\mathrm{S}}^{(i)}(j) \right) \tag{3.69}$$

- number of errors in the $(b-1)$-bit string received by Bob and deprived of the parity bit, $\mathbf{q}_{\mathrm{S}}^{(i)}$:

$$N' \triangleq \left(\sum_{j=1}^{b-1} \mathbf{p}_{\mathrm{S}}^{(i)}(j) \oplus \mathbf{q}_{\mathrm{S}}^{(i)}(j) \right) \tag{3.70}$$

- number of errors in $(b-1)$-bit string obtained from $\mathbf{q}_{\mathrm{S}}^{(i)}$ by means of syndrome decoding, $\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}$:

$$R' \triangleq \left(\sum_{j=1}^{b-1} \mathbf{p}_{\mathrm{S}}^{(i)}(j) \oplus \hat{\mathbf{p}}_{\mathrm{S}}^{(i)}(j) \right) \tag{3.71}$$

- number of errors in b-bit string obtained from $\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}$ by adding the (eventually flipped) parity bit, $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$:

$$R \triangleq \left(\sum_{j=1}^{b} \mathbf{x}_{\mathrm{S}}^{(i)}(j) \oplus \hat{\mathbf{x}}_{\mathrm{S}}^{(i)}(j) \right) \tag{3.72}$$

Residual bit error rate

The residual bit error rate can be computed from the joint statistical description of the random variables N, N' and R'. As it will be shown later on, R is in fact a deterministic

function of N, N' and R'. The joint probability distribution of these random variables can be expressed as

$$p_{R'N'N}(\beta,\alpha,\delta) = p_{R'|N'}(\beta|\alpha)p_{N'|N}(\alpha|\delta)p_N(\delta). \tag{3.73}$$

In the following, we derive an explicit expression for each contribution on the right-hand side of Eq. (3.73).

Finding $p_N(\delta)$. Assuming that the channel can be modeled as a BSC (Binary Symmetric Channel) with i.i.d errors and transition probability Q, $p_N(\delta)$ is a binomial distribution, that is

$$p_N(\delta) = \binom{b}{\delta} Q^\delta (1-Q)^{b-\delta}. \tag{3.74}$$

Finding $p_{N'|N}(\alpha|\delta)$. Let us start from considering that, given that there are δ errors in $\mathbf{x}_{\mathrm{S}}^{(i)}$, the number of errors α in $\mathbf{p}_{\mathrm{S}}^{(i)}$ is equal either to δ (the parity bit is correct) or to $\delta-1$ (the parity bit is not correct). Being the errors uniformly distributed and given that there are δ errors in $\mathbf{x}_{\mathrm{S}}^{(i)}$, the probability that the parity bit is not correct is equal to the probability that one of the δ errors is exactly on the parity bit, that is, δ/b. On the other hand, the probability that the parity bit is correct and that all the δ errors are in $\mathbf{p}_{\mathrm{S}}^{(i)}$ is equal to $1-\delta/b$. In conclusion, the probability distribution $p_{N'|N}(\alpha|\delta)$ can be expressed as

$$p_{N'|N}(\alpha|\delta) = \begin{cases} \frac{\delta}{b}, & \text{if } \alpha = \delta - 1, \\ 1 - \frac{\delta}{b} & \text{if } \alpha = \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{3.75}$$

Finding $p_{R'|N'}(\beta|\alpha)$. The last probability distribution requires a slightly more complex procedure in order to be derived. First of all, given an $(2^p-1, 2^p-p-1)$ Hamming code $\mathcal{H}$ with parity check matrix $\mathbf{H}$, let us define $C(\alpha)$ as the number of codewords with Hamming weight $w_{\mathrm{H}}(\alpha)$, that is

$$C(\alpha) = |\{\gamma \in \mathcal{C} : w_{\mathrm{H}}(\gamma) = \alpha\}|. \tag{3.76}$$

The following equality holds [108]:

$$\binom{b}{\alpha} = (\alpha+1)C(\alpha+1) + C(\alpha) + (b-\alpha+1)C(\alpha-1),$$

so that, given $C(\alpha)$ and $C(\alpha-1)$, we can compute $C(\alpha+1)$.

In addition to $C(\alpha)$, we define the function $D(\alpha,\beta)$, representing the number of words with weight α for which the closest Hamming codewords have weight β. We know that the Hamming decoding procedure may lead to the correction of a single error (i.e., $\beta = \alpha-1$), to the introduction of an additional error (i..e, $\beta = \alpha+1$) or eventually not even alter the

number of errors ($\beta = \alpha$). We are therefore interested in computing $D(\alpha, \beta)$ just for the pairs (α, β) for which $\beta = \alpha + \xi$, where $\xi \in \{-1, 0, 1\}$. We get:

$$\begin{cases} D(\alpha, \alpha) = C(\alpha) \\ D(\alpha, \alpha + 1) = (\alpha + 1)C(\alpha + 1) \\ D(\alpha, \alpha - 1) = [(b-1) - (\alpha - 1)]C(\alpha - 1) \end{cases} \tag{3.77}$$

Furthermore, given the syndrome of $\mathbf{p}_{\mathrm{S}}^{(i)}$, $s(\mathbf{p}_{\mathrm{S}}^{(i)}) = \mathbf{H}\mathbf{p}_{\mathrm{S}}^{(\mathbf{i})}$, the following equality holds:

$$\hat{\mathbf{p}}_{\mathrm{S}}^{(i)} = \underset{\gamma : \mathbf{H}\gamma = s(\mathbf{p}_{\mathrm{S}}^{(i)})}{\arg\min} \{d_{\mathrm{H}}(\gamma, \mathbf{q}_{\mathrm{S}}^{(i)})\} \tag{3.78}$$

Finally, let $\mathbf{e}$ and δ be the $(b-1)$-bit error words respectively before and after the Hamming decoding, i.e.,

$$\begin{cases} \mathbf{e} = \mathbf{q}_{\mathrm{S}}^{(i)} - \mathbf{p}_{\mathrm{S}}^{(i)} \\ \hat{\mathbf{e}} = \hat{\mathbf{p}}_{\mathrm{S}}^{(i)} - \mathbf{p}_{\mathrm{S}}^{(i)} \end{cases} \tag{3.79}$$

so that

$$\begin{cases} \alpha = w_{\mathrm{H}}(\mathbf{e}) = d_{\mathrm{H}}(\mathbf{q}_{\mathrm{S}}^{(i)}, \mathbf{p}_{\mathrm{S}}^{(i)}) \\ \beta = w_{\mathrm{H}}(\hat{\mathbf{e}}) = d_{\mathrm{H}}(\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}, \mathbf{p}_{\mathrm{S}}^{(i)}) \end{cases} \tag{3.80}$$

We remark that, according to Eq.(3.78), $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}$ have the same syndrome, and $\hat{\mathbf{e}}$ then turns out to be a code word.[8] Then, by using Eq. (3.79), we can rewrite Eq.(3.78) as

$$\hat{\mathbf{p}}_{\mathrm{S}}^{(i)} = \mathbf{p}_{\mathrm{S}}^{(i)} + \hat{\mathbf{e}} = \mathbf{p}_{\mathrm{S}}^{(i)} + \underset{\hat{\mathbf{e}} \in \mathcal{H}(p)}{\arg\min} \{d_{\mathrm{H}}(\hat{\mathbf{e}}, \mathbf{e})\}. \tag{3.81}$$

Furthermore, being α the number of errors in $\mathbf{q}_{\mathrm{S}}^{(i)}$ and β the number of errors in $\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}$, the transition probability from α to β errors after the Hamming decoding is equal to

$$p_{R'|N'}(\beta|\alpha) = \frac{D(\alpha, \beta)}{\binom{b-1}{\alpha}}. \tag{3.82}$$

Expressing R as a function R', N' and N. Once the joint probability distribution of R', N' and N is obtained, we still have to find how R is dependent on the realizations of such random variables. First, let us consider that if δ is odd, this relationship depends on the parity bit handling; according to the protocol described in the first part of this section, the parity bit is reinserted unaltered if the syndrome difference relative to $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\mathbf{q}_{\mathrm{S}}^{(i)}$ is not equal to the zero string, and is flipped otherwise. Summarizing, when δ is odd we can distinguish four different cases:

[8] we remind that, given a code $\mathcal{C}$, a word belongs to $\mathcal{C}$ if its syndrome is the zero string. In our setting, where $\mathcal{H}(p)$ is the $(2^p - 1, 2^p - p - 1)$ Hamming code with parity check matrix $\mathbf{H}$, we have $\mathbf{H}\hat{\mathbf{e}} = \mathbf{H}(\hat{\mathbf{p}}_{\mathrm{S}}^{(i)} - \mathbf{p}_{\mathrm{S}}^{(i)}) = \mathbf{H}\hat{\mathbf{p}}_{\mathrm{S}}^{(i)} - \mathbf{H}\mathbf{p}_{\mathrm{S}}^{(i)} = s(\hat{\mathbf{p}}_{\mathrm{S}}^{(i)}) - s(\mathbf{p}_{\mathrm{S}}^{(i)}) = \mathbf{0}$; consequently $\hat{\mathbf{e}} \in \mathcal{H}(2^p - 1, 2^p - p - 1)$.

1. $\beta = \alpha$, $\alpha = \delta$: the parity bit is correct and the syndromes of $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\mathbf{q}_{\mathrm{S}}^{(i)}$ coincide; the parity bit is flipped before being reinserted (thus introducing a new error).

2. $\beta = \alpha$, $\alpha \neq \delta$: the parity bit is not correct and the syndromes of $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\mathbf{q}_{\mathrm{S}}^{(i)}$ coincide; the parity bit is flipped before being reinserted (and therefore it does not increase the number of errors).

3. $\beta \neq \alpha$, $\alpha = \delta$: the parity bit is correct and the syndromes of $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\mathbf{q}_{\mathrm{S}}^{(i)}$ are not identical; the parity bit is reinserted without modifications (and therefore it does not increase the number of errors).

4. $\beta \neq \alpha$, $\alpha \neq \delta$: the parity bit is not correct and the syndromes of $\mathbf{p}_{\mathrm{S}}^{(i)}$ and $\mathbf{q}_{\mathrm{S}}^{(i)}$ are not identical; the parity bit is reinserted without modifications (thus introducing a new error).

On the other hand, if δ is even, the number of residual errors is equal to δ, since no correction is performed.

In conclusion, R is a deterministic function of β, α and δ such that

$$R = f_{R'N'N}(\beta,\alpha,\delta) = \begin{cases} \delta, & \text{if } \delta \text{ is even} \\ \beta+1, & \text{if } \delta \text{ is odd } \wedge \{(\delta=\alpha)\wedge(\alpha=\beta)\} \vee \{(\delta\neq\alpha)\wedge(\alpha\neq\beta)\} \\ \beta, & \text{otherwise} \end{cases} \tag{3.83}$$

Final computation. Finally, we can express P_{out} as

$$P_{out} = \frac{E[R]}{n}. \tag{3.84}$$

The expectation of R can be written as

$$\begin{aligned} E[R] &= \sum_{\delta=0}^{b}\sum_{\alpha=\alpha_1}^{\alpha_2}\sum_{\beta=\beta_1}^{\beta_2} f_{R'N'N}(\beta,\alpha,\delta)\, p_{R'N'N}(\beta,\alpha,\delta) && (3.85)\\ &= \sum_{\delta=0}^{b}\sum_{\alpha=\alpha_1}^{\alpha_2}\sum_{\beta=\beta_1}^{\beta_2} f_{R'N'N}(\beta,\alpha,\delta)\, p_{R'|N'}(\beta|\alpha)\, p_{N'|N}(\alpha|\delta)\, p_N(\delta) && (3.86)\\ &= \sum_{\delta=0}^{b}\sum_{\alpha=\alpha_1}^{\alpha_2}\sum_{\beta=\beta_1}^{\beta_2} f_{R'N'N}(\beta,\alpha,\delta)\frac{D(\alpha,\beta)}{\binom{b-1}{\alpha}}\cdot p_{N'N}(\alpha,\delta)\cdot\binom{b}{\delta}\epsilon^{\delta}(1-\epsilon)^{b-\delta}, && (3.87) \end{aligned}$$

where

$$\begin{aligned} \alpha_1 &= \max\{0,\delta-1\} \qquad \beta_1 = \max\{0,\alpha-1\} \\ \alpha_2 &= \min\{b-1,\delta\} \qquad \beta_2 = \min\{b-1,\alpha+1\}. \end{aligned}$$

Expectation of the fraction of disclosed bits

The procedure for computing the expectation of the fraction of bits disclosed by Winnow is by far less complicated than the one required for the residual bit error rate. More specifically, we just consider the following facts:

1. 1 bit is disclosed for each block by the protocol in order to perform the parity check;
2. $p = \lceil log_2(n) \rceil$ bits are revealed for blocks with mismatching parities in order to perform Hamming decoding.

We therefore define the probability $P_{blk,odd}$ that an odd number of errors occurs in a block of b bits. Assuming that the quantum channel can be modelled as binary symmetric with transition probability Q, we get:

$$P_{blk,odd} = \sum_{i \in \mathcal{I}} \binom{b}{i} Q^i (1-Q)^{b-i}, \tag{3.88}$$

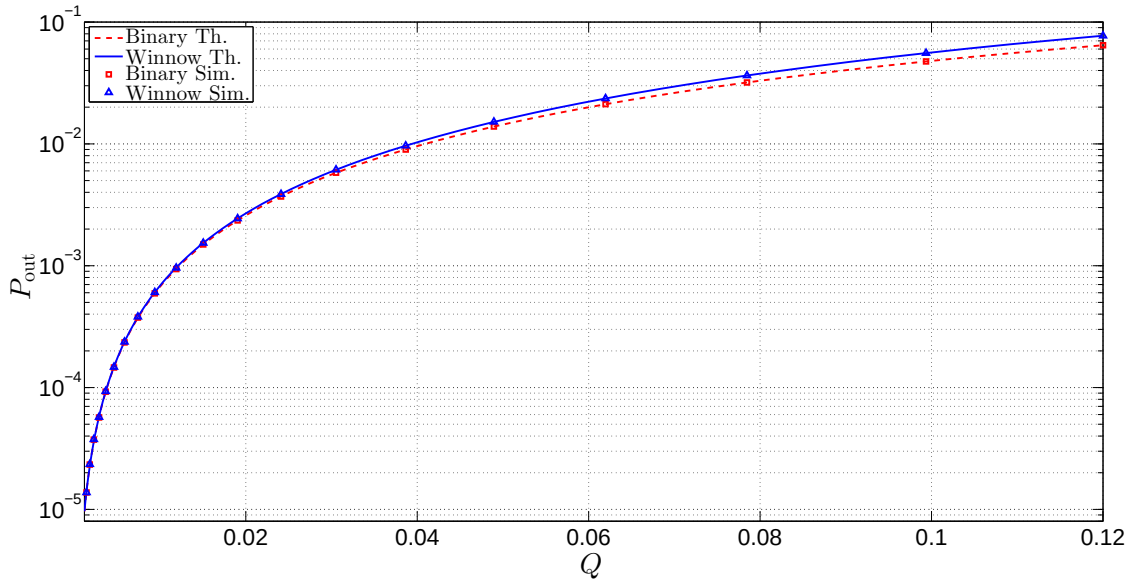
being $\mathcal{I} = \{i \in [1, b], i \text{ odd}\}$, and the error correction leakage per block can be written as

$$L_{\text{EC,blk}} = 1 + p \cdot P_{blk,odd} = 1 + p \cdot \sum_{i \in \mathcal{I}} \binom{b}{i} Q^i (1-Q)^{b-i}. \tag{3.89}$$

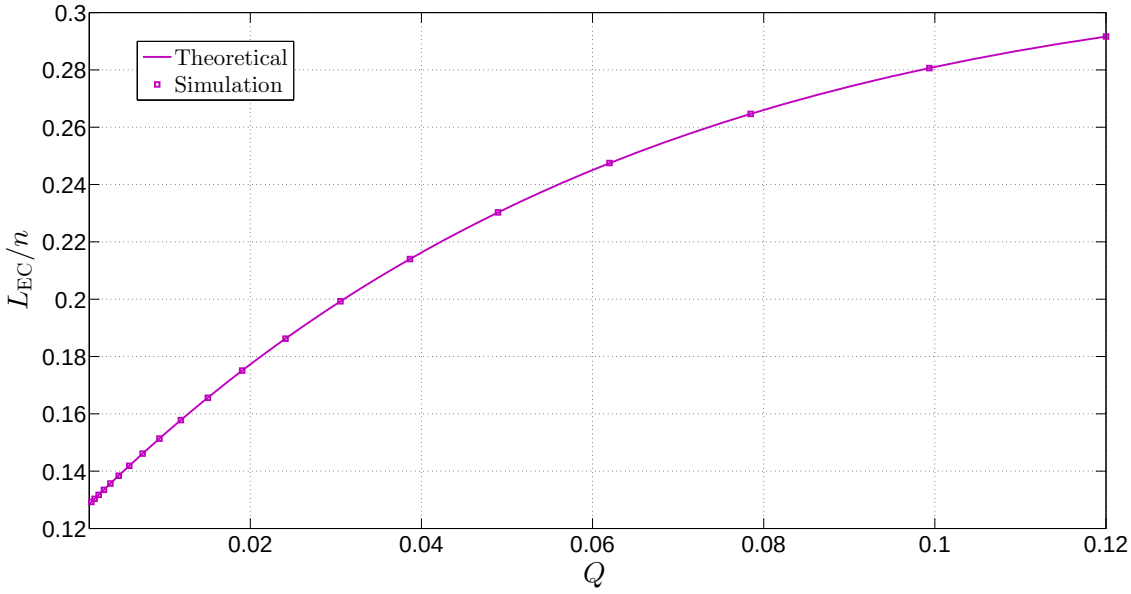
Theoretical analysis and simulations

In this section we evaluate the performance of the Winnow scheme according to the theoretical analysis and to simulations results. Also, we compare them with the error correction mechanism leveraged by the *Cascade* algorithm, that is, the binary error search (*Binary*). The cascading principle, in fact, can be applied also to Winnow, thus yielding an improvement in the reconciliation efficiency but also a higher interactivity. Since the analysis would be by far more involved, we therefore decided to compare the core correction mechanisms adopted in these two protocols.

Figure 3.7a shows the residual bit error rate, P_{out}, after a single iteration of the Binary and of the Winnow protocol, respectively, as a function of the quantum bit error rate Q and with block size equal to 8 bits; both simulation results and theoretical values are reported and, as it can be seen, they perfectly match. Simulations were performed with 1000 random, independent realizations of 320000-bit sequences. It can be noticed that the plots of the two protocols are very similar, though Binary performs slightly better, with an advantage which becomes more appreciable for higher values of Q. This is due to the fact that the Binary protocol always corrects one error per block, whereas Hamming codes may even introduce errors when a block exhibits an odd number of errors greater than 2. On the other hand, if we consider the associated fraction of disclosed bits (figure 3.7b), we see that in a single iteration the two protocols disclose exactly the same number of bits: given a block of length b, in fact, both Binary and Winnow disclose $\lceil \log_2(b) \rceil + 1$ bits. It should be stressed, however, that this correspondence does not hold if multiple iterations are performed, as shown in the following.



(a) Residual bit error rate

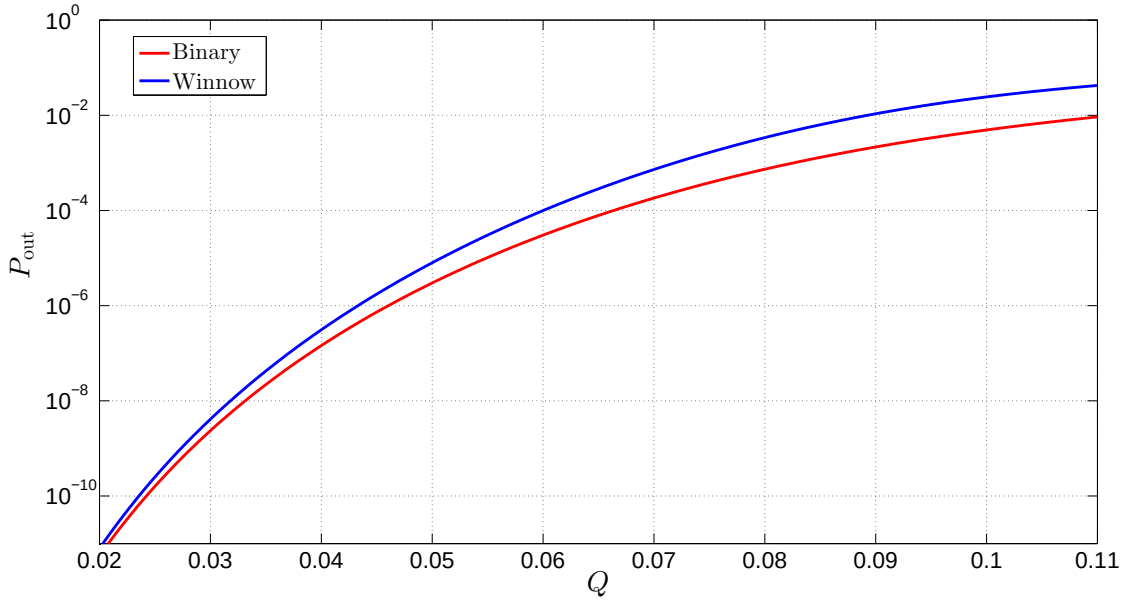


(b) Fraction of disclosed bits

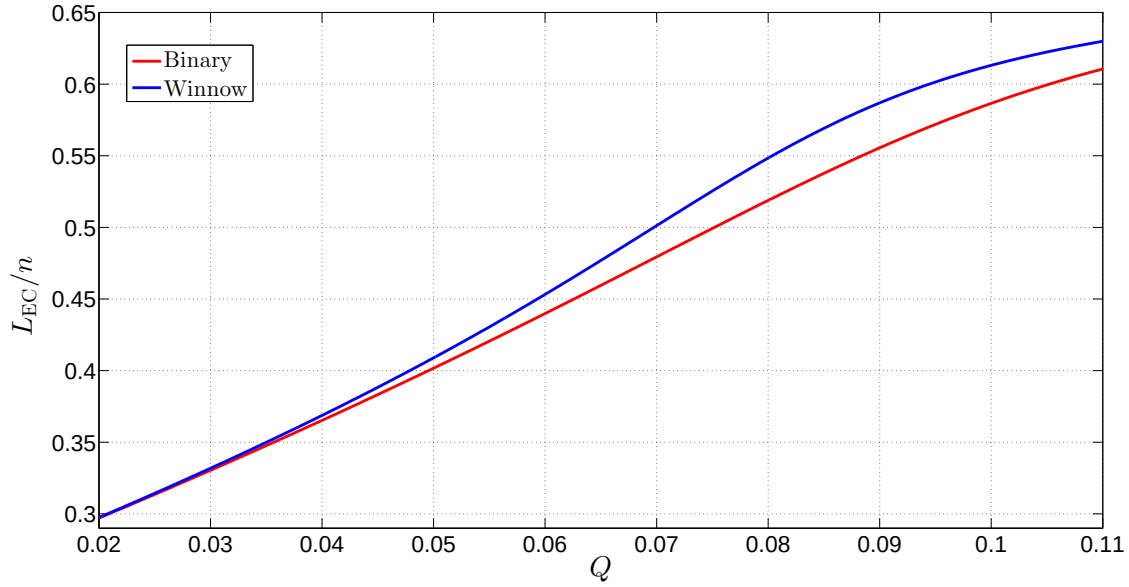
Figura 3.7: Comparison of a single iteration of Binary and Winnow with block size equal to 8 bits.

Let us now consider (figures 3.8a and 3.8b) the case of 4 protocol iterations with block sizes equal to $\mathbf{L} = [8, 16, 32, 64]$ bits, respectively. We note that Binary and Winnow have approximately the same performance for low QBER values, but then Binary outperforms Winnow; in the considered interval, however, the difference never exceeds a factor of approximately $3 \cdot 10^{-2}$ for both P_{out} and L_{EC}/n. We also point out that for moderate QBERs, both Binary and Winnow achieve excellent correction capability; more precisely, it can be shown that the specific $\mathbf{L}$ we here consider ensures a residual bit error rate below 10^{-8} for $Q = 3.1 \cdot 10^{-2}$ (see next paragraph on the block size optimization).

The observed advantage of the Binary protocol over Winnow in terms of both P_{out} and L_{EC} can be explained by considering how the Hamming decoding algorithm works.



(a) Residual bit error rate



(b) Fraction of disclosed bits

Figura 3.8: Comparison of Binary and Winnow with 4 iterations and block sizes $\mathbf{L} = [8, 16, 2, 64]$.

We recall, in fact, that by definition a Hamming code is able to correct 1 error and detect 2 errors, but if more errors are present then the outcome depends on the specific word which is fed as input to the algorithm. More specifically, we recall a few significant cases:

1. if Hamming decoding is applied to a word with 1 error, then the error will be corrected;

2. if Hamming decoding is applied to a word with 2 error, then the decoded word will end up with exactly 3 errors;

3. if Hamming decoding is applied to a word with 3 error, then the decoded word could have either 3 or 4 errors, depending on the specific pattern.

Hence, if there is more than 1 error per block, then Winnow may even introduce additional errors, thus worsening the residual bit error rate at the end of the protocol iterations. Furthermore, the newly introduced errors propagate in subsequent iterations, thus leading to an overall performance degrading.

On the other hand, the Winnow protocol exhibits a significant advantage in terms of interactions between the transmitter and the receiver. Even if the messages of the Binary protocol are sent in parallel for multiple blocks, in fact, they cannot compare to the single message sent from Alice to Bob for a single Winnow protocol iteration. In figure 3.9 we show the expected number of interactions for both Cascade and Winnow for three different reconciliation block sizes L_i and for different QBERs. As it can be seen, Winnow converges to 1 interaction, whereas the Cascade protocol rapidly reaches $\log_2(L_i)$ interactions per block.

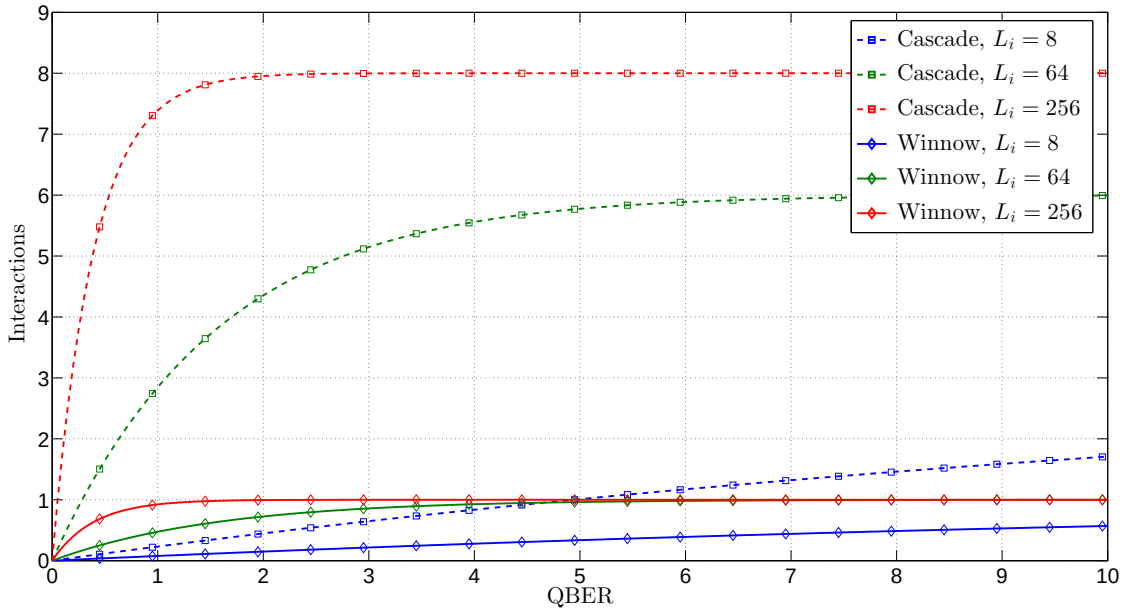


Figura 3.9: Expected number of interactions for Cascade and Winnow depending on the reconciliation block size.

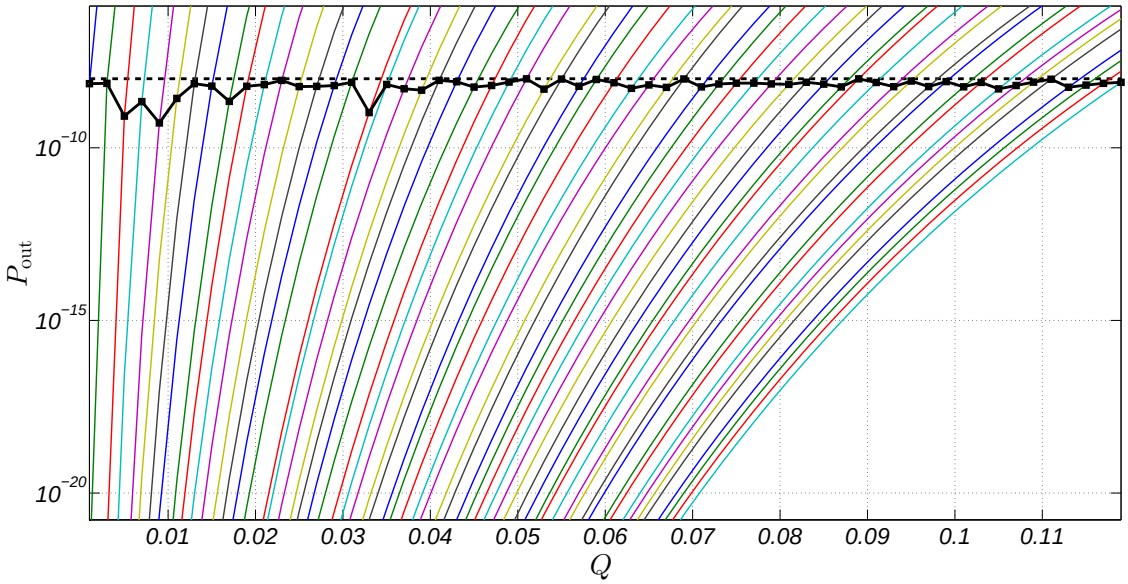
Block size optimization.

Given the maximum number of iterations, $I_{\max}$, the maximum block size, $L_{\max}$, and the target bit error rate at the output, T_{out}, we have investigated the optimal sequence of block sizes in order to minimize the fraction of disclosed bits while ensuring the target bit error rate. We recall, in fact, that the block size is a crucial parameter in each iteration of an information reconciliation protocol. In particular, both Binary and Winnow are optimal[9] if only 1 error is present in a block, since they can correct at most 1 error per

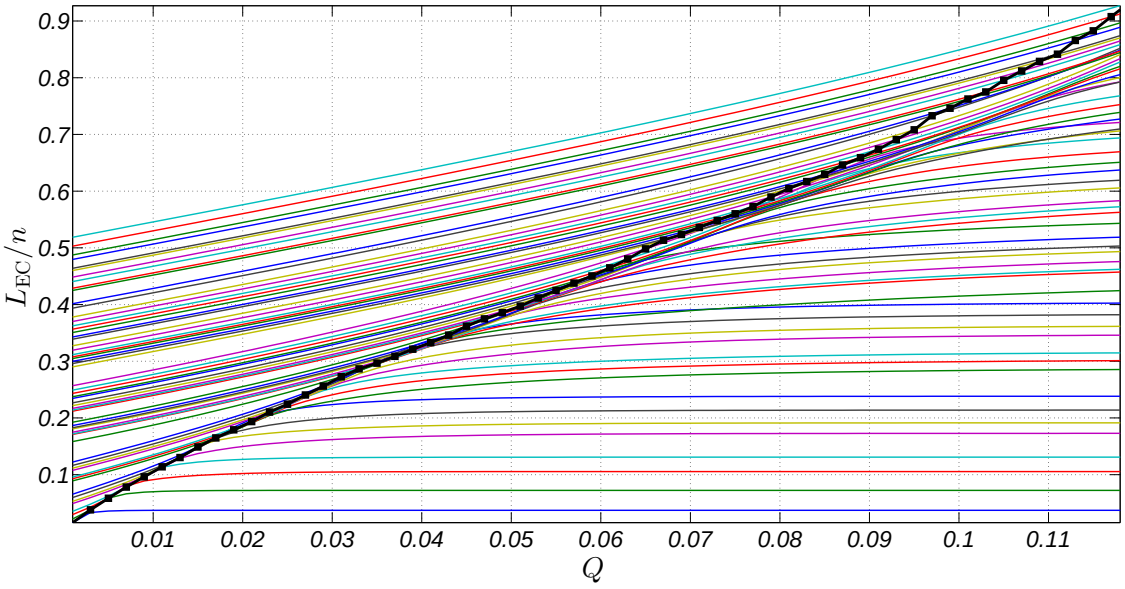
[9] in this context, "optimal" means that they correct all errors.

iteration. Furthermore, the preliminary parity check does not reveal an even number of errors, which therefore remain undetected, eventually up to the next protocol iteration.

In figures 3.10a and 3.10b, the thin lines represent the plots relative to different block sizes combinations and the thick one corresponds to the optimal choice with respect to the fraction of disclosed bits. The target bit error rate is set to $T_{\text{out}} = 10^{-8}$ (dashed line in figure 3.10a), with $I_{\max} = 6$ and $L_{\max} = 1024$ bits. As we can see in figure 3.10a, the optimal choice of the block sizes rapidly changes with the QBER, so that the target bit error rate condition is fulfilled. Figure 3.10b shows that the optimal (i.e., minimum) fraction of disclosed bits grows with the QBER almost linearly.



(a) Residual BER



(b) fraction of disclosed bits

Figura 3.10: Winnow block size optimization with $I_{\max} = 4$, $L_{\max} = 512$ bits and $T_{\text{out}} = 10^{-8}$.

Moreover, it is interesting to consider how the reconciliation efficiency changes for

different values of the target bit error rate and as a function of the QBER. Figure 3.11 shows this plot for the Winnow protocol. As expected, we see that lower values of T_{out} yields better reconciliation efficiencies η_{EC}, since the requirement on the residual bit error rate are less stringent. Also, we note that the gap between the three curves increases with Q. For each T_{out}, in fact, we expect that there exists a maximum Q_{max} that can be corrected by the Winnow while ensuring the output bit error rate requirement: the closer is Q to Q_{max}, the higher will be η_{EC}. Furthermore, we observe that the curves are not smooth; this can be explained by looking at figures 3.10a and 3.10b: optimal block lengths, in fact, do not change gracefully but are sudden, since each optimal combination is kept until the bound on the target bit error rate is satisfied, and, as soon as it is no longer verified, the optimal block size suddenly jumps to a new combination.

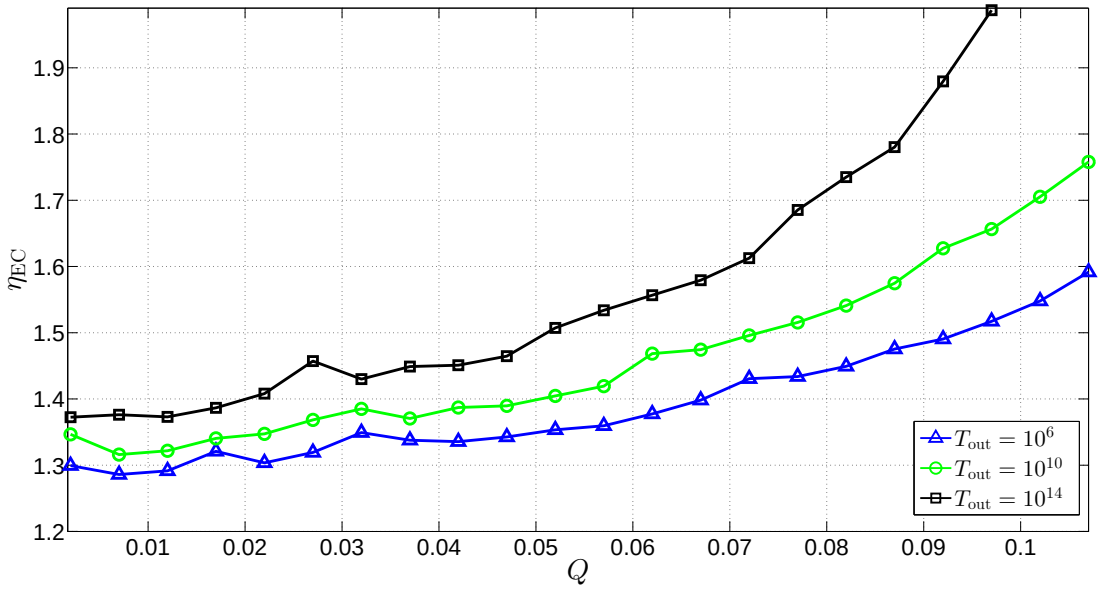


Figura 3.11: Winnow reconciliation efficiency for different target bit error rates, with $I_{\text{max}} = 6$ and $L_{\text{max}} = 1024$ bits.

3.5.3 LDPC codes for information reconciliation

In the past few years, LDPC codes [109] have gained an increasing interest in the scientific community dealing with information reconciliation for quantum key distribution. LDPC codes are in fact *capacity-approaching* linear codes which allow for highly efficient decoding even for large block sizes and at present they are widely deployed in classical systems. In particular, their efficiency is attractive while trying to design a *non-interactive* information reconciliation protocol which allows to increase the secret key rate and/or the maximum transmission distance.

Early works on LDPC codes for information reconciliation applications in QKD appeared starting from 2004 [110, 111, 112], but the proposed systems had to cope with some crucial issues [113], mainly the design of an optimized generator matrix, and the code rate adaptation by puncturing and shortening algorithms. The problem of *rate adaptability*, in

fact, is critical in the setting of quantum key distribution due to the intrinsic unsteadiness of the quantum channel.[10]

Despite these foreseen obstacles, in the last few years several works improved the feasibility of LDPC error correction as applied to QKD. In [98] the authors propose a solution for optimizing LDPC codes for binary symmetric channels. More specifically, an evolutionary optimization algorithm, called *Density Evolution* [114], is used and decoding is performed by means of belief propagation. A discrete family of codes is designed for different values of the transition probability (i.e., the quantum bit error rate), with a block length of 10^6 and so that a residual bit error rate below $1.5 \cdot 10^{-6}$ is achieved. Unfortunately, since the number of codes is limited, the reconciliation efficiency η_{EC} exhibits a saw behaviour [98, Fig.1]; as the number of codes increases the saw shaping is gradually smoothed. The authors of [98] propose to use *local randomization* in order to selectively worsen the error rate on the strings to be reconciled and thus make the designed codes work in their optimal region. The proposed solution outperforms Cascade for transition probabilities greater than 2%, getting an advantage up to nearly 20% as the transition probability gets close to 10%. Furthermore, the maximal admissible transition probability threshold is raised up to nearly 11% (i.e., the theoretical limit which allows a reconciled string to be extracted), whereas the Cascade protocol does not exceed 9.5%.

A further step has been made by the same authors in [106, 107], where information rate adaptability is investigated, and a solution for rate modulation based on puncturing and shortening is proposed. With respect to [98], this solution allows for a smoother reconciliation efficiency and for a much lower implementation complexity, as the proposed protocol adjusts the rate of pre-built LDPC codes. Conversely, the LDPC codes without shortening and puncturing proposed in [98] behave slightly better near their threshold, but this disadvantage is definitely negligible in the overall performance assessment and the efficiencies of the two solutions nearly coincide. As a conclusion to this paragraph, let us underline that the approaches described in [106, 98, 107] fall within the *hashing approach* formalized in §3.5.1.

Finally, for the sake of completeness, we point out that there exist also alternative proposals based on a *systematic encoding* approach. In particular, the authors of [115, 104] assume a composite channel model. On the quantum channel, which is regarded as a "hard-output" binary symmetric channel, the information bits are transmitted, while the classical channel, which is considered as a "soft-output" AWGN channel, is the transmission medium for redundancy bits. At the receiver, the metrics derived from the two channels are jointly processed according to a specific weighting function and decoding is performed by means of belief propagation. The authors claim that, given an LDPC code with code rate 0.5, decoded with 100 iterations and with optimal metrics weight, their solution ensures a residual bit error rate in the order of 10^{-6} even for QBER $= 0.11$, while, if the LDPC code has a code rate equal to 0.61, residual bit error rates reach up to 10^{-7} for $Q = 0.015$.

[10] It should be noted that each non-interactive protocol has to face with this difficulty; indeed, the Cascade and the Binary protocols smartly overcome this problem thanks to their interactivity, which, on the other hand, turns out to be also their main drawback.

Implementation of an LDPC-based rate-adaptive information reconciliation scheme

In this section, we briefly recall the rate-adaptive protocol based on LDPC codes proposed in [106], with some improvements resulting from [116].

The protocol takes as input the following parameters:

- a fixed codeword length, n,
- a set $\mathcal{C}$ of N LDPC codes with rates $\{R_i\}_{i\in[1,N]}$ and parity check matrices $\mathbf{H}_1, \ldots, \mathbf{H}_N$,
- a target frame error rate Φ,
- an estimate for the QBER, $\hat{Q}$.

Depending on these parameters, the best punctured LDPC code obtainable from the codes in $\mathcal{C}$ is derived. First, the target reconciliation efficiency η_{EC} is computed according to the result shown in [116], which is based on results similar to those presented in [117, 118]:

$$\eta_{\mathrm{EC}} \approx \eta_{\mathrm{LDPC}}(h_2(Q) + \sqrt{\frac{v(\hat{Q})}{n}}Q^{-1}(1-\Phi))/h_2(\hat{Q}) \tag{3.90}$$

where $\eta_{\mathrm{LDPC}} > 1$ is a parameter describing the efficiency of the codes which takes into accounts non-idealities (such as limited codeword length). Then, the optimal LDPC code and its puncturing parameters are derived by the following procedure:

Choice of the LDPC code and of the puncturing parameters.

1. Choose the i-th LDPC code in $\mathcal{C}$ with the rate R_i closest to the optimal one for the estimated QBER, that is

$$R_i = \underset{R^*\in\mathcal{C}}{\arg\min}\{R^* + \eta_{\mathrm{EC}}h_2(\hat{Q}) - 1\} \tag{3.91}$$

2. Determine the optimal puncturing positions $\mathbf{P}$; this could be done, for instance, by using the untainted puncturing algorithm proposed in [119].

3. Find the optimal number of punctured bits for the estimated QBER as

$$p = n - \frac{nR_i}{1-\eta_{\mathrm{EC}}h_2(\hat{Q})} \tag{3.92}$$

We are now ready to introduce the error correction mechanism.

Rate-adaptive LDPC error correction mechanism.
$\forall$ pair of $(n-p)$-bit sequences $(\mathbf{x}_{\mathrm{S}}^{(i)}, \mathbf{y}_{\mathrm{S}}^{(i)})$, compute the reconciled block $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$ as follows

1. Define a pair of n-bit words $(\mathbf{v}^{(i)}, \mathbf{w}^{(i)})$ such that, given $\mathcal{P} = \{\mathbf{P}(1), \ldots, \mathbf{P}(p)\}$ and $\bar{\mathcal{P}} = \{1, \ldots, n\}/\mathcal{P}$:

$$\mathbf{v}^{(i)}(\mathcal{P}) = U_p, \quad \mathbf{v}^{(i)}(\bar{\mathcal{P}}) = \mathbf{x}_{\mathrm{S}}^{(i)} \tag{3.93}$$
$$\mathbf{w}^{(i)}(\mathcal{P}) = U'_p, \quad \mathbf{w}^{(i)}(\bar{\mathcal{P}}) = \mathbf{y}_{\mathrm{S}}^{(i)} \tag{3.94}$$

   where U_p and U'_p denote two distinct uniformly random p-bit sequences.

2. Compute the syndrome of the sifted key at Alice

$$s(\mathbf{x}_{\mathrm{S}}^{(i)}) = \mathbf{H}_i \mathbf{x}_{\mathrm{S}}^{(i)} \tag{3.95}$$

   and send it to Bob.

3. At Bob, perform a *maximum-likelihood* decoding with non-zero syndrome (see §3.5.1 on hashing approach), with log-likelihood of the punctured symbols set to 0.

4. Output the decoded word $\hat{\mathbf{x}}_{\mathrm{S}}^{(i)}$

Finally, we implemented the described algorithm with a set of 4 LDPC codes. The codes, with rates $\{0.5, 0.6, 0.7, 0.8\}$, are designed with the *progressive edge growth* (PEG) algorithm [120] given the degree distributions provided in [121] and the fixed codeword length $n = 10^4$. We then compared the reconciliation efficiency of the Winnow scheme ($I_{\max} = 6$ and $L_{\max} = 1024$) with the proposed rate-adaptive LDPC error correcting protocol. In order to allow a fair comparison, we chose $T_{\mathrm{out}} = 10^{-7}$ for the Winnow protocol and $\Phi = 10^{-3}$ for the LDPC correction, being the block size $n = 10^4$. As we can see in figure 3.12,[11] Winnow performs better for low QBER values, i.e., for $Q \lesssim 4 \cdot 10^{-2}$, where the LDPC correction suffers from its less flexible rate adaptation. On the other hand, LDPC correction rapidly outperforms Winnow for higher values of Q, where the LDPC codes work close to their optimal region.

In principle, more efficient LDPC codes with $n = 10^4$ could be designed with different techniques and parameters or, analogously, we could leverage the *cascade* effect for Winnow, so that better reconciliation efficiencies could be achieved. However, besides these technicalities, figure 3.12 highlights the fundamental advantages of the two schemes. Winnow is intrinsically rate-adaptable, its information leakage dynamically changes, so that syndromes are disclosed on the public channel only for those blocks with mismatching parities; it is then more suitable for scenarios where there is at most 1 error per block. On the other hand, LDPC correction exhibits better reconciliation efficiencies in the codes

[11] we only consider values of Q in $[0, 7.5 \cdot 10^{-2}]$, since, with the chosen parameters and with the given set of LDPC codes, the target frame error rate $\Phi = 10^{-3}$ cannot be ensured for higher values of Q.

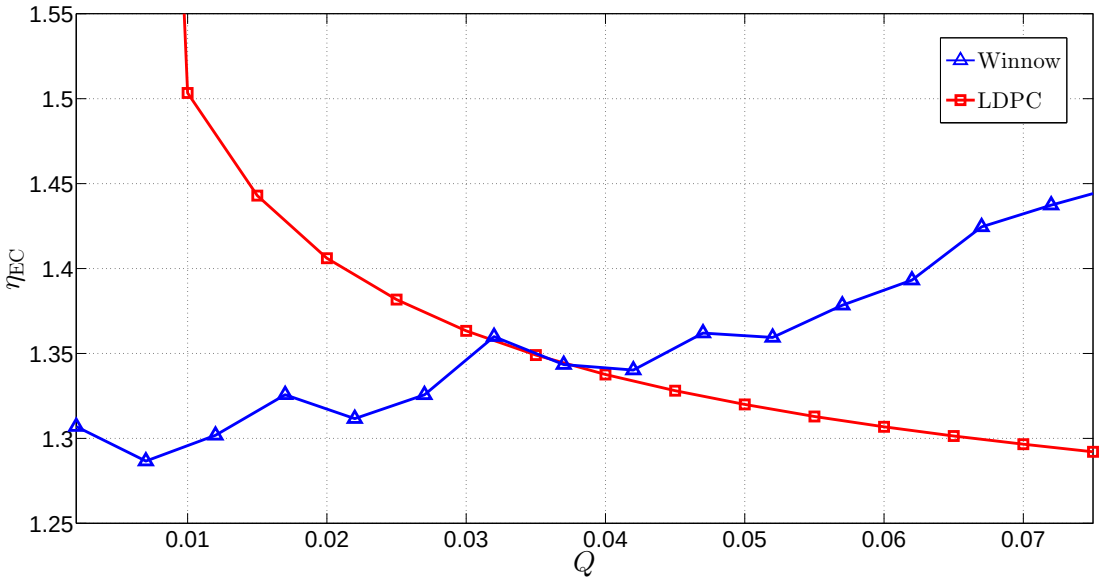


Figura 3.12: Comparison of the reconciliation efficiency for Winnow ($I_{\max} = 6$, $L_{\max} = 1024$, $T_{\text{out}} = 10^7$) and LDPC correction ($n = 10^4$, $\Phi = 10^{-3}$).

optimal region, but, for a given code rate, the same amount of redundancy bits is disclosed on the public channel; hence, for low QBER values, it turns out to be less efficient than Winnow or Cascade. Finally, let us remark that the choice of Winnow is more convenient in scenarios where the number of sifted bits is limited to a few thousands or hundreds (see §4.3 for an experimental application); in that setting, in fact, the efficiency of LDPC codes rapidly drops. On the other hand, as shown in [98] and in subsequent works, LDPC codes provide unequalled efficiency for very long sifted sequences (e.g., $n = 10^6$).

3.5.4 Error verification

After the information reconciliation phase, Alice and Bob must verify that their reconciled strings, X_{R} and Y_{R}, are equal with a sufficiently high probability, so that the correctness constraint (3.2) is ensured. An effective solution, used in [74], is based on universal$_2$ hash functions: Alice and Bob agree on a random universal$_2$ hash function, $g : \{0,1\}^n \to \{0,1\}^{\log_2\lceil 1/\varepsilon_{\text{cor}}\rceil}$, and compute the respective hashes, $g_{\text{A}} = g(X_{\text{R}})$ and $g_{\text{B}} = g(Y_{\text{R}})$. The two hashes are then disclosed on the public channel, and if they match, i.e., $g_{\text{A}} = g_{\text{B}}$, the key distillation procedure continues, if not the whole protocol aborts. A proof that this algorithm ensures ε_{cor}-correctness is provided in [122, Theorem 1].

3.6 Privacy amplification

Privacy amplification is the last step of the secret key agreement scheme described in section 3.1 and, by compressing the reconciled keys, X_{R} and Y_{R}, it allows to distill a pair of privacy amplified keys, S_{A} and S_{B}, on which the adversary has a negligible information. As mentioned in §3.3.3, the nature of this information depends on the assumed attacker model and the key which is output to the privacy amplification algorithm should fulfill

both a secrecy and a uniformity constraint. In this sense, the connection with randomness extractors, described in chapter 2, is evident and privacy amplification for a QKD protocol can be seen as a randomness extraction problem in a quantum adversarial scenario.

In the QKD setting, it is of fundamental importance that the adversary does not know the privacy amplification function before the sifting procedure, because she could otherwise adapt her attack strategy to that specific function. Hence, seeded randomness extractors are to be used in order to address the privacy amplification target: the seed should be chosen uniformly at random at each protocol run and it can be disclosed on the public channel after the sifting procedure. Depending on the specific QKD application, the choice of the optimal seeded extractor may be driven by constraints on the computational overhead, on the extraction efficiency or on the seed length (see §2.3 for a discussion).

In the following sections, we describe some different approaches for determining a crucial parameter for the privacy amplification, that is, the output key length which allows to provide the required level of secrecy. We first recall a fundamental result on the asymptotic information leakage under selective individual attacks, and contextually propose a tighter bound for a specific class of privacy amplification functions. Secondly, we introduce the notion of finite-key analysis and describe a generalization to an arbitrary QKD protocol of the finite-key result recently published in [J1].

3.6.1 Asymptotic information leakage under selective individual attacks

To the best of the author's knowledge, the first results for determining the length of the privacy amplification key appeared in [123, 21]. Here the authors use the mutual information as a security measure (see §3.3.3) and the analysis does not take into account finite-key effects. The main result is stated in the following theorem, where universal$_2$ hash functions are used as seeded randomness extractors.

Theorem 6. *Let $\mathbf{x}_{\mathrm{R}}$ be a random m-bit string with uniform distribution over $\{0,1\}^m$, let $\mathbf{z}_{\mathrm{P}} = e(\mathbf{x}_{\mathrm{R}})$ for an arbitrary eavesdropping function $e : \{0,1\}^m \to \{0,1\}^e$ for some $e \leq m$, let $\ell < m - e$ be a positive security parameter and let $\ell = m - e - s$. If Alice and Bob choose $S = G(\mathbf{x}_{\mathrm{R}})$ as their secret key, where $G(\cdot)$ is chosen at random from a universal$_2$ class of hash functions from $\{0,1\}^m$ to $\{0,1\}^\ell$, then Eve's expected information about the secret key S, given $\mathbf{z}_{\mathrm{P}}$ and G, satisfies*

$$I(\mathbf{S}; \mathbf{G}, \mathbf{z}_{\mathrm{P}}) \leq \frac{2^{-s}}{\ln(2)}. \tag{3.96}$$

In the remainder of this section, we propose a tighter bound on $I(\mathbf{S}; \mathbf{G}, \mathbf{z}_{\mathrm{P}})$ and we briefly discuss the effects of finite-key effects on the choice of the privacy amplification output length. More results are shown in chapter 4 for some specific QKD implementations.

Tight bound for leaked information under selective individual attacks

In this section we describe a privacy amplification framework for bounding the amount of leaked information in the presence of selective individual attacks; this analysis is presented

in [C1].

Let us start by noting that the nature of leaked information is twofold: *deterministic* and known to the legitimate parties on one hand, due to the disclosure of L_{EC} bits in the information reconciliation phase, and *random* and not known to the legitimate parties on the other one, due to eavesdropping on the quantum channel. Therefore, in order to perform an effective privacy amplification, we could consider the two contributions separately, even though, typically, they are treated together.[12] We then start by proposing a common framework for the compensation of both the deterministic and the unknown leakage; we call the former *bit deletion* and the latter *core privacy amplification.*

Let us consider the processing (either bit deletion or core privacy amplification) of the n-bit strings $\mathbf{x}$ and $\mathbf{y}$. Let $\mathbf{x}_{\mathrm{G}}$ and $\mathbf{y}_{\mathrm{G}}$ be the ℓ-bit string which are output to Alice and Bob respectively and let $\mathbf{z}$ be the u-bit string which represents the bits of information leaked to Eve. For a generic attack in which Eve has observed some u-bit linear function $\mathbf{z}$ of $\mathbf{x}$, we can write the input-output relationships as follows (we omit $\mathbf{y}_{\mathrm{G}}$ as, after the information reconciliation procedure, it should be $\mathbf{x} = \mathbf{y}$ except with probability $\varepsilon_{\mathrm{cor}}$):

$$\mathbf{z} = \mathbf{G}_{\mathrm{E}}\mathbf{x} \tag{3.97}$$

$$\mathbf{x}_{\mathrm{G}} = \mathbf{G}_{\mathrm{L}}\mathbf{x} \tag{3.98}$$

where $\mathbf{G}_{\mathrm{E}}$ is a $u \times n$ matrix and $\mathbf{G}_{\mathrm{L}}$ is a $\ell \times n$ hashing matrix randomly chosen. As output to the privacy amplification, one wishes to obtain:

1. perfect uniformity: $H_{\mathbf{G}_{\mathrm{L}}}(\mathbf{x}_{\mathrm{G}}) = \ell$
2. perfect secrecy: $I_{\mathbf{G}_{\mathrm{L}},\mathbf{G}_{\mathrm{E}}}(\mathbf{x}_{\mathrm{G}}; \mathbf{z}) = 0$

where the subscripts $\mathbf{G}_{\mathrm{L}}$ and $\mathbf{G}_{\mathrm{E}}$ in the entropy and mutual information underline that these quantities depend on the two matrices; please also note that classical mutual information is a composable security definition in the attack scenario we here consider (see discussion in §3.3.3).

Let $\mathcal{N}(\cdot)$ denote the null space of a matrix. The following propositions are proven:

Proposition 9. *If* $\mathbf{x}$ *is uniformly distributed and* $\mathbf{G}_{\mathrm{L}}$ *has full row rank, then the entropy of* $\mathbf{x}_{\mathrm{G}}$ *is maximal.*

Dimostrazione. Let us start by considering that the entropy of an ℓ-bit string $\mathbf{x}_{\mathrm{G}}$ is bounded by ℓ, that is, $H(\mathbf{x}_{\mathrm{G}}) \leq \ell$, and the equality holds when ℓ bits are independent and identically distributed. Furthermore, given that $\mathbf{x}_{\mathrm{G}} = \mathbf{G}_{\mathrm{L}}\mathbf{x}$, the probability mass function of $\mathbf{x}_{\mathrm{G}}$ can be expressed as

$$p_{\mathbf{x}_{\mathrm{G}}}(\mathbf{b}) = P[\mathbf{x}_{\mathrm{G}} = \mathbf{b}] = P[\mathbf{G}_{\mathrm{L}}\mathbf{x} = \mathbf{b}] = P[\mathbf{x} \in I_{\mathbf{G}_{\mathrm{L}}}(\mathbf{b})] = \sum_{\mathbf{a} \in I_{\mathbf{G}_{\mathrm{L}}}(\mathbf{b})} p_{\mathbf{x}}(\mathbf{a}). \tag{3.99}$$

[12] some protocols for information reconciliation (e.g., Cascade [96] and Winnow [97]) already embedded a bit deletion mechanism in the original proposals. In the present work, however, we prefer to separate reconciliation from deletion, as the latter is more properly situated in the context of privacy amplification.

where $I_{\mathbf{G}_\mathrm{L}}(\mathbf{b})$ is the anti-image of $\boldsymbol{b}$ given $\mathbf{G}_\mathrm{L}$, that is,

$$I_{\mathbf{G}_\mathrm{L}}(\mathbf{b}) = \{\mathbf{a} : \mathbf{G}_\mathrm{L}\mathbf{a} = \mathbf{b}\} = \mathbf{a}_0 + \mathcal{N}(\mathbf{G}_\mathrm{L}), \tag{3.100}$$

being $\mathbf{a}_0$ an arbitrary particular solution such that $\mathbf{G}_\mathrm{L}\mathbf{a}_0 = \mathbf{b}$.

Now, since we assume $\mathbf{x}$ to be uniformly distributed over $\{0,1\}^n$ and $\mathrm{rank}(\mathbf{G}_\mathrm{L}) = \ell$, we can rewrite Eq.(3.99) as follows:

$$p_{\mathbf{x}_\mathrm{G}}(\mathbf{b}) = \sum_{\mathbf{a}\in I_{\mathbf{G}_\mathrm{L}}(\mathbf{b})} 2^{-n} = |I_{\mathbf{G}_\mathrm{L}}(\mathbf{b})|2^{-n} = |\mathcal{N}(\mathbf{G}_\mathrm{L})|2^{-n} = 2^{n-\mathrm{rank}(\mathbf{G}_\mathrm{L})}2^{-n} = 2^{-\mathrm{rank}(\mathbf{G}_\mathrm{L})} = 2^{-\ell}. \tag{3.101}$$

Hence, given that $\mathbf{x}$ is uniformly distributed and $\mathrm{rank}(\mathbf{G}_\mathrm{L}) = \ell$, we have $H(\mathbf{x}_\mathrm{G}) = \ell$, that is, the entropy of $\mathbf{x}_\mathrm{G}$ is maximal. In addition, one can easily prove that the condition on the row rank of $\mathbf{G}_\mathrm{L}$ is not only sufficient but also necessary. On the contrary, having a uniformly distributed input $\mathbf{x}$ is not necessary, as other distributions may satisfy the maximal entropy condition.

□

Proposition 10. *If* $\dim\mathcal{N}(\mathbf{G}_\mathrm{E}) - \dim\left(\mathcal{N}(\mathbf{G}_\mathrm{E}) \cap \mathcal{N}(\mathbf{G}_\mathrm{L})\right) = \mathrm{rank}(\mathbf{G}_\mathrm{L})$ *and* $\mathbf{x}$ *is uniform over* $\{0,1\}^n$*, then* $\mathbf{x}_\mathrm{G}$ *is uniform and perfectly secret.*

Dimostrazione. In order to prove the claimed statement, we must show that $\mathbf{x}_\mathrm{G}$ and $\mathbf{z}$ are statistically independent, that is, $p_{\mathbf{x}_\mathrm{G}|\mathbf{z}} = p_{\mathbf{x}_\mathrm{G}}$. As shown in the proof of proposition 9, if $\mathbf{x}$ is uniformly distributed over $\{0,1\}^n$, then also the distribution of $\mathbf{x}_\mathrm{G}$ is uniform over $\{0,1\}^{\mathrm{rank}(\mathbf{G}_\mathrm{L})}$. Hence,

$$p_{\mathbf{x}_\mathrm{G}}(\mathbf{b}) = \frac{1}{\mathrm{rank}(\mathbf{G}_\mathrm{L})}, \qquad \forall\ \mathbf{b} \in \{0,1\}^{\mathrm{rank}(\mathbf{G}_\mathrm{L})}. \tag{3.102}$$

By Bayes's theorem [2], we can write $p_{\mathbf{x}_\mathrm{G}|\mathbf{z}}$ as follows

$$\begin{aligned} p_{\mathbf{x}_\mathrm{G}|\mathbf{z}}(\mathbf{b}|\mathbf{c}) = \frac{p_{\mathbf{x}_\mathrm{G}\mathbf{z}}(\mathbf{b},\mathbf{c})}{p_{\mathbf{z}}(\mathbf{c})} = \frac{P[\mathbf{G}_\mathrm{L}\mathbf{x} = \mathbf{b}, \mathbf{G}_\mathrm{E}\mathbf{x} = \mathbf{c}]}{P[\mathbf{G}_\mathrm{E}\mathbf{x} = \mathbf{c}]} = \frac{P[\mathbf{x} \in I_{\mathbf{G}_\mathrm{L}}(\mathbf{b}), \mathbf{x} \in I_{\mathbf{G}_\mathrm{E}}(\mathbf{c})]}{P[\mathbf{x} \in I_{\mathbf{G}_\mathrm{E}}(\mathbf{c})]} = \\ = \frac{|\mathcal{N}(\mathbf{G}_\mathrm{E}) \cap \mathcal{N}(\mathbf{G}_\mathrm{L})|}{|\mathcal{N}(\mathbf{G}_\mathrm{E})|} = \frac{2^{\dim(\mathcal{N}(\mathbf{G}_\mathrm{E})\cap\mathcal{N}(\mathbf{G}_\mathrm{L}))}}{2^{\dim(\mathcal{N}(\mathbf{G}_\mathrm{E}))}}, \end{aligned} \tag{3.103}$$

where, again, $I_{\mathbf{G}_\mathrm{L}}(\mathbf{b})$ and $I_{\mathbf{G}_\mathrm{E}}(\mathbf{c})$ represent, respectively, the anti-image of $\mathbf{b}$ according to $\mathbf{G}_\mathrm{L}$ and the anti-image of $\mathbf{c}$ according to $\mathbf{G}_\mathrm{E}$. Given that, by hypothesis, $\dim(\mathcal{N}(\mathbf{G}_\mathrm{E})) - \dim(\mathcal{N}(\mathbf{G}_\mathrm{E}) \cap \mathcal{N}(\mathbf{G}_\mathrm{L})) = \mathrm{rank}(\mathbf{G}_\mathrm{L})$, we get

$$p_{\mathbf{x}_\mathrm{G}|\mathbf{z}}(\mathbf{b}|\mathbf{c}) = \frac{2^{\dim(\mathcal{N}(\mathbf{G}_\mathrm{E})\cap\mathcal{N}(\mathbf{G}_\mathrm{L}))}}{2^{\dim(\mathcal{N}(\mathbf{G}_\mathrm{E}))}} = \frac{1}{2^{\mathrm{rank}(\mathbf{G}_\mathrm{L})}}, \qquad \forall\ \mathbf{x}_\mathrm{G} \in \{0,1\}^{\mathrm{rank}(\mathbf{G}_\mathrm{L})},\ \mathbf{z} \in \{0,1\}^u. \tag{3.104}$$

Therefore, under the above assumptions, $\mathbf{x}_\mathrm{G}$ is statistically independent of $\mathbf{z}$.

□

From these propositions, the following corollaries immediately follow:

Corollary 1. $\mathbf{x}_{\mathrm{G}}$ *is uniform and perfectly secret if the following conditions hold:*

1. $\mathbf{x}$ *is uniformly distributed over* $\{0,1\}^n$,
2. $\mathrm{rank}(\mathbf{G}_{\mathrm{L}}) = \ell$,
3. $\dim(\mathcal{N}(\mathbf{G}_{\mathrm{E}})) - \dim(\mathcal{N}(\mathbf{G}_{\mathrm{E}}) \cap \mathcal{N}(\mathbf{G}_{\mathrm{L}})) = \ell$.

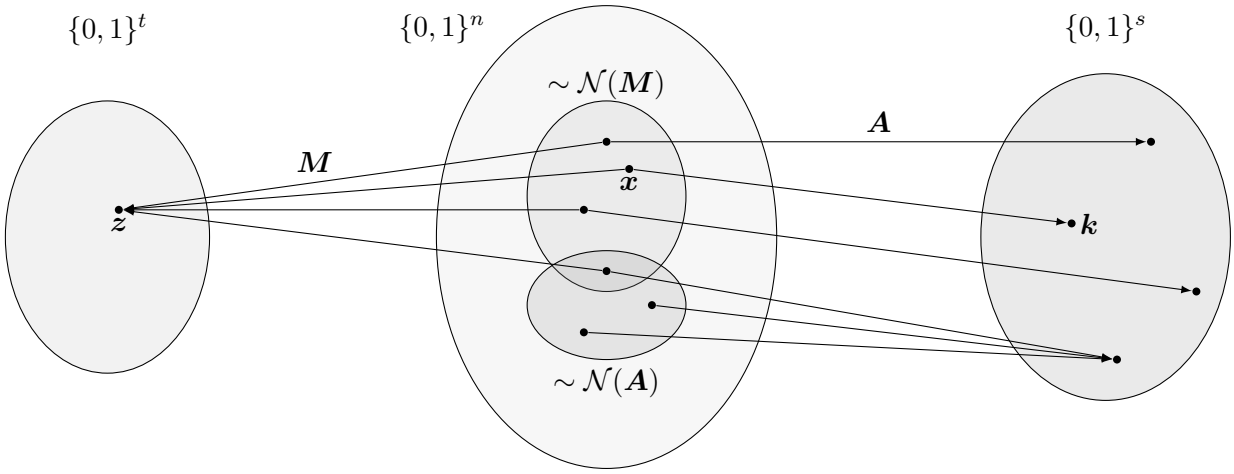


Figura 3.13: Illustration of Proposition 10.

Corollary 2. *The information leaked to Eve about* $\mathbf{x}_{\mathrm{G}}$*, given the eavesdropper matrix* $\mathbf{G}_{\mathrm{E}}$ *and the privacy amplification matrix* $\mathbf{G}_{\mathrm{L}}$*, is*

$$I_{\mathbf{G}_{\mathrm{L}},\mathbf{G}_{\mathrm{E}}}(\mathbf{x}_{\mathrm{G}};\mathbf{z}) = \ell - (\dim \mathcal{N}(\mathbf{G}_{\mathrm{E}}) - \dim(\mathcal{N}(\mathbf{G}_{\mathrm{E}}) \cap \mathcal{N}(\mathbf{G}_{\mathrm{L}}))) \quad (3.105)$$

$$= \ell - \mathrm{rank}(\mathbf{G}_{\mathrm{L}} \boldsymbol{N}_{\mathbf{G}_{\mathrm{E}}}) \quad (3.106)$$

where $\boldsymbol{N}_{\mathbf{G}_{\mathrm{E}}}$ *is any matrix whose columns form a basis for* $\mathcal{N}(\mathbf{G}_{\mathrm{E}})$.

Compensating error correction leakage: bit deletion

Let $\mathbf{x}_{\mathrm{R}}$ and $\mathbf{x}_{\mathrm{D}}$ be respectively the n-bit reconciled key and the m-bit bit-deleted reconciled key and let $\mathbf{z}_{\mathrm{D}}$ be the p information bits associated with $\mathbf{x}_{\mathrm{R}}$ and known to the eavesdropper thanks to the communication over the public channel. We can describe the input-output relationship according to the structure detailed in the previous section, replacing $\mathbf{G}_{\mathrm{E}}$ with $\mathbf{D}_{\mathrm{E}}$ and $\mathbf{G}_{\mathrm{L}}$ with $\mathbf{D}_{\mathrm{L}}$, that is:

$$\mathbf{z}_{\mathrm{D}} = \mathbf{D}_{\mathrm{E}} \mathbf{x}_{\mathrm{R}} \quad (3.107)$$

$$\mathbf{x}_{\mathrm{D}} = \mathbf{D}_{\mathrm{L}} \mathbf{x}_{\mathrm{R}} \quad (3.108)$$

where $\mathbf{D}_{\mathrm{E}}$ is a $p \times n$ matrix, *known* to the legitimate parties, which, given $\mathbf{x}_{\mathrm{R}}$ as input, outputs the bits transmitted on the public channel and $\mathbf{D}_{\mathrm{L}}$ is the $m \times n$ matrix which,

given $\mathbf{x}_{\mathrm{R}}$ as input, outputs the *bit-deleted* reconciled key $\mathbf{x}_{\mathrm{D}}$. Since $\mathbf{D}_{\mathrm{E}}$ is completely known to the legitimate parties as soon as the reconciliation protocol is specified, a matrix $\mathbf{D}_{\mathrm{L}}$ which perfectly counteracts the error correction leakage can be designed. In particular, it should be noted that, in addition to the conditions listed in corollary 1, matrix $\mathbf{D}_{\mathrm{L}}$ has to be a sub-matrix of the identity, since it has to output a subset of $\mathbf{x}_{\mathrm{R}}$ which corresponds to the reconciled key deprived of some bits in order to balance the ones revealed for error correction. We now present a bit deletion algorithm as applied to the Winnow protocol.

Bit deletion in Winnow. In this analysis, we restrict the bit deletion algorithm to blocks with mismatching parity. For each of these b-bit blocks, $\mathbf{x}_{\mathrm{S}}^{(i)}$, and given a $(2^p-1, 2^p-p-1)$ Hamming code with parity check matrix $\mathbf{H} \in \{0,1\}^{p\times b}$, $p = \log_2(b)$ and $p' = p+1$, Winnow [97], after the preliminary parity bit, sends over the public channel its syndrome according to the chosen code, that is, $\mathbf{c}^{(i)} = [\oplus\mathbf{x}_{\mathrm{S}}^{(i)}, \mathbf{H}\mathbf{p}_{\mathrm{S}}^{(i)}]$ for each $i \in \{1,\dots,B_{\mathrm{mis}}\}$, being B_{mis} the number of blocks with mismatching parities (please refer to §3.5.2 for the notation). In order to counteract the disclosure of these p' bits, the authors of [97] propose to remove in each block the bits at positions $\{2^j\}$, where $j \in \{0,\dots,p'-1\}$.

For example, if we choose a $(7,4)$ Hamming code, $\mathbf{D}_{\mathrm{L}}$ is a diagonal block matrix,

$$\mathbf{D}_{\mathrm{L}} = \begin{bmatrix} \mathbf{D}_{\mathrm{L}}^{(1)} & 0 & \cdots & \cdots & 0 \\ 0 & \mathbf{D}_{\mathrm{L}}^{(2)} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & \mathbf{D}_{\mathrm{L}}^{(B_{\mathrm{mis}})} \end{bmatrix}, \tag{3.109}$$

whose blocks are B_{mis} identical matrices $\mathbf{D}_{\mathrm{L}}^{(i)} \in \{0,1\}^{(b-p')\times b}$ defined as

$$\mathbf{D}_{\mathrm{L}}^{(i)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad i \in \{1,\dots,B_{\mathrm{mis}}\}. \tag{3.110}$$

We highlight that each $\mathbf{D}_{\mathrm{L}}^{(i)}$ has full rank; this involves that also $\mathbf{D}_{\mathrm{L}}$ has full rank, given its diagonal structure. Thus, if $\mathbf{x}_{\mathrm{S}}$ is uniformly distributed, proposition 9 ensures that the entropy of the bit-deleted key is maximal for such a choice of $\mathbf{D}_{\mathrm{L}}$.

On the other hand, in the considered scenario $\mathbf{D}_{\mathrm{E}}$ is a block diagonal matrix

$$\mathbf{D}_{\mathrm{E}} = \begin{bmatrix} \mathbf{D}_{\mathrm{E}}^{(1)} & 0 & \cdots & \cdots & 0 \\ 0 & \mathbf{D}_{\mathrm{E}}^{(2)} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & \mathbf{D}_{\mathrm{E}}^{(B_{\mathrm{mis}})} \end{bmatrix}, \qquad i \in \{1,\dots,B_{\mathrm{mis}}\}, \tag{3.111}$$

whose blocks are B_{mis} identical matrices $\mathbf{D}_{\mathrm{E}}^{(i)} \in \{0,1\}^{p'\times b}$ defined as

$$\mathbf{D}_{\mathrm{E}}^{(i)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad i \in \{1, \ldots, B\}. \tag{3.112}$$

Again, each $\mathbf{D}_{\mathrm{E}}^{(i)}$ has full rank, and so does $\mathbf{D}_{\mathrm{E}}$. It is now easy to see that such $\mathbf{D}_{\mathrm{L}}$ and $\mathbf{D}_{\mathrm{E}}$ fulfill also condition 3 in corollary 1, so that the output of the bit deletion is perfectly secret with respect to the error correction leakage.

Counteracting selective individual attacks: core privacy amplification

According to the proposed framework, given the bit-deleted key $\mathbf{x}_{\mathrm{D}} \in \{0,1\}^m$, the final key $\mathbf{x}_{\mathrm{P}} \in \{0,1\}^\ell$ is obtained as $\mathbf{x}_{\mathrm{P}} = \mathbf{P}_{\mathrm{L}}\mathbf{x}_{\mathrm{D}}$, for a given realization of the matrix $\mathbf{P}_{\mathrm{L}} \in \{0,1\}^{\ell \times m}$, whereas the information leaked to the eavesdropper is modelled as $\mathbf{z}_{\mathrm{P}} = \mathbf{P}_{\mathrm{E}}\mathbf{x}_{\mathrm{D}}$. A customary solution in practical QKD systems is to choose $\mathbf{P}_{\mathrm{L}}$ as a randomly generated binary (Toeplitz) matrix [124]. Since both the class of binary matrices and its Toeplitz subclass are universal$_2$, the bound in theorem 6 holds, and is usually invoked to guarantee the security of the distilled key. However in such (and similar) bounds the eavesdropper is assumed to have learned exactly (or at most) e bits of information from the reconciled key, and the measure of information leakage $I(\mathbf{x}_{\mathrm{P}}; \mathbf{z}_{\mathrm{P}}, \mathbf{P}_{\mathrm{L}})$ is taken as average over all possible realizations of $\mathbf{P}_{\mathrm{L}}$.

We consider selective individual attacks (see §3.3.2) where the eavesdropper learns each transmitted bit with probability q (and learns nothing of it with probability $1-q$), independent of all the others. Each realization of such an attack can be modeled as linear with $\boldsymbol{N}_{\mathbf{P}_{\mathrm{E}}} = \boldsymbol{I}_{-\mathbf{j}}$, being $\boldsymbol{I}_{-\mathbf{j}}$ the matrix obtained from the $m \times m$ identity by erasing the columns with indexes in $\mathbf{j}$, corresponding to the bits observed by Eve.

Based on this attack model, we can derive the information leaked to Eve after privacy amplification, for each choice of the hashing matrix $\mathbf{G}_{\mathrm{L}}$.

Proposition 11. *For a given hashing matrix $\mathbf{P}_{\mathrm{L}} \in \{0,1\}^{\ell \times m}$, the average (over the attack statistics) of the information leaked to Eve in the final key $\mathbf{x}_{\mathrm{P}}$ is given by*

$$I_{\mathbf{P}_{\mathrm{L}}}(\mathbf{x}_{\mathrm{P}}; \mathbf{z}_{\mathrm{P}}) = \sum_{\mathbf{j} \in \mathcal{I}_m} q^{\ell(\mathbf{j})}(1-q)^{m-\ell(\mathbf{j})} \mathrm{rank}(\mathbf{P}_{\mathrm{L},-\mathbf{j}}) \tag{3.113}$$

where $\mathcal{I}_m$ denotes the set of all possible index vectors and $\ell(\mathbf{j})$ is the length of vector $\mathbf{j}$.

Dimostrazione. Since, according to the described attack model, the eavesdropped sequences I_m are binomially distributed with parameters (m, q), by taking the average with respect to $\mathbf{G}_{\mathrm{E}}$ (or, equivalently, with respect to the observed indexes vectors $\mathbf{j}$) of Eq. (3.106) in corollary 2, we get Eq. (3.113)

□

Since the value of q can be estimated by Alice and Bob prior to privacy amplification, Eq. (3.113) would yield them the expected amount of information leaked. However, it

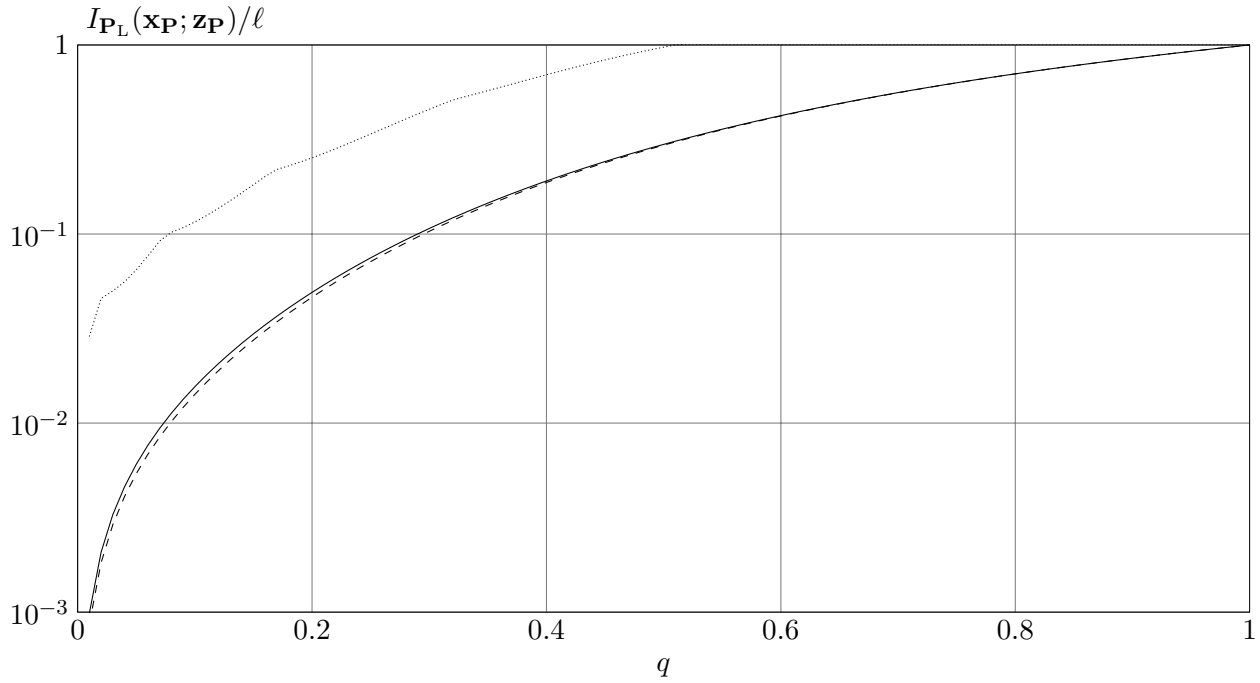


Figura 3.14: Average fraction of key information leaked to Eve under a selective individual attack versus the attack rate q, for the classes of binary (solid line) and binary Toeplitz (dashed line) matrices. Here, $m = 12$, $\ell = 8$. The upper bound in [21, Corollary 4] is also plotted for comparison (dotted line).

should be pointed out that even for moderate values of m the calculation of all the ranks in Eq. (3.113) becomes impractical.

In Figure 3.14 we plot the result of averaging equation 3.113 over a uniform choice of $\mathbf{G}_L$ in the class of full row rank binary matrices and in its Toeplitz subclass. It can be seen that the result is much lower than the average bound in [21], thus showing that Eq. (3.113) provides a significantly tighter bound for the specific attack model we hereby consider.

3.6.2 Finite-key analysis

In standard QKD unconditional security proofs, the final secret key length is upper-bounded by the asymptotic limit which is achievable in the limit of infinitely long keys (see for instance [51]), with the use of shorter blocks leading to lower key rates. However, in QKD implementations, the length of processed blocks is chosen as a trade-off between link duration constraints and memory resources on one side and efficiency (in terms of secret key rate) on the other. This trade-off usually results in long blocks, of at least a million sifted bits. However, in some scenarios such a choice may rather be constrained by the physical channel, as in the perspective use with satellites, where the passage of the orbiting terminal over the ground station is restricted to a few minutes in the case of Low-Earth-Orbit (LEO) satellite [125, 126] or to a fraction of one hour for the Medium-Earth-Orbit (MEO) ones [127]. Hence, for practical use of QKD in cryptography, it is of

crucial importance to develop and test methods that give the achievable secure key rates in the bounded key length scenario, since the number of exchanged bits between the two parties is always finite. In the last years, great efforts from the quantum communication community were directed to this subject, due to its relevance for a number of application scenarios [94, 128, 95, 93, 129, 130, 72]. In this section, we propose a generalization of the finite-key analysis appeared in [J1], which provides security against selective individual attacks (see also §4.3).

Let us consider the following setup. For a given sifted key at Alice, $\mathbf{x}_{\mathrm{S}}$, we define a k-bit subset, $\mathbf{Z}$, and an n-bit subset, $\mathbf{X}$. Similarly, we define two subsets of Bob's sifted key, $\mathbf{y}_{\mathrm{S}}$, thus getting $\mathbf{Z}'$ and $\mathbf{X}'$, respectively. The pair of k-bit strings is devoted to the eavesdropping estimation, whereas the pair of n-bit strings is used for the actual key distillation. We further assume that if an error rate on the k-bit sequences higher than a given threshold Q_{tol} is detected, that is, if

$$Q_{\mathbf{Z}} = \frac{\sum_{i=1}^{k} \mathbf{Z}(i) \oplus \mathbf{Z}'(i)}{k} \leq Q_{\mathrm{tol}}, \tag{3.114}$$

the protocol aborts (we denote by p_{abort} the probability that this happens). Given that Eve performs an intercept-and-resend attack on each qubit, we define the following probabilities:

- $q_{\mathbf{Z}}$, the probability that introduces an error in $\mathbf{Z}'$,
- $q_{\mathbf{X}}$, the probability that Eve knows the attacked bit in $\mathbf{X}$.

As an example, in the efficient BB84 protocol, we get $q_{\mathbf{Z}} = 1/2$ and $q_{\mathbf{X}} = 1$. In BB84 and in B92, if Eve randomly chooses the eavesdropping basis, we get $q_{\mathbf{Z}} = 1/4$ and $q_{\mathbf{X}} = 1/2$.

We now prove the following result.

Theorem 7. *The distilled ℓ-bit key $\mathbf{S}$ is δ_{sec}-PS if*

$$\exists\ a \in \mathbb{N}: \quad f(a,\ell) \leq \frac{\delta_{\mathrm{sec}}}{1 - p_{\mathrm{abort}}} \tag{3.115}$$

where

$$f(a,\ell) = \ell\ \max_{q} \left[I_{qq_{\mathbf{X}}}(a+1, n-a) I_{1-qq_{\mathbf{Z}}}(k(1-Q_{\mathrm{tol}}), kQ_{\mathrm{tol}}+1) \right] + \frac{2^{-(n_{\mathrm{EC}}-\ell-a)}}{\ln 2}, \tag{3.116}$$

with $n_{\mathrm{EC}} = n - L_{\mathrm{EC}} - \lceil \log_2(1/\varepsilon_{\mathrm{cor}}) \rceil$ and $I_x(a,b)$ denoting the regularized incomplete beta function [131, section 6.6],

$$I_x(a,b) = \frac{B(x;a,b)}{B(1;a,b)}\ , \quad B(x;a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\, dt. \tag{3.117}$$

Dimostrazione. Let t be the number of qubits known by Eve in $\mathbf{X}$ among the n sifted bits. Then the Rényi entropy of order 2 for the sifted key, given all the information available to the eavesdropper, is lower-bounded by

$$H_2(\mathbf{X}|V) \geq n_{\mathrm{EC}} - t, \tag{3.118}$$

being $H_2(\mathbf{X}|V) = -\sum_v p_V(v) \log_2 \left(\sum_{\mathbf{s}} p^2_{\mathbf{S}|V}(\mathbf{s}|v) \right)$.

Let us define the following pairs of complementary events:

$$\begin{cases} A = \{Q_{\mathbf{Z}} > Q_{\text{tol}}\} & \text{(protocol abort)} \\ \bar{A} = \{Q_{\mathbf{Z}} \leq Q_{\text{tol}}\} & \text{(no abort)} \end{cases} \tag{3.119}$$

$$\begin{cases} R = \{H_2(\mathbf{X}|V) \geq n_{\text{EC}} - a\} & \text{(tolerable eavesdropping)} \\ \bar{R} = \{H_2(\mathbf{X}|V) < n_{\text{EC}} - a\} & \text{(non-tolerable eavesdropping)} \end{cases} \tag{3.120}$$

Also, let us denote by $p_{\mathbf{S}|V}(\boldsymbol{s}|v)$ the conditional probability mass function of the key $\mathbf{S}$ given the adversarial classical information V, for $\mathbf{S} = \boldsymbol{s}$ and $V = v$. Then, by splitting $\bar{A}$ into $R \cap \bar{A}$ and $\bar{R} \cap \bar{A}$, we get

$$\begin{aligned} H(\mathbf{S}|V) &= \mathbb{E}[-\log_2 p_{\mathbf{S}|V}(\mathbf{S}|V)|\bar{A}] && (3.121) \\ &= \mathbb{E}[-\log_2 p(\mathbf{S}|V)|R \cap \bar{A}]P[R|\bar{A}] && (3.122) \\ &+ \mathbb{E}[-\log_2 P(\mathbf{S}|V)|\bar{R} \cap \bar{A}]P[\bar{R}|\bar{A}]. && (3.123) \end{aligned}$$

The multiplication of $H(\mathbf{S}|V)$ by the probability of not aborting yields

$$\begin{aligned} P[\bar{A}]H(\mathbf{S}|V) &= \mathbb{E}[-\log_2 p(\mathbf{S}|V)|R \cap \bar{A}]P[R|\bar{A}]P[\bar{A}] && (3.124) \\ &+ \mathbb{E}[-\log_2 P(\mathbf{S}|V)|\bar{R} \cap \bar{A}]P[\bar{R}|\bar{A}]P[\bar{A}]\,. && (3.125) \end{aligned}$$

Then, we can write

$$\begin{aligned} P[\bar{A}]\ell - P[\bar{A}]H(\mathbf{S}|V) &= P[\bar{A}]\ell - \mathbb{E}[-\log_2 p_{\mathbf{S}|V}(\mathbf{S}|V)|R \cap \bar{A}]P[R \cap \bar{A}] && (3.126) \\ &- \mathbb{E}[-\log_2 p_{\mathbf{S}|V}(\mathbf{S}|V)|\bar{R} \cap \bar{A}]P[\bar{R} \cap \bar{A}] && (3.127) \end{aligned}$$

and, again splitting $\bar{A}$ into $R \cap \bar{A}$ and $\bar{R} \cap \bar{A}$, we get

$$\begin{aligned} P[\bar{A}]\ell - P[\bar{A}]H(\mathbf{S}|V) &= (P[R \cap \bar{A}] + P[\bar{R} \cap \bar{A}])\ell && (3.128) \\ &- \mathbb{E}[-\log_2 P(\mathbf{S}|V)|R \cap \bar{A}]P[R \cap \bar{A}] && (3.129) \\ &- \mathbb{E}[-\log_2 P(\mathbf{S}|V)|\bar{R} \cap \bar{A}]P[\bar{R} \cap \bar{A}] && (3.130) \\ &= P[R \cap \bar{A}](\ell - \mathbb{E}[-\log_2 P(\mathbf{S}|V)|R \cap \bar{A}]) && (3.131) \\ &+ P[\bar{R} \cap \bar{A}](\ell - \mathbb{E}[-\log_2 P(\mathbf{S}|V)|\bar{R} \cap \bar{A}]) && (3.132) \end{aligned}$$

Then, by observing that, trivially, $P[R \cap \bar{A}] < 1$ and $\mathbb{E}[-\log_2 P(\mathbf{S}|V)|\bar{R} \cap \bar{A}] > 0$, we write

$$P[\bar{A}](\ell - H(\mathbf{S}|V)) \leq \ell - \mathbb{E}[-\log_2 P(\mathbf{S}|V)|R \cap \bar{A}] + P[\bar{R} \cap \bar{A}]\ell \tag{3.133}$$

By applying corollary 4 in Ref. [21] to $\mathbf{S}$ we get

$$\mathbb{E}[-\log_2 P(\mathbf{S}|V)|R\cap\bar{A}] \geq \ell - \frac{2^{-(n_{\mathrm{EC}}-\ell-a)}}{\ln 2} \tag{3.134}$$

$$\ell - \mathbb{E}[-\log_2 P(\mathbf{S}|V)|R\cap\bar{A}] \leq \frac{2^{-(n_{\mathrm{EC}}-\ell-a)}}{\ln 2} \tag{3.135}$$

Finally, by plugging (3.135) into (3.133), we get

$$P[\bar{A}](\ell - H(\mathbf{S}|V)) \leq \frac{2^{-(n_{\mathrm{EC}}-\ell-a)}}{\ln 2} + \ell P[\bar{R}\cap\bar{A}]\,. \tag{3.136}$$

From (3.118), we can upper bound the probability on the right-hand side of (3.136) as

$$P[H_2(\mathbf{X}|V) < n_{\mathrm{EC}} - a, Q_{\mathbf{Z}} \leq Q_{\mathrm{tol}}] \leq P[t > a, Q_{\mathbf{Z}} \leq Q_{\mathrm{tol}}] \tag{3.137}$$

$$= P[t > a]P[Q_{\mathbf{Z}} \leq Q_{\mathrm{tol}}], \tag{3.138}$$

since the two events in the right-hand side brackets of equation (3.137) refer to disjoint qubit sets, namely those in $(\mathbf{X}, \mathbf{X}')$ and in $(\mathbf{Z}, \mathbf{Z}')$, respectively, and are therefore independent. Furthermore, according to the selective individual attack model with attack rate q, t is a binomial random variable with parameters $(n, qq_{\mathbf{X}})$. Similarly, the number of measured errors on $\mathbf{Z}'$, $kQ_{\mathbf{Z}}$, is a binomial random variable with parameters $(k, qq_{\mathbf{Z}})$. Therefore, we can rewrite equation (3.138) as

$$P[t > a]P[Q_{\mathbf{Z}} \leq Q_{\mathrm{tol}}] = (1 - F_{n,qq_{\mathbf{X}}}(a))(F_{k,qq_{\mathbf{Z}}}(kQ^{\mathbb{Z}}_{\mathrm{tol}})) \tag{3.139}$$

$$= I_{qq_{\mathbf{X}}}(a+1, n-a) I_{1-qq_{\mathbf{X}}}(k(1-Q^{\mathbb{Z}}_{\mathrm{tol}}), kQ^{\mathbb{Z}}_{\mathrm{tol}}+1), \tag{3.140}$$

with $F_{n,qq_{\mathbf{X}}}(\cdot)$ denoting the cumulative distribution function of a binomial random variable with parameters $(n, qq_{\mathbf{X}})$, and similarly for $F_{k,qq_{\mathbf{Z}}}(\cdot)$. The last step is then assured by equation 6.6.4 in Ref. [131].

Eventually, condition (3.115), together with definition (3.116) and given that $P[Q_{\mathbf{Z}} \leq Q_{\mathrm{tol}}] = 1 - p_{\mathrm{abort}}$, ensures that, for any $q \in [0,1]$, we get

$$\ell - H(\mathbf{S}|V) \leq \frac{\delta_{\mathrm{sec}}}{1 - p_{\mathrm{abort}}}, \qquad \forall\ a, \ell. \tag{3.141}$$

□

Based on (3.115), we can therefore choose the optimal secret key length as

$$\ell = \max\left\{ b : \min_a f(a,b) \leq \frac{\delta_{\mathrm{sec}}}{1 - p_{\mathrm{abort}}} \right\}. \tag{3.142}$$

In chapter 4, we are going to describe an application of this result and compare, in a realistic scenario, the corresponding secret key rate with the one achievable with $\varepsilon_{\mathrm{sec}}$-GS secrecy.

Capitolo 4

Experimental free-space Quantum Key Distribution

Quantum key distribution may be considered the first successful example of quantum information protocol that reached the everyday applications. Commercial devices communicating via optical cables are already operated worldwide; in particular, we mention the Cerberis by ID Quantique,[1] the Cygnus by SeQureNet,[2] the Q-Box by MagiQ,[3] and the QLE by Quintessence Labs.[4] However, the need for optical fibers infrastructures, which entail both high costs and, as long as quantum repeaters are not available, a non-scalable network topology,[5] make the deployment of such schemes unpractical for most application scenarios.

Hence, free space QKD is considered very attractive, as it would eliminate the need for cable links between terminals. Its applications include terrestrial links [132], earth-to-moving-terminal links (e.g., [133]), and the very relevant case of key exchange with orbiting terminals, that is, satellite QKD. This extension of the QKD application has been fostered for years, being included in the major Quantum Information road-maps [134, 135, 136], and has been the subject of several feasibility studies [137, 126, 138, 139].

This chapter is devoted to the description of the results obtained in three different QKD experiments over free-space links, performed by using a full-fledged system capable of distilling secret keys in real-time. The architecture of the implemented software is detailed in §4.1.

A first experiment implementing the B92 protocol over a 50 meters indoor free-space link is then described in §4.2. The experiment led to the publication of [C2,C3] and proves the feasibility of real-time key distillation with our prototype setup while considering the selective individual attacker model.

A second experiment, based on the efficient BB84 protocol, is then detailed in §4.3. The experiment investigates the effect of noise on the secret key rate computed with finite-

[1] `http://www.idquantique.com`
[2] `http://www.sequrenet.com`
[3] `http://magiqtech.com`
[4] `http://quintessencelabs.com`
[5] an end-to-end connection is in fact required for creating each shared secret key pair.

key analysis and according to two different attack models, i.e., general quantum attacks and selective individual attacks. The described results were published in [J1].

We finally detail a novel approach to QKD over free-space links, which leverage the atmospheric turbulence as a resource for enabling key distillation even when the average QBER is higher than the security threshold, that is, $Q > 11\%$. In particular, a classical probe is jointly used with the quantum channel for performing an adaptive real-time selection of the transmitted packets corresponding to a sufficiently low QBER. It was shown that a correlation between the probe signal intensity and the QBER exists and can be successfully exploited for key generation. The work is to be submitted [P1].

Let us conclude this introduction by mentioning that a further experiment for investigating the feasibility of inter-satellite experimental QKD has been partially prepared as a part of this thesis, within a joint project with Thales-Alenia Space Italy funded by the European Space Agency (*Applications of optical-quantum links to GNSS*, Specification, ESA Statement of Work TEC-MMO/2010/47). Unfortunately, due to some technical problems, the experiment has not been performed yet and, therefore, was not included in this manuscript. We refer the interested reader to the preliminary results presented in [140].

4.1 Software implementation

We start by describing the architecture of the developed software, for whose implementation a consistent part of this Ph.D. has been spent. The structure of the software has been designed to be as modular as possible, so that modules implementing new algorithms of protocols can be easily plugged in. Also, it can be used with any QKD protocol with minor changes; as of now, the system is fully compatible with the BB84, eBB84 and B92 protocol.

The whole software has been developed in Matlab, as this work is mainly focused on investigating the feasibility of experimental QKD schemes and on their analysis rather than on raw, speed performance. As shown in figure 4.1, the software has three core components: the FPGA software interface, the processing and the networking modules. The modules are described in more detail in the following subsections.

The developed software can be used either in *local* or in *network* mode, being the former suitable for off-line key distillation and the latter designed for real-time processing. In general, raw and sifted keys are stored in indexed files as soon as they are successfully sent or received, by Alice and Bob respectively. On the other hand, final keys are stored in a SQL database, so that they are readily available to other applications. In principle, this may open a security breach for real-world applications, as a naïve implementation of the database may incur into attacks such as SQL injections; however, we are not delving into these issues, since we believe that this solution is sound for the sake of our experimental setup and is beyond the scope of this work.

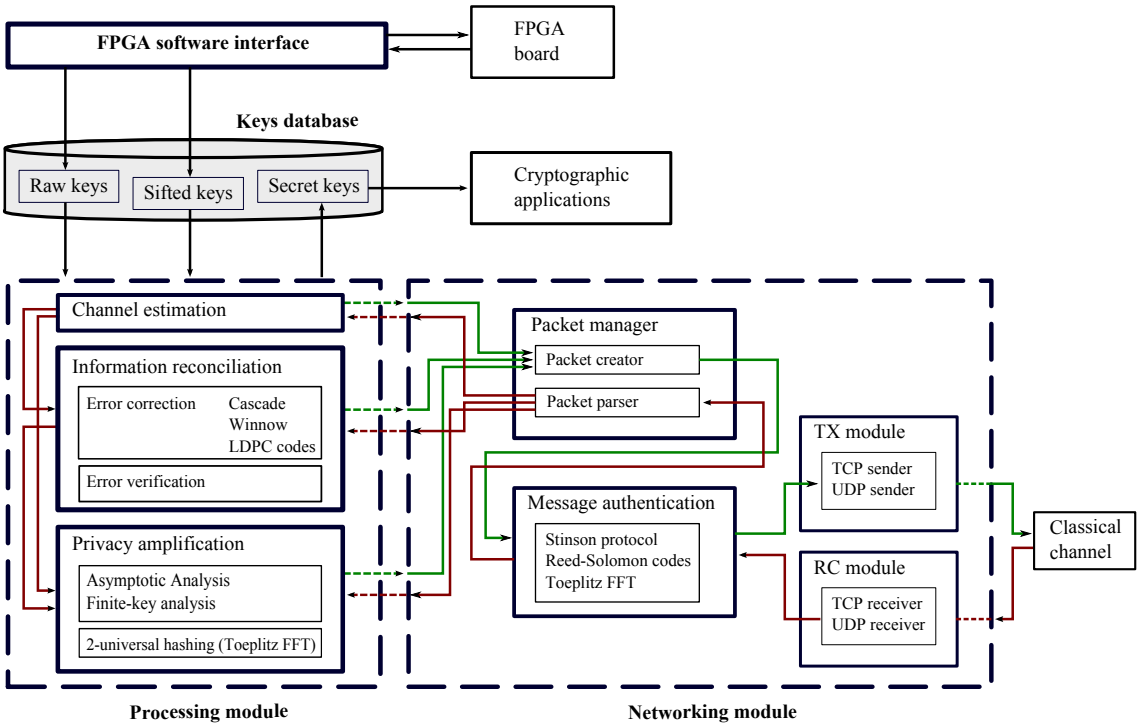


Figura 4.1: Software architecture.

4.1.1 FPGA software interface

The optical subsystem is controlled by two FPGAs, one at the transmitter and one at the receiver. Details on the hardware setup can be found, for each described experiment, at the beginning of the corresponding section.

We are providing a technical description of this module. Suffice it to say that, in general, this module handles the communication with the FPGA, that is performed via UDP over Ethernet. Configuration sequences and raw key packets are sent to Alice's FPGA, whereas Bob receives the output of the detectors, by means of the Matlab scripts `FPGA_transmitter` and `FPGA_receiver`, respectively.

4.1.2 Processing module

The processing module consists of three main blocks, each devoted to a different phase of the secret key agreement protocol, namely: channel estimation, information reconciliation and privacy amplification.

The channel estimation module computes the transmission losses and, according to the underlying QKD protocol, it selects a subset of sifted bits for estimating the QBER. Furthermore, as for the security countermeasures described in section 4.2, it computes the overall estimated attack probability.

For performing error correction, the user can choose between the Binary protocol [96], the Winnow protocol (see §3.5.2) and the solution based on LDPC codes presented in §3.5.3 and based on [106]. Binary and Winnow share the same iterative structure and their implementation is then strictly modular. Two different set of functions have been

developed, one for performing the actual error correction and the other for the protocol analysis and optimization. As for the first set, the following functions are defined:

`correctionWinnow`	perfoms the Winnow error correction on single blocks.
`correctionBinary`	performs the binary error search on single blocks.
`iterationBW`	performs a single iteration of either the Binary or the Winnow protocol; it splits the sifted sequences into blocks, check the parity of corresponding blocks at the transmitter and at the receiver, calls the error correction function (`correctionWinnow` or `correctionBinary`) and finally permutes the corrected sequences.
`globalBW`	performs a complete series of iterations of either the Binary or the Winnow protocol, by iteratively calling `iterationBW`.

On the other hand, the second set is made of the following functions:

`analyzeIterationBW`	evaluates the performance of a single iteration for either Binary or the Winnow protocol, according to the results presented in §3.5.2
`analyzeGlobalBW`	evaluates the performance of a complete series of iterations of either the Binary or the Winnow protocol
`optimizationBW`	chooses the optimal block size combination for leaking the minimum number of bits on the public channel while still fulfilling a target output bit error rate T_{out} (see §3.5.2); it uses the analytical results output to `analyzeGlobalBW`.

Please note that for every function, it is possible to specify whether bit deletion (see §3.6.1) should be enabled or not. From a practical point of view, the computation of the optimal block sizes for different values of the QBER, Q, and of the target output bit error rate, T_{out}, is performed once; resulting optimization data is stored in a file, one for each T_{out}, so that they are readily available for real-time key distillation.

The LDPC-based error correction procedure described in §3.5.3 is implemented by means of the following functions:

`chooseLdpcCode`	given the estimate of the QBER, it selects the LDPC code with rate R_i closest to the optimal, among the available ones (up to date, 4 different codes are available, with rates $\{0.5, 0.6, 0.7, 0.8\}$; more details can be found in §3.5.3)
`getLdpcPuncturing`	chooses the optimal puncturing positions according to the protocol described in [119] in order to perform rate adaptation.
`correctionLdpc`	performs the actual syndrome decoding for correcting errors in the input block.
`globalLdpc`	performs the whole rate adaptable LDPC protocol: splits the sifted sequences into blocks, chooses the best available LDPC code, computes the syndromes at Alice and performs syndrome decoding at Bob.

Error verification is implemented by a simple function, which compares the hashes of the reconciled keys according to the protocol described in §3.5.4.

Then, the privacy amplification module consists of two main blocks. The first is devoted to the choice of the final key length, which depends on the attacker model and on the (eventual) finite-key analysis; the module takes as input the error and, eventually, the loss estimates, together with the information reconciliation leakage and the required security parameters. In particular, the following functions are implemented:

`getAsymptoticKeyLength`	computes the length of the privacy amplified key according to the asymptotic bound (see §3.4).
`getPragmaticSecKeyLength`	computes the length of the privacy amplified key according to (4.17) or to (4.21), depending on the considered scenario.
`getGeneralSecKeyLength`	computes the length of the privacy amplified key according to (4.19).

The second module implements the actual privacy amplification algorithm and consists of a single function, `privacyAmplification`, which takes as input the reconciled key (eventually after bit deletion), the extractor type (see §2.3). Here, only one solution is available, that is, 2-universal hashing by matrix multiplication; in particular, we use of the class of Toeplitz matrices, which take advantage of a compact representation and of an efficient implementation based on the Fast Fourier Transform (FFT) [37].

4.1.3 Networking module

The networking module is based on a protocol for handling the communication on the classical channel which was jointly designed with the author of [141]. The protocol works at the application layer, so that it can transparently use either TCP or UDP as transport protocols; furthermore, no restrictions on the physical channel and data-link layer are present, though, in the experiments we performed, we used either an ethernet or a wireless 802.11 connection.

The protocol has been designed according to two different *state-transition* models; its structure, in fact, is intrinsically asymmetric, since the two communication parties, Alice and Bob, perform different tasks during the key distillation procedure.

For the authentication of messages sent through the classical channel, three main protocols are available, based on different familes of $\varepsilon_{\text{auth}}$-almost strongly universal hash functions: Toeplitz matrices (as for privacy amplification), Reed-Solomon codes [142] and Stinson's construction [143]. We recall from [144] that a family of hash functions $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ is $\varepsilon_{\text{auth}}$-almost strongly universal hash functions if it is $\varepsilon_{\text{auth}}$-almost universal (see definition 17), and if it further verifies the condition

$$|\{h \in \mathcal{H} : h(x) = y\}| = \varepsilon_{\text{auth}}|\mathcal{H}|, \qquad \forall\, x \in \mathcal{X},\ \forall\, y \in \mathcal{Y}, \tag{4.1}$$

thus bounding the number of functions that may generate a given tag.[6] In particular, the message is first hashed according to an $\varepsilon_{\text{auth}}$-strongly universal hash function and

[6] this is fundamental for authentication schemes, where the adversary should not gain information on the authentication function, as he could otherwise forge message tags.

the authentication tag is then XORed with a secret key of equal length. The security ensured by these solutions can be tailored depending on the system security requirements, and is captured by the parameter $\varepsilon_{\text{auth}}$. The hash functions used for authentication are periodically renewed, in order to ensure that the information potentially leaked to the adversary (see, e.g., [145]) is periodically wiped out. Details on these protocols and on their practical implementation are provided in [141].

Before moving to a more detailed description of the networking module, let us specify that up to date the protocol is fully compatible only with the B92 QKD scheme. Minor changes in the sifting procedure, however, would make it usable with other protocols.

Networking packets

The messages sent over the classical channel for performing the key distillation are encoded into packets with a common header structure, shown in figure 4.2; the payload is then formatted differently depending on the specific packet type. The content of each field of the header is described in the following.

`seq_num` 16-bit sequence number, it is required for synchronizing the transmitter and the receiver and to validate transition events (a transition is triggered only if a packet with the expected sequence number is received).

`type` Identifies the packet type. Currently allowed types and corresponding IDs are listed in table 4.1. The `more` Fragmentation flag, specifies if the packet refers to a single message which was split in multiple packets. It is set to 1 if the packet with subsequent sequence number carries the continuation of the same message, to 0 otherwise (single message packet or last packet of a split message).

`auth` Flag which specifies if the current packet contains the authenticating tag, which is appended to the message. `exp_hash` Flag for signalling that the sender's hash function has expired.

`unused` Dummy field used to make the header length a multiple of 8 bits.

`l_b_l` Last block length: specifies the number of bits in the last byte of the packet.

The packet type defines the content of each packet and is associated with a state transition, as described in the next subsection.

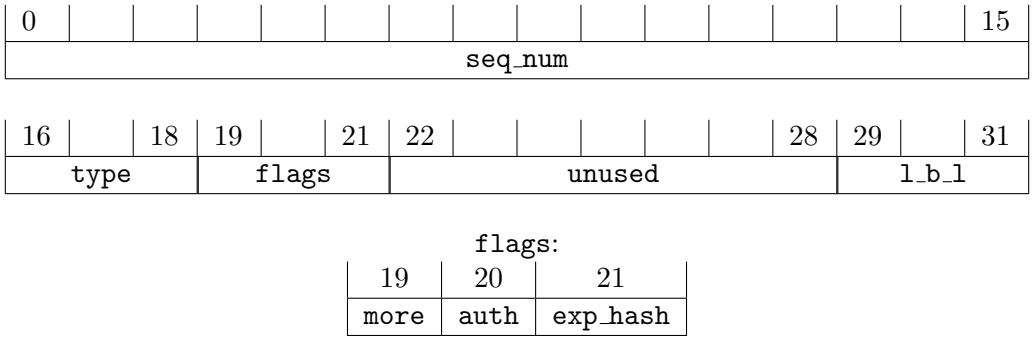


Figura 4.2: Packet header.

`START` Packet for signalling the protocol start, it does not include any payload. It may be sent by Alice and Bob.

`SIFT` Packet containing the sifting information, sent by Bob to Alice. The following cases are distinguished:

1. if an unprocessed sifted key is available, the packet contains an authenticated message with the information required for starting the processing.
2. if no sifted key is available but a key was generated in the previous iteration, the packet contains the header and the authentication tag for the previous round.
3. if no sifted key is available and no key was generated in the previous round, the packet is not authenticated and it contains just the header.

In case 1 and 2, the authentication tag is calculated for the concatenation of all the packets (of any type) that Bob sent from the last authenticated `SIFT` message, including the current one, if a key was generated in the previous round, otherwise just for the current SIFT packet.

`PROCESS` Packet containing data needed to perform the key processing, that is, parameters and data for information reconciliation and privacy amplification. The content of the packet depends on the specific protocol chosen for performing the two key distillation steps. The last `PROCESS` packet sent by Alice during a key generation round includes a tag that authenticates all the messages that were sent starting from the reception of the authenticated `SIFT` packet.

`ABORT` Packet resetting the protocol if some error occurred. Every partial key material which was not properly validated before receiving this packet is wiped out.

For each system, a time-out for the reception of packets is fixed, so that if no packet is received within the allowed time window, the protocol is reset to the *`STARTING`* state.

type	ID
START	0
NEWHASH	1
SIFT	2
PROCESS	3
ABORT	7

Tabella 4.1: Packet types.

State transition models

The state transition diagrams for Alice and Bob are shown in figure 4.3a and 4.3b, respectively. States are represented by blocks and each transition is labelled with the triggering event or packet type, followed by the triggered packet type which is subsequently sent.

In general, in fact, a state transition is triggered either by a packet reception or by a time-out event. If the received packet sequence number is correct and if its type is admitted by the current state, a transition towards the next step is performed. Otherwise, and in the additional case of a time-out event, an `ABORT` message is sent and system is reset to the *STARTING* state.

Let us now describe the system states more in details. For each state, we first provide a brief description and then separately detail the actions taken by Alice and Bob.

STARTING

Start up system state. No active or valid connection to the communication party is present.

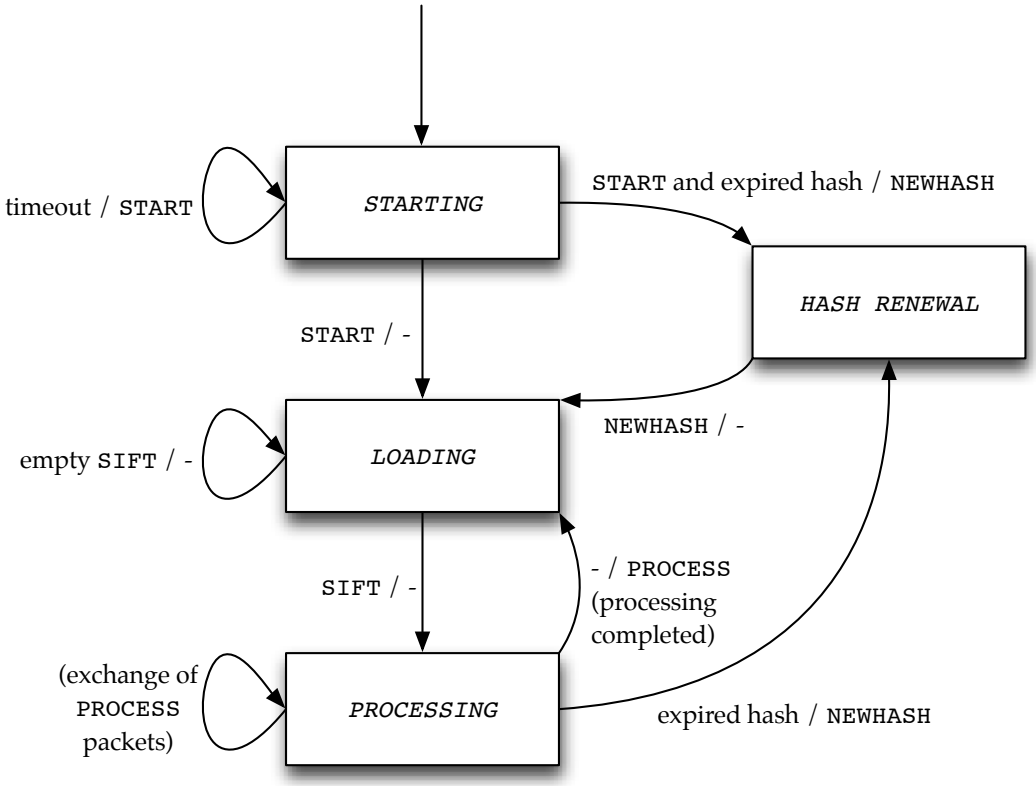
- *Alice.* Periodically send a `START` packet to Bob. If no valid hash function is set up at Alice or if it has expired, the `exp_hash` flag is set to 1. Upon reception of Bob's `START` packet, a transition to the *LOADING* state is triggered, unless an expired hash function has been signalled by one of the two sides. In that case, the transition is to the *HASH RENEWAL* state.

- *Bob.* Upon reception of Alice's `START` packet, reply with a further `START` packet; if a valid hash function is not available, the `exp_hash` flag is set to 1. Then, switch to the *SIFTING* state or, if an expired hash function has been signalled by one of the two sides, to the *HASH RENEWAL* state.

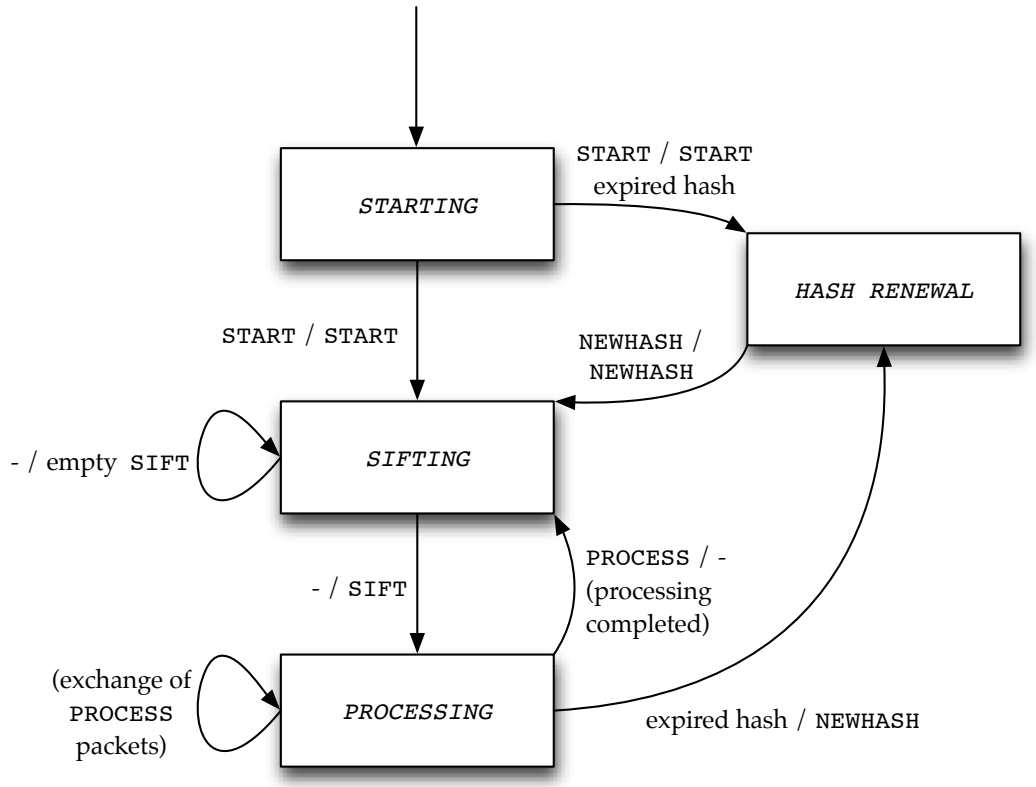
HASH RENEWAL

State triggered when one of the communication parties does have a valid hash function for authentication, and has successfully signalled the event with a `START`, `PROCESS`, or `SIFT` packet. While in this state, Alice chooses the keys to be used for generating the hash function and transmits their IDs to Bob.

- *Alice.* Retrieve the seed for defining a new hash function from the key database and generate it. Then, send a `NEWHASH` packet containing the addresses (IDs) of the retrieved keys; the packet is authenticated with the new hash function. The



(a) Alice



(b) Bob

Figura 4.3: State transition diagrams.

new hash function is correctly validated if Bob replies with a `NEWHASH` packet with valid authentication; in that case, a transition to the *`LOADING`* state is triggered. Otherwise an `ABORT` packet is sent and the whole system is reset to the *`STARTING`* state.

- *Bob.* Wait for a `NEWHASH` packet from Alice. Upon reception of a valid packet, fetch from the key database the keys identified by the enclosed IDs and generate the new hash function. Then, check that the packet has been correctly authenticated by using the new hash function. If this is the case, send an empty validated `NEWHASH` packet to Alice and then trigger a transition to the *`SIFTING`* state, otherwise send an `ABORT` packet and reset to the *`STARTING`* state.

`SIFTING`

State triggered by Bob when a valid `START` packet is received. This state is available only for Bob.

- *Bob.* The following cases are distinguished:

  1. if at least one unprocessed sifted key is available, select it and send to Alice an authenticated `SIFT` packet containing: the sifted key ID, the positions of the sifted bits in the raw key, and the bits to be used for error estimation. Then, trigger a transition to the *`PROCESSING`* state.
  2. if no sifted key is available, send an empty `SIFT` packet to Alice, eventually including the authentication tag for the previous key distillation round if a key was successfully generated. No state transition occurs (wait for available sifted keys).

`LOADING`

State triggered by Alice when a valid `SIFT` packet is received. This state is available only for Alice.

- *Alice.* The following cases are distinguished

  1. **Authentication and sifting** - Upon reception of an authenticated `SIFT` packet with non-empty payload, check authentication tag. If it is not valid, send an `ABORT` packet and trigger a transition to the *`STARTING`* state. If it is valid, store the key generated in the previous round into the key database and perform the following steps
     (a) load the raw key with the index specified in Bob's `SIFT` packet.
     (b) perform sifting and channel estimation.
     (c) send the first `PROCESS` packet to Bob containing information reconciliation protocol parameters, redundancy bits, and privacy amplification parameters (type, seed). In this is the only `PROCESS` packet that is going to be sent

during the current round, an authentication tag of the concatenation of the packet together with the distilled key is embedded, and no state transition occurs. Otherwise, if an iterative protocol for error correction is used, a transition to the *PROCESSING* state is triggered.

2. **Authentication only** - Upon reception of an authenticated SIFT packet with no payload, check the authentication tag: if valid, store the key distilled in the previous round in the key database and stay in the *LOADING* state; if not, send an ABORT packet and trigger a transition to the *STARTING* state.
3. **Empty SIFT packet** - Upon reception of an empty, unauthenticated SIFT packet, do nothing.

PROCESSING

State for classical processing of the sifted keys. Please note that Alice reaches this state only if an iterative protocol for error correction is used, as specified in the description of the *LOADING* state.

- *Alice.* Send and receive PROCESS packets, according to the information reconciliation protocol currently in use. In the last PROCESS packet, besides the data for error correction, include the authentication tag for the concatenation of all the PROCESS packets sent by Alice and of the final key. Then trigger a transition to the *LOADING* state. If an error occurs, send an ABORT packet and switch to the *STARTING* state.
- *Bob.* Send and receive PROCESS packets, according to the information reconciliation protocol currently in use. Upon reception of the last PROCESS packet, generate the final key and check the authenticity of all the received PROCESS packets. If the authentication tag is valid, store the final key in the key database and trigger a transition to the *SIFTING* state; otherwise, send an ABORT packet and switch to the *STARTING* state.

Comments

Let us finally make some concluding remarks on the networking module.

First, the described network protocol ensures that the messages that led to the generation of a key, and were confirmed by an authenticated reply, are genuine and not tampered with by third parties. In fact, all the information that leads to the generation of a key is authenticated by an unconditionally secure authentication scheme, as required by any QKD scheme. Of course, denial of service (DoS) attacks are trivially deployable, by, e.g., flooding either Alice or Bob with ABORT packets, but it should recalled that in general, in QKD systems, the attacker can *always* and *easily* disrupt the key distribution by simply altering the quantum channel statistics (e.g., by performing an intercept-and-resend attack) or by making the channel itself unusable (e.g., by preventing a line-of-sight communication in free-space applications). This kind of attack, however, is inconclusive for the attacker in most application scenarios, as no key is produced and used for further security mechanisms he may want to attack.

Second, the correctness of the final key is ensured by the fact that, in her last `PROCESS` packet, Alice authenticates also the privacy amplified key. In particular, if an $\varepsilon_{\text{auth}}$-almost strongly universal hash function is used for authentication, then ε_{cor}-correctness is ensured with $\varepsilon_{\text{cor}} = \varepsilon_{\text{auth}}$. On the other hand, XORing the tag with a secret key ensures that no information is leaked to the adversary on the final key. Please note, however, that implementing a stand-alone error verification step would be straightforward.

4.2 Experimental B92 over 50 meters free-space link

In this section we report the setup and the results obtained with an experimental QKD system based on the B92 protocol (see §3.4), as described in [C2] and [C3]. The system is designed to provide protection against selective individual attacks, such as intercept-and-resend, unambiguous state discrimination and photon number splitting.

4.2.1 Transmission setup and protocol

The optical setup for our prototype is shown in fig. 4.4. The transmitter (Alice) uses two infra-red (850 nm) attenuated diode lasers to send the bits 0 and 1, encoded in the vertical $|\updownarrow\rangle$ and in the +45° linear $|\nearrow\rangle$ polarization of photons, respectively, that is, according to the preparation basis $\mathbb{P}$. A 808 nm laser beam is also used along for synchronization. The receiver (Bob) uses a dichroic mirror (DM) to separate the information qubits from the synchronization signal: the latter is reflected and detected by an avalanche photodiode, whereas the qubits, trasmitted by the DM, impinge on a 50/50 beam splitter (BS). On either output of the BS, a polarizer and a single photon avalanche photodiode (SPAD) detect the −45° linear $\langle\nwarrow|$ or horizontal $\langle\leftrightarrow|$ polarization photons, respectively, that is, measurements are performed in the $\mathbb{M}$ basis. Each click of either SPAD corresponds to the reception of a sifted 0 or 1, respectively.

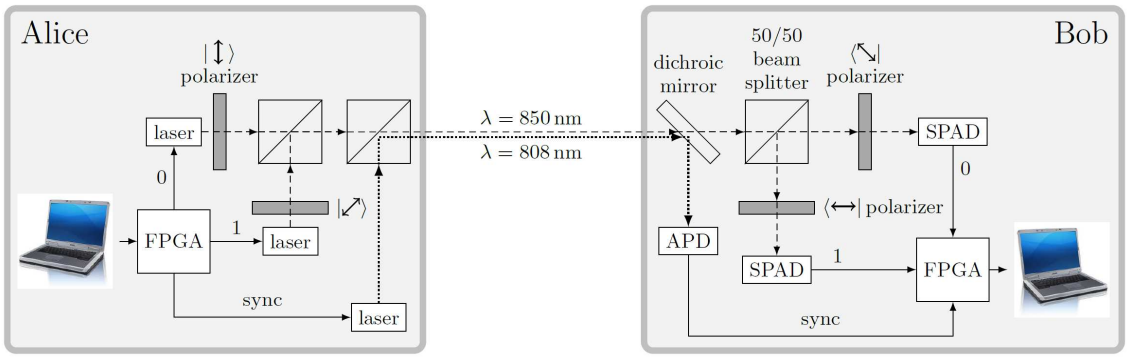


Figura 4.4: Schematic representation of our optical setup

The transmitted data structure is shown in fig. 4.5. A raw key of 288 kbit is divided into 50 packets of 5760 bits each, which are in turn divided into 12 frames for the ease of synchronization. In fact, each frame consists of 11 header slots and 240 payload slots, each with a duration of 800 ns. The header consists of the pattern '100000xxxx1', where 'xxxx' is the 4-bit frame number, encoded one bit per slot in a pulse-duration modulation of the

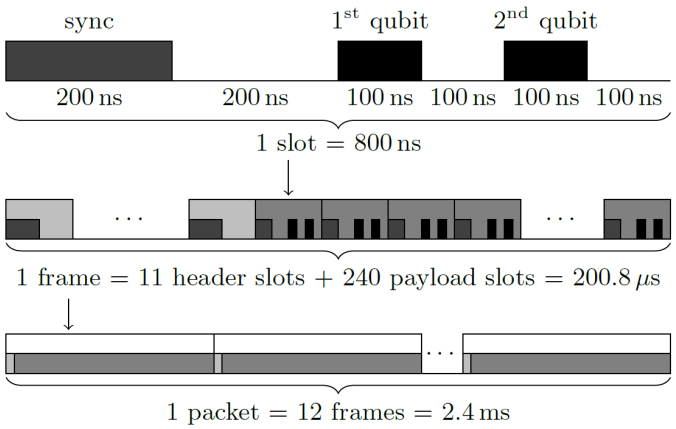


Figura 4.5: Data frame structure

synchronization beam (a 400 ns or 200 ns pulse encodes the bit 1 or 0, respectively). As regards the payload slots, the first 200 ns are used to send the synchronization beam, then, after the synchro-laser, Alice waits 200 ns and then sends two bits separated by 200 ns. The resulting raw key rate is therefore upper bounded as $R_{\text{raw}} \leq 2.39\,\text{Mbit/s}$.

The measured sifted key rate R_{sift} allows to estimate the total loss along the source-channel-detector chain, $\alpha = R_{\text{sift}}/R_{\text{raw}}$. This includes also the fraction of pulses that carry no photons, due to the Poissonian statistics of the faint source, and the B92 protocol efficiency $\eta_{\text{B92}} = 1/4$.

4.2.2 Attack model

We consider selective individual attacks, where Eve measures each photon independently with probability $0 < q < 1$, using either basis, $\mathbb{X}$ or $\mathbb{Z}$, randomly chosen. In the *Intercept-and-Resend (IR)* attack [146] (see §3.4.1 for a description of the attack as applied to the BB84 protocol), each measured bit is resent with the same encoding as used by Alice, thus increasing the error rate at Bob. In particular, observe that by considering Alice and Bob's sifted keys as input and output, respectively, the quantum channel can be modeled as a binary symmetric channel (BSC) with some error probability Q. When a single qubit is observed by Eve according to an IR attack, the error probability at Bob for the corresponding bit is set to 1/4 due to the random and independent choice of the basis used by Alice and Eve. More precisely, it was shown in [84] that 1/4 is a lower bound on the error probability induced by the IR attack, for any basis chosen by the eavesdropper to measure the incoming qubits and resend them to Bob. Hence, an individual IR attack with probability q increases the quantum bit error rate (QBER) value to

$$Q' = (1-q)Q + q\frac{1}{4} = Q + q\left(\frac{1}{4} - Q\right), \tag{4.2}$$

whereas it is conservatively assumed to leave channel losses unaffected.

On the other hand, in the *unambiguous state discrimination (USD)* attack [85] only the 0's that are measured with the $\mathbb{Z}$ basis and the 1's that are measured with the $\mathbb{X}$ basis

are retransmitted to Bob, thereby introducing further losses at the legitimate receiver but no additional errors. When a qubit is observed by Eve and resent according to the USD scheme, the random choice of the basis introduces a further loss factor of 1/4. Hence, individual USD attacks with probability q increase channel losses to the value

$$\alpha' = (1-q)\alpha + q\frac{\alpha}{4} = \alpha - \frac{3}{4}q\alpha. \tag{4.3}$$

We also consider the *photon number splitting (PNS)* attack [67]. In this case, qubits carried by two or more photons might be observed by Eve without introducing any effect at Bob's receiver. However, since this attack can only be successfully carried out on multiple photon qubits, the probability that one bit of the sifted key is observed by the adversary is upper bounded by

$$q_{\mathrm{PNS}} \leq P[n_{\mathrm{ph}} > 1 | n_{\mathrm{ph}} > 0] = \frac{1 - e^{-\mu}(1+\mu)}{1 - e^{-\mu}}. \tag{4.4}$$

where n_{ph} is the number of photons in a generic bit at the transmitter output, that is Poisson distributed with mean μ.

Eventually, while considering the above attacks in the design of privacy amplification, we upper bound the amount of information available the eavesdropper with what she would get by correctly detecting all the observed qubits. As for the PNS attack, we observe that our estimate is more conservative as compared with the one obtainable by using decoy states, which allow to precisely estimate the number of single photon pulses (see [50, Section IV.B]). In particular, in the context of selective individual attacks, our technique yields a lower secret key rate, though relying on a simplified hardware setup.

4.2.3 Channel estimation

In each round of the key-agreement protocol, Bob sends the positions of the received qubits over the public channel and discloses the value of a fraction of them, in order to allow the transmitter to estimate the channel losses and the QBER. The objective of channel estimation is twofold: it predicts losses and the error rate introduced by the noisy quantum channel in order to properly perform the *key reconciliation* stage. Moreover, it is used to reveal the presence of an eavesdropper,that is, to determine the probability q that a photon has been observed by Eve, according to the attack schemes described in Section 4.2.2. A misdetection probability lower than P_{miss} is assured, where the misdetection event represents the case in which Eve is observing on average more photons than the number predicted by the channel estimation protocol.

The QBER is estimated at each round by randomly choosing N_{qber} bits from the sifted key to be disclosed over the public channel. Then, the maximum likelihood (ML) estimate of Q is simply defined as

$$Q_{\mathrm{ML}} = \frac{1}{N_{\mathrm{qber}}} \sum_{i=1}^{N_{\mathrm{qber}}} e_i, \tag{4.5}$$

where $e_i = 1$ if there is an error in the corresponding bit of the publicly disclosed portion of the sifted key, and $e_i = 0$ otherwise. Then, the estimator Q_{ML} is a random variable that exhibits a different statistical description conditioned on the fact that an adversary is implementing the IR attack or not. More in details, the mean and standard deviation of Q_{ML} are given by

$$m_{\mathrm{ML}} = Q \quad , \quad \sigma_{\mathrm{ML}} = \sqrt{\frac{Q(1-Q)}{N_{\mathrm{qber}}}}. \tag{4.6}$$

when the photons sent by Alice are not measured by any eavesdropper, whereas

$$m'_{\mathrm{ML}} = Q' \quad , \quad \sigma'_{\mathrm{ML}} = \sqrt{\frac{Q'(1-Q')}{N_{\mathrm{qber}}}}. \tag{4.7}$$

when the system is subject to an IR attack. In order to be able to reveal the presence of an eavesdropper that is carrying IR attacks with a given probability q_{IR}, the legitimate parties guarantee that the number of qubits used for QBER estimation is high enough to discriminate the case in which the BSC error probability is Q or Q'. In other words, it must hold

$$m_{\mathrm{ML}} + \beta\sigma_{\mathrm{ML}} < m'_{\mathrm{ML}} - \beta\sigma'_{\mathrm{ML}}, \tag{4.8}$$

with β denoting an appropriate multiplicative factor that determines the confidence interval of the QBER estimate. For the sake of tractability, we approximate the random variable Q_{ML} with a Gaussian random variable with same mean and same standard deviation. Then, it is possible to guarantee a miss detection probability up to, e.g., $P_{\mathrm{miss}} = 5 \cdot 10^{-3}$ by imposing

$$\beta = Q^{-1}(P_{\mathrm{miss}}) \approx 2.6, \tag{4.9}$$

with $Q^{-1}(\cdot)$ denoting the inverse of the Q-function, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du$. Then, on substituting (4.2), (4.6) and (4.7) in (4.8), after simple algebraic manipulations, it is possible to determine the maximal undetectable IR attack probability as a function of the parameter N_{qber}, namely

$$q_{\mathrm{IR}} = \frac{\left(\frac{1}{4} - Q\right) \cdot \left(2\beta\sqrt{\frac{Q(1-Q)}{N_{\mathrm{qber}}}} + \frac{\beta^2}{N_{\mathrm{qber}}}\right) + \frac{\beta^2}{N_{\mathrm{qber}}}\left(2Q^2 - \frac{Q}{2}\right)}{\left(\frac{1}{4} - Q\right)^2 - \frac{\beta^2}{N_{\mathrm{qber}}}\left(\frac{Q}{2} - Q^2 - \frac{1}{16}\right)}. \tag{4.10}$$

On assuming that all the errors introduced by the quantum channel are corrected during the key reconciliation phase, the QBER estimate can also be refined at Bob by counting the number of bits that are flipped after reconciliation. In this way, it is possible to decrease the maximal undetectable IR attack probability to the value obtained by substituting N_{qber} with N_{sift} in (4.10).

Analogously, channel losses are estimated by counting all Bob's sifted bits. Similarly to the case for the QBER estimation, the ML estimator for channel losses is obtained as

$$\alpha_{\mathrm{ML}} = \frac{1}{N_{\mathrm{raw}}} \sum_{i=1}^{N_{\mathrm{raw}}} a_i, \tag{4.11}$$

where $a_i = 1$ for the indexes corresponding to bits in the raw key that made Bob's detectors click, and $a_i = 0$ otherwise. Again, the channel losses estimator α_{ML} is a random variable with mean and standard deviation given by

$$m_{\mathrm{ML}} = \alpha \quad , \quad \sigma_{\mathrm{ML}} = \sqrt{\frac{\alpha(1-\alpha)}{N_{\mathrm{raw}}}}, \tag{4.12}$$

or

$$m'_{\mathrm{ML}} = \alpha' \quad , \quad \sigma'_{\mathrm{ML}} = \sqrt{\frac{\alpha'(1-\alpha')}{N_{\mathrm{raw}}}}, \tag{4.13}$$

depending on the presence of an eavesdropper carrying a USD attack. By following closely similar steps to those used to determine q_{IR} in (4.10), the maximal undetectable USD attack probability can be found as

$$q_{\mathrm{USD}} = \frac{\frac{3}{2}\beta\left(\sqrt{\frac{\alpha(1-\alpha)}{N_{\mathrm{raw}}}} - \left(\frac{1}{2}-\alpha\right)\frac{\beta}{N_{\mathrm{raw}}}\right)}{\frac{9}{16}\alpha\left(1+\frac{\beta^2}{N_{\mathrm{raw}}}\right)}. \tag{4.14}$$

4.2.4 Classical processing

Errors introduced on the sifted key by the quantum channel (polarization degradation due to the atmosphere, noise in the devices, etc) are corrected by implementing the Winnow scheme (see §3.5.2). The probability of a reconciliation failure is kept below a fixed value P_{fail}, by guaranteeing a residual BER on the reconciled key smaller than $P_{\mathrm{fail}}/N_{\mathrm{rec}}$, where $N_{\mathrm{rec}} = N_{\mathrm{sift}} - N_{\mathrm{qber}}$ is the number of sifted . Given these constraints, the number of iterations of the protocol and the block sizes for parity checking are chosen to minimize the number of bits N_{rev} revealed over the public channel, as detailed in §3.5.2.

The reconciled keys at Alice and Bob are compressed following a two steps procedure aiming at reducing the information leakage to Eve. First, N_{rev} out of the $N_{\mathrm{rec}} = N_{\mathrm{sift}} - N_{\mathrm{qber}}$ bits of the reconciled key are deleted according to the procedure of *bit deletion* described in §3.6.1 and in [97, Section 3], in order to eliminate the information revealed over the public channel to perform key reconciliation. In this way, the information leaked to the eavesdropper during the key reconciliation stage is reduced exactly to zero (see §3.6.1 for a discussion).

After bit deletion, privacy amplification is obtained by hashing with a full column rank, random, binary Toeplitz matrix [124], renewed at each round. The number of rows in the Toeplitz matrix, that is, the final secret key length, is a design parameter for this phase of the key processing, which depends on the amount of information on the key that Eve is estimated to have gathered during the previous stages of the protocol.

For instance, the system described in [147] complies with two different methods in estimating the information gathered by Eve with IR attacks. One method is due to Bennett [84] and it links the number of errors revealed after key reconciliation with the information gained by the eavesdropper. More precisely, on denoting with e the total number of errors revealed on the sifted key, the information leaked to the eavesdropper is

approximated by the value

$$\frac{4e}{\sqrt{2}} + \beta\sqrt{(4+\sqrt{2})e}, \tag{4.15}$$

where the confidence margin was chosen as $\beta = 5$. A second estimate, provided in [65] defines an upper bound on Eve's Renyi entropy in the limit of long transmissions, that is when $N_{\text{sift}} \to \infty$. Then, the information leakage due to multiphoton pulses is handled separately, and added to the previous quantity.

On the other hand, in our experiment, according to the three attack models and the channel estimation scheme described in the previous sections, each bit in the reconciled key is assumed to have been observed by the eavesdropper with probability not larger than $q_{\text{tot}} = q_{\text{IR}} + q_{\text{USD}} + q_{\text{PNS}}$, independently of all the others. Then Eve's Rényi information on the reconciled key is a binomially distributed random variable $t \sim \mathcal{B}(N_{\text{rec}} - N_{\text{rev}}, q_{\text{tot}})$, and we can use the results in [21] to determine a probabilistic upper bound on the information I_{leak} leaked to Eve after privacy amplification. In fact, with probability at least $1 - P_{\text{miss}}$, it is

$$I_{\text{leak}} \leq I_{\text{tar}}(N_{\text{sec}}, b) = N_{\text{sec}} P[t > b] + \frac{1}{2^{(N_{\text{rec}} - N_{\text{rev}} - N_{\text{sec}} - b)} \ln 2}, \tag{4.16}$$

for any value of b, and with N_{sec} denoting the length of the secure key at the output of the privacy amplification stage. Under the constraint that $I_{\text{leak}} < \theta_{\text{sec}}$, the secure key rate is thus maximized by choosing

$$N_{\text{sec}} = \max\left\{ a \,:\, \min_{b} I_{\text{tar}}(a, b) \leq \theta_{\text{sec}} \right\}. \tag{4.17}$$

This result has a similar structure as compared with the one stated in theorem 7. Nevertheless, a few comments are to be made. First, theorem 7 considers only IR attacks, whereas here we are taking into account also USD and PNS attacks. Second, in theorem 7 an upper bound to the tolerable error rate on the eavesdropping basis is fixed, whereas here we estimate the attack rate based on a Gaussian approximation of induced error rates and losses, that is, we introduce a further assumption on their distribution. However, theorem 7 could be further generalized to the considered case, so that an upper bound on the tolerable losses is set.

4.2.5 Authentication and transmission over the public channel

In our prototype, communication on the public discussion channel between Alice and Bob is implemented with user datagram protocol (UDP) over IP, and by means of 802.11g wireless transmissions. Therefore no security services are leveraged other than the unconditionally secure authentication we provide at the application layer.

The concatenation of all messages transmitted by a terminal in a protocol round is hashed by means of a keyed function to a 100 bit tag, which is then XORed with a one time pad (OTP). The hash function is chosen from the Stinson ε-almost strongly universal$_2$ class [143], and is renewed every 25 rounds. The hashing key and the OTP altogether require 250 secure bits per round, that are taken from the previously generated keys, thus lowering the *net key rate*. More details on Stinson's authentication can be found in [141].

Transmission parameters	
packet rate	$R_{\text{ptk}} = 12.5\,\text{pkt/s}$
raw key rate	$R_{\text{raw}} = 72\,\text{kbit/s}$
Channel parameters	
overall loss rate	$\alpha = 6.4 \cdot 10^{-2}$
quantum bit error rate	$\epsilon = 2.1 \cdot 10^{-2}$
sifted key rate	$R_{\text{sift}} = 4.6\,\text{kbit/s}$
undetected eavesdropper rate	$q_{\text{tot}} < 0.41$
Security parameters	
secret key rate	$R_{\text{sk}} = 600\,\text{bit/s}$
prob. of failed reconciliation	$P_{\text{fail}} < 0.02$
information leakage rate	$R_{\text{leak}} \leq 0.2\,\text{kbit/s}$
prob. of higher leakage	$P_{\text{miss}} < 5 \cdot 10^{-3}$

Tabella 4.2: Performance measurements at the Palazzo della Ragione experiment.

4.2.6 Experimental results

Our prototype was publicly demonstrated on October 3-4, 2011 at Palazzo della Ragione in Padua with indoor daylight conditions over a 50 m free-space link along the south wall of the Great Hall. It was kept up and running for 5 hours on October 3rd, and for 8 hours on October 4th. Along with the key agreement procedure the two terminals carried out the secure exchange text messages and images provided by guests and visitors over a wireless radio link. The distilled secure keys were used for OTP encryption and decryption at the transmitted and receiver side, respectively. As for the communication of this application, we employed the transport control protocol (TCP) over Internet protocol (IP) and IEEE 802.11g wireless transmission.

The measured performance parameters for the QKD system in the setting are summarized in Tab. 4.2. These results are derived while imposing $\theta_{\text{sec}} = 1$ bit, i.e., by requiring that the adversary knows only 1 bit per key (the average measured secret key length was of approximately 2400 bits). However, according to the operational interpretation provided in lemma 1 and in lemma 2, this value should be significantly lowered, so that $\theta_{\text{sec}} \ll 1$.

4.3 Efficient BB84 in free-space with finite-key analysis

In this section we propose the results which were recently published in [J1]. More precisely, we report the analysis and the experimental results obtained for a QKD system based on the efficient BB84 protocol (see §3.4.2). Based on the results presented in [74], this work is inspired by the need for proving the experimental feasibility of quantum key distribution with finite-length keys and investigates the required number of qubits as a function of the key size and of the ambient quantum bit error rate. The experiment, in fact, is performed in different channel conditions, and assuming two distinct attack models: individual attacks or general quantum attacks. The results indicate that viable conditions for effective symmetric, and even one-time-pad, cryptography are achievable.

4.3.1 Experimental setup

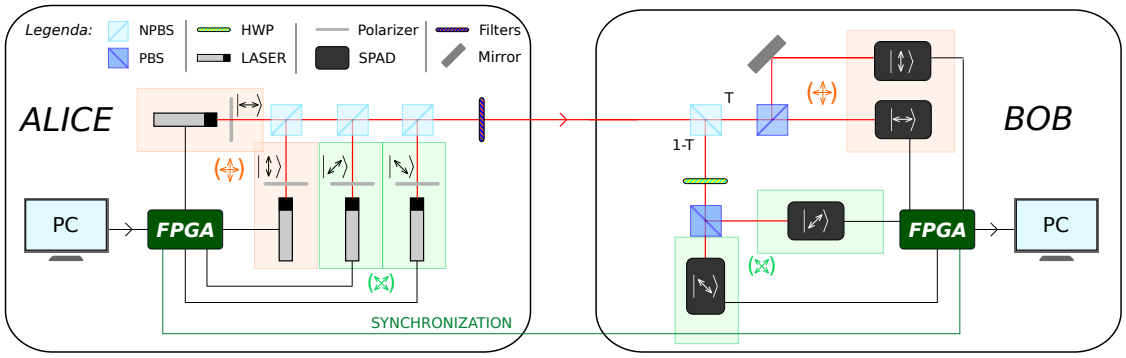


Figura 4.6: **Schematic of the experimental setup**. The qubits are generated by attenuating four differently polarized lasers. The FPGA board controls which laser should be turned on in each qubit transmission. At the receiver side, by a beam splitter with transitivity T, Bob perform the measurement in the $\mathbb{X}$ (with probability T) or $\mathbb{Z}$ basis (with probability $1-T$). NPBS, beam splitter; PBS, polarizing beam splitter; HWP, half wave plate; Filters, neutral density filters, SPAD, single photon avalanche diode.

The optical setup of our prototype implementing the quantum communication is shown in Fig. 4.6. The transmitter (Alice) uses four infrared (850nm) attenuated diode lasers driven by a Field Programmable Gate Array (FPGA) to send the bits 0 and 1 encoded in the different polarization bases of the photons. By properly configuring the FPGA, it is possible to set the probabilities $p_{\mathbb{X}}$ and $p_{\mathbb{Z}}$. The receiver (Bob) uses a variable beam splitter (BS) with transmission T to send the received qubits to the measures in the two bases. The probability $p_{\mathbb{X}}$ is equal to the transmissivity T of the BS. On one BS output, a polarizing beam splitter (PBS) and two single photon avalanche photodiodes (SPAD) measure the photons in the $\mathbb{X}$ basis; on the other side a half-wave plate (HWP) is positioned before the PBS to allow the measurement in the $\mathbb{Z}$ basis. The counts detected by the four SPAD are stored on a second FPGA. A cable between the two FPGAs is also used along for synchronization.

As for the transmitted qubits, we used the same data structure of [C3], that is, the one shown in figure 4.5 and described in §4.2.1, with the only difference that a raw key is now composed by N packets of 2880 bits each. It is worth noting that the experimental setup of this protocol is very similar to the original BB84: the main difference lies in the interpretation of received bits in the two different bases.

4.3.2 Classical post-processing

As described in chapter 3, after the quantum transmission and the sifting of the raw data, four subsequent tasks take place: parameters estimation, information reconciliation, error verification and privacy amplification.

The first task, parameters estimation, is required to measure the quantum bit error rate (QBER) on the $\mathbb{Z}$ basis, $Q_{\mathbb{Z}}$. In fact, we here assume that the quantum channel is stable, i.e., that QBER on the $\mathbb{X}$-basis, $Q_{\mathbb{X}}$, is constant in time (note that, in general,

$Q_{\mathbb{X}} \neq Q_{\mathbb{Z}}$) and does not need to be estimated at each protocol run. It should be noted that this assumption does not affect the security of the scheme, as if $Q_{\mathbb{X}}$ increases (for instance because an attacker is tampering with the channel), then the information reconciliation will fail. The failure will be detected during the error verification phase, and the protocol will abort. On the other hand, the empirical QBER in the $\mathbb{Z}$ basis is dynamically computed at each protocol run according to equation (3.40), in order to check for the presence of an eavesdropper. The protocol aborts if $\hat{Q}_{\mathbb{Z}} > Q^{\mathbb{Z}}_{\text{tol}}$, where $Q^{\mathbb{Z}}_{\text{tol}}$ is a given channel error tolerance on the $\mathbb{Z}$ basis which has been determined a priori based on the expected behavior of the quantum channel and the required level of security.[7] The probability that the protocol aborts is denoted by p_{abort}.

After the parameter estimation phase, information reconciliation is performed. We recall that information reconciliation (according to the direct reconciliation approach, §3.5) aims at correcting the discrepancies between $\mathbf{X}$ and $\mathbf{X}'$ that the channel may have introduced, thus allowing Bob to compute an estimate $\hat{\mathbf{X}}$ of $\mathbf{X}$. As a practical solution, we have chosen the Winnow scheme (see §3.5.2) which, by leveraging Hamming codes of different lengths over multiple iterations, allows an adaptive and lowly interactive error correction and represents a good trade-off between the high interactivity required by CASCADE and the low flexibility of LDPC code with limited key length.[8] We fix an upper bound P_{fail} to the probability of a reconciliation failure and, under this constraint, we optimize the parameters of the Winnow scheme in order to minimize the expected (average) classical information leakage $\mathbb{E}[L_{\text{EC}}]$. First, given the average QBER on the $\mathbb{X}$ basis $Q_{\mathbb{X}}$, a threshold $Q^{\mathbb{X}}_{\max} > Q_{\mathbb{X}}$ is fixed so that the empirical QBER $\hat{Q}_{\mathbb{X}}$ in the sifted key is higher than $Q^{\mathbb{X}}_{\max}$ with probability less than $P_{\text{fail}}/2$. Then, the block sizes are chosen so that the output BER is lower than $P_{\text{fail}}/(2n)$ whenever $\hat{Q}_{\mathbb{X}} < Q^{\mathbb{X}}_{\max}$ and $\mathbb{E}[L_{\text{EC}}]$ is minimized, as detailed in [C3].

Subsequently, an error verification mechanism such as the one proposed in [74] and described in §3.5.4 ensures that the protocol is ε_{cor}-correct, i.e., that $P[\mathbf{X} \neq \hat{\mathbf{X}}] < \varepsilon_{\text{cor}}$, by comparing hashes of $(\lceil \log_2(P_{\text{fail}}/\varepsilon_{\text{cor}}) \rceil)$ bits. Namely, Alice chooses the hash function g randomly and uniformly from a class of universal$_2$ hash functions [34] (the class of Toeplitz matrices in our experimental setup) and computes her hash value $g_{\text{A}} = g(\mathbf{X})$. She then sends g_{A} and a compact representation of g to Bob, who computes $g_{\text{B}} = g(\hat{\mathbf{X}})$. The protocol aborts if the two hashes are different, i.e., if $g_{\text{A}} \neq g_{\text{B}}$.

Finally, during the so-called privacy amplification, $\mathbf{X}$ and $\hat{\mathbf{X}}$ are compressed by means of a function which is, again, randomly and uniformly chosen from a class of universal$_2$ hash functions, in order to get the final secret keys $\mathbf{S}$ and $\hat{\mathbf{S}}$. The length ℓ of the final key and the corresponding amount of compression depend on the required level of secrecy, on the overall classical information leakage $L_{\text{EC}} + \lceil \log_2(P_{\text{fail}}/\varepsilon_{\text{cor}}) \rceil$, on the assumed attacker's model and on the estimate of the information leaked to the eavesdropper during the transmission over the quantum channel.

[7] The $\mathbb{Z}$-basis error tolerance, $Q^{\mathbb{Z}}_{\text{tol}}$, is in fact optimized for maximizing the final secret key rate, as a function of the expected channel behavior and of the required level of security, captured in the following by the parameter ε_{sec}.

[8] see remark at the end of this section.

Remarks on the choice of the Winnow protocol. As for the choice of the Winnow scheme, let us make the following observations, which were partially introduced in §3.5. In the low and moderate key length regime, as opposed to the asymptotic limit, the value of the QBER in a single key exhibits random fluctuations around its average value, especially in realistic experimental conditions, as in the scenario we are here considering. In this case, the error reconciliation protocol must be designed with some margin on the maximum QBER it can correct, with respect to the average QBER value (the choice of $Q^{\mathbb{X}}_{\max} > Q_{\mathbb{X}}$ in our scheme). However, forward error correction schemes such as LDPC, BCH or Reed-Solomon always disclose the same number of bits over the public channel, even for those keys in which the QBER is much lower than the design value. As an approximation in [74] the reference value of $r = 1.1\ n\ h_2(Q^{\mathbb{X}}_{\max})$ bit is used, but it should be noted that such value (10% over the asymptotic Shannon limit) is only achieved by LDPC codes with very long keys (for instance $n = 10^6$, in [98]). On the contrary, interactive protocols such as Cascade or Winnow have the advantage of intrinsically adjusting to the QBER of each single key, since, for blocks that have no errors, only a single parity check bit is disclosed over the public channel, and, therefore, they are particularly appropriate in the low QBER regime. Observe, also, that Winnow, while yielding approximately the same performance as Cascade in terms of residual error rate and number of disclosed bits (see §3.5.2), requires less rounds of interactive communication between Alice and Bob, since it does not implement $\log_2(L)$ rounds of a binary search to locate the bit errors in length L blocks. That said, and considering that the classical channel used for the post-processing is much faster with respect to the quantum one, the Winnow scheme is suitable for the application scenario we consider.

4.3.3 Finite-length secret key rate

We here consider two possible attack models, that is, general and intercept-and-resend attacks. Correspondingly, we use the security definitions 23 and 22.

According to definition 23, general secrecy requires that the final shared keys are secret with respect to the most general quantum attacks, and it is based on the secrecy criterion provided in [70] and recalled in (3.21). In particular, since we now consider a protocol which possibly aborts with probability p_{abort}, we should rewrite (3.21) as

$$\frac{1}{2}\|\rho_{SE} - \rho_U \otimes \rho_E\|_1 \leq \frac{\varepsilon_{\text{sec}}}{1 - p_{\text{abort}}}. \tag{4.18}$$

Then, if the bases $\mathbb{X}$ and $\mathbb{Z}$ are chosen as described above and assuming that Alice uses an ideal single photon source, the authors of [74] show that an ε_{sec}-GS key can be extracted out of the reconciled key, with length

$$\ell \leq n(1 - \widetilde{h}_2(Q^{\mathbb{Z}}_{\text{tol}} + \mu)) - L_{\text{EC}} - \log_2 \frac{2P_{\text{fail}}}{\varepsilon^2_{\text{sec}}\varepsilon_{\text{cor}}} \tag{4.19}$$

where $\mu = \sqrt{\frac{n+k}{nk}\frac{k+1}{k}\ln\frac{2}{\varepsilon_{\text{sec}}}}$, $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary Shannon entropy function, $\widetilde{h}_2(x) = h_2(x)$ for $0 \leq x \leq 0.5$ and $\widetilde{h}_2(x) = 1$ for $x > 0.5$.

On the other hand, pragmatic secrecy (definition 22) ensures that the final key is secret with respect to intercept-and-resend (IR) attacks [146], i.e., a specific class of selective individual attacks, which, however, represents the most realistic and feasible attack strategy based on the experimental technology nowadays available: collective or more general attack models (see [51]), in fact, require ancillary qubits and quantum memories in order to be deployed. Again, since the protocol may abort with probability p_{abort}, we can rewrite (3.17) as

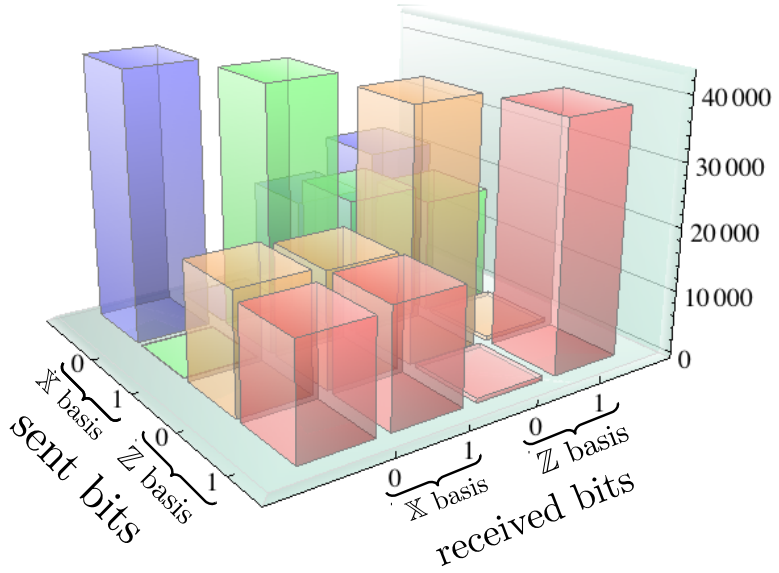


Figura 4.7: **Experimental bits.** Joint empirical distribution of sent and received bits, as obtained in one experiment with the best channel conditions (corresponding to $Q_{\mathbb{X}} = 0.33\%$ and $Q_{\mathbb{Z}} = 1.48\%$). The probabilities of sending and measuring in the $\mathbb{X}$ and $\mathbb{Z}$ basis were $p_{\mathbb{X}} = 0.51$ and $p_{\mathbb{Z}} = 0.49$, respectively.

$$H(\mathbf{U_S}) - H(S|V) \leq \frac{\delta_{\text{sec}}}{1 - p_{\text{abort}}} \tag{4.20}$$

Now, the pragmatic security of the distilled key can be assessed through theorem 7, by substituting $q_{\mathbf{X}} = 1$, $q_{\mathbf{Z}} = 1/2$ and by setting $n_{\text{EC}} = n - L_{\text{EC}} - \lceil \log_2(P_{\text{fail}}/\varepsilon_{\text{cor}}) \rceil$. Also, based on (3.116), we can choose the optimal secret key length as

$$\ell = \max \left\{ b : \min_a f(a, b) \leq \delta_{\text{sec}} \right\} \tag{4.21}$$

Please note that, in order to allow a comparison with the tight bound (4.19), we have derived the secure key length in the hypothesis that Alice uses a single photon source.

Finally, given the probability ε_{rob} that the protocol aborts even if the eavesdropper is inactive [74], we can compute the final raw to secret key rate for both general and pragmatic secrecy as

$$r(\ell, n, k, \varepsilon_{\text{rob}}) = (1 - \varepsilon_{\text{rob}}) \frac{\ell}{M(n, k)} \tag{4.22}$$

where $M(n, k) = n + k + 2\sqrt{nk}$ is the expected number of qubits that have to be sent until n sifted key bits and k parameter estimation bits are collected.

4.3.4 Experimental results

We conducted experiments with different noisy channels yielding different values for the average QBERs $Q_{\mathbb{X}}$ and $Q_{\mathbb{Z}}$, each of them realized with different encoding probabilities $(p_{\mathbb{Z}}, p_{\mathbb{X}})$. We varied the noise value in the channel by coupling to the receiver an external unpolarized source of suitable intensity, that increased the background signal. It is worth noting that by this operation we are modelling the following depolarizing channel

$$\mathcal{C} : \rho \to (1 - P)\rho + \frac{P}{4} \sum_{j=0}^{3} \sigma_j \rho \sigma_j, \tag{4.23}$$

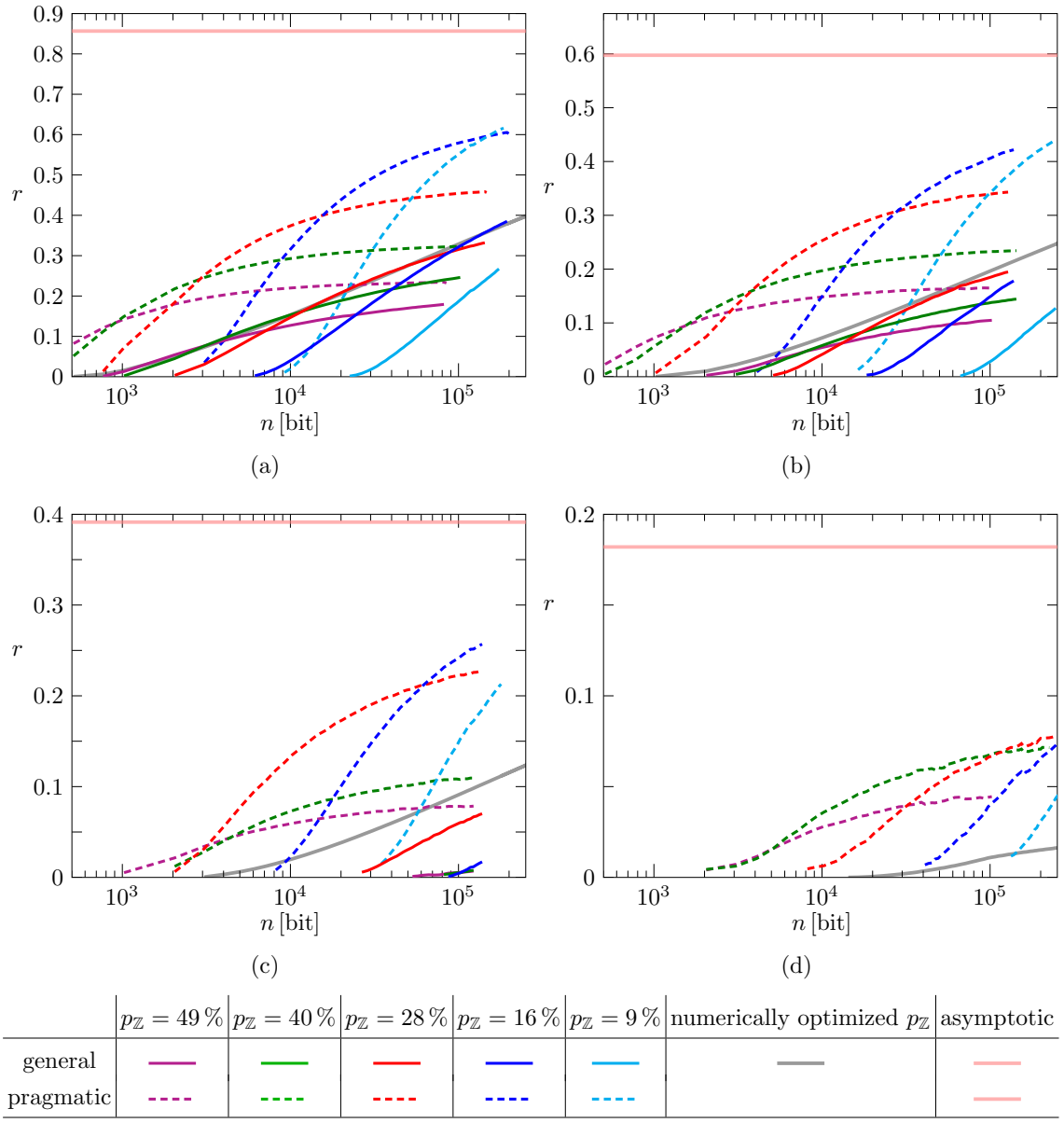


Figura 4.8: **Experimental key rates.** Experimental secret key rates r vs. sifted key length n for different probabilities of encoding and measuring on the two bases $p_{\mathbb{Z}}$, $p_{\mathbb{X}} = 1 - p_{\mathbb{Z}}$ and for different channel conditions (values of the average QBERs $Q_{\mathbb{X}}, Q_{\mathbb{Z}}$): (a) $Q_{\mathbb{X}} = 0.3\,\%$, $Q_{\mathbb{Z}} = 1.5\,\%$; (b) $Q_{\mathbb{X}} = 2.4\,\%$, $Q_{\mathbb{Z}} = 3.9\,\%$; (c) $Q_{\mathbb{X}} = 4.9\,\%$, $Q_{\mathbb{Z}} = 6.0\,\%$; (d) $Q_{\mathbb{X}} = 8.3\,\%$, $Q_{\mathbb{Z}} = 8.1\,\%$. For each case we report the key rates obtained for ε_{sec}-GS (solid lines) and δ_{sec}-PS (dashed lines) keys with $\varepsilon_{\text{sec}} = 10^{-10}$, $\delta_{\text{sec}} = \frac{2}{\ln 2}\varepsilon_{\text{sec}}^2$, $P_{\text{fail}} = 10^{-3}$ and a correctness parameter $\varepsilon_{\text{cor}} = 10^{-10}$. The standard deviation of experimental rates are on the order of 10^{-3} for both ε_{sec}-GS and δ_{sec}-PS keys. Error bars are not reported in the plot for the sake of clarity. For comparison, we also report the asymptotic key rate in the infinite length limit, and the ε_{sec}-GS bound achievable by optimizing the probability $p_{\mathbb{Z}}$ and the thresholds $Q_{\text{tol}}^{\mathbb{Z}}, Q_{\text{max}}^{\mathbb{X}}$ for each value of n.

where σ_j are the Pauli matrices, being σ_0 the identity and P the parameter representing the probability that any detected photon is coming from the background.

In figure 4.7 we show the joint empirical distribution of the transmitted and received bits on the $\mathbb{X}$ and $\mathbb{Z}$ bases obtained in one run with the best environmental conditions (i.e., with additional background), for the case $p_{\mathbb{Z}} = 49\%$ and $p_{\mathbb{X}} = 51\%$. As expected, in this case the QBER is very low: the main source of errors are imperfections in the waveplates used in the measurement, yielding $Q_{\mathbb{X}} = 0.33\%$ and $Q_{\mathbb{Z}} = 1.48\%$ on average.

In Figure 4.8 we show the measured experimental key rates for each data set and for both general and pragmatic secrecy. First of all, let us recall that, in order to consistently compare the secrecy rates obtained with general and pragmatic secrecy, the security parameters ε_{sec} and δ_{sec} have to be chosen so that $\delta_{\text{sec}} = \frac{2}{\ln 2}\varepsilon_{\text{sec}}^2$, as shown in proposition 8. As a performance reference, we plot the asymptotic theoretical bound $r = 1 - h_2(Q_{\mathbb{X}}) - h_2(Q_{\mathbb{Z}})$, holding in the limit of infinite length keys (labelled as "asymptotic" in Fig. 4.8) and the optimal theoretical bound for ε_{sec}-GS keys (labelled as "numerically optimized $p_{\mathbb{Z}}$" in Fig. 4.8). The experimental key rates are obtained by the following procedure: for each data set the n-bit sifted key $\mathbf{X}$ and the k-bit parameter estimation string $\mathbf{Z}$ ($\mathbf{X}'$ and $\mathbf{Z}'$) at Alice's (Bob's) side are obtained by the experiment. The error correction is performed on $\mathbf{X}$ and $\mathbf{X}'$ by using the Winnow scheme; in particular, the Winnow parameters were chosen so that a maximum of 6 subsequent iterations is allowed with block sizes up to 256 bits. We then performed privacy amplification by compressing the error-free keys by multiplication with a random binary Toeplitz matrix. The amount of compression depends on ℓ, the secret key length, given by Eq. (4.19) and (4.21) for general and pragmatic security, respectively. On the other hand, the optimal bound for ε_{sec}-GS keys is numerically derived by maximizing the secret key rate r (Eq. (4.22), with ℓ given by Eq. (4.19)) over p_Z, $Q_{\text{tol}}^{\mathbb{Z}}$ and $Q_{\text{max}}^{\mathbb{X}}$ for each n.

In the numerical procedure used to find the optimal bound for ε_{sec}-GS keys, since an analytical expression is not available for L_{EC} or ε_{rob}, L_{EC} is approximated as $L_{\text{EC}} = 1.1 \cdot n \cdot h_2(Q_{\mathbb{X}})$ and, similarly, ε_{rob} is replaced by the following upper bound (see equation A5 of ref. [122] for details):

$$\varepsilon_{\text{rob}} \leq \exp\left[-\frac{k(Q_{\text{tol}}^{\mathbb{Z}} - Q_{\mathbb{Z}})^2}{1 - 2Q_{\mathbb{Z}}} \ln\left(\frac{1 - \mathrm{Q}_{\mathbb{Z}}}{\mathrm{Q}_{\mathbb{Z}}}\right)\right]. \tag{4.24}$$

Experimental values obtained for ε_{rob} show that such bound is rather loose. On the other hand, as $Q_{\mathbb{X}}$ increases, the approximate expression for L_{EC} is lower than the average value for the Winnow scheme. As a consequence, the experimental secret key rates may slightly exceed the optimal bound in some low QBER cases, as we can see in fig. 4.8a.

As a further comment, we note that, for an asymmetric channel with $Q_{\mathbb{X}} < Q_{\mathbb{Z}}$, using the $\mathbb{Z}$ basis for key encoding and $\mathbb{X}$ for eavesdropper detection provides a higher optimal secret key rate (4.22). However, when the two error rates $Q_{\mathbb{X}}$ and $Q_{\mathbb{Z}}$ have similar values, a minor gain in r is obtained. For instance, when $n = 10^6$, $\varepsilon_{\text{cor}} = \varepsilon_{\text{sec}} = 10^{-10}$, with $Q_{\mathbb{Z}} = 4\%$ and $Q_{\mathbb{X}} = 2\%$, we can achieve $r = 0.31$; by exchanging the role of $\mathbb{Z}$ and $\mathbb{X}$, $r = 0.33$ can be achieved.

In situations such as satellite quantum communications, the amount of sifted bits is expected to fluctuate as it depends on the variable channel conditions during the passage. From the experimental point of view it is easier to fix the values of $p_{\mathbb{Z}}$ and $p_{\mathbb{X}}$ and accumulate data as long as possible. The value of $p_{\mathbb{X}}$ will constrain the ratio between k and n according to the relation $p_{\mathbb{X}} = \frac{1}{1+\sqrt{k/n}}$. In the performed experiments, we thus fixed the value of $p_{\mathbb{Z}}$ and $p_{\mathbb{X}} = 1 - p_{\mathbb{Z}}$. For each value of the background noise we run different acquisitions with $p_{\mathbb{Z}}$ belonging to the discrete set $\{9\%, 16\%, 28\%, 40\%, 49\%\}$.

Experimental results for the ε_{sec}-GS key rates are plotted with thin solid lines, while δ_{sec}-PS key rates are plotted with thin dashed lines; different colors correspond to different $(p_{\mathbb{Z}}, p_{\mathbb{X}})$. We used $P_{\text{fail}} = 10^{-3}$, $\varepsilon_{\text{cor}} = 10^{-10}$ and $\varepsilon_{\text{sec}} = 10^{-10}$. As expected, pragmatic secrecy always allows

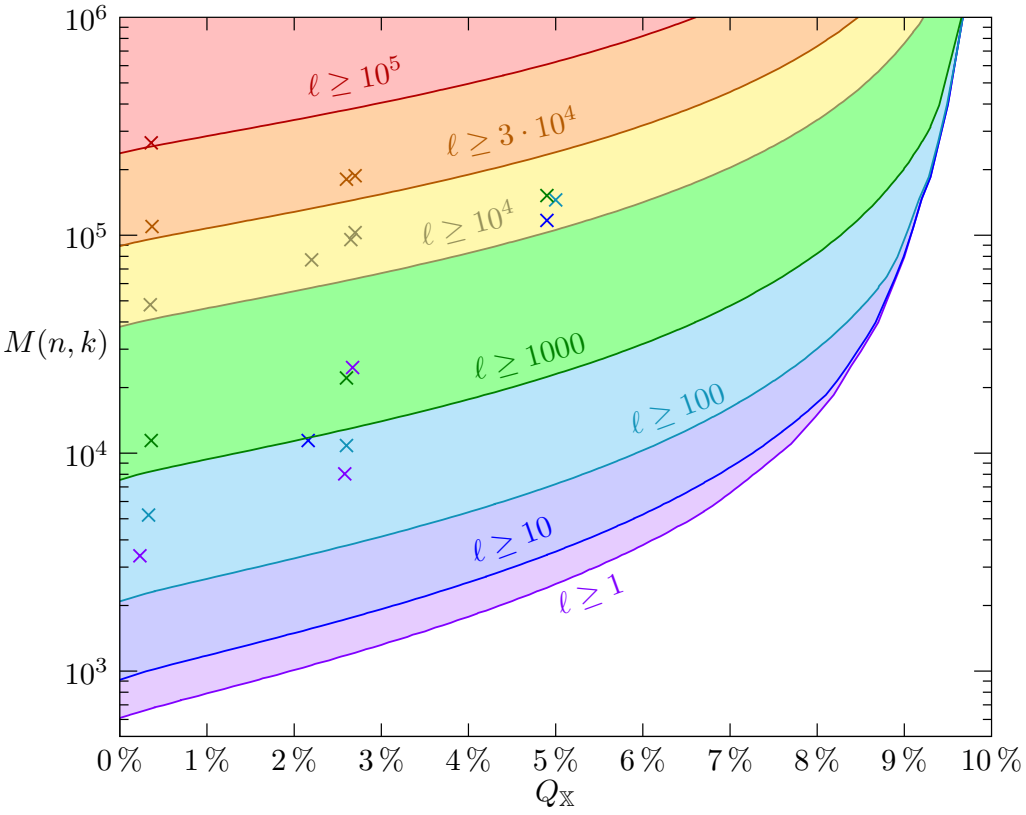


Figura 4.9: **Required bits for a secret key.** Minimum number of received bits $M(n,k)$ needed to obtain a ε_{sec}-GS key of a given length ℓ (as labelled on each curve) versus the quantum BER $Q_{\mathbb{X}}$. Different colors divide the regions with different secret key lengths. Crosses represent our experimental results, the colored regions and the solid lines that delimit them are derived from the numerically optimized bound, assuming $Q_{\mathbb{Z}} = Q_{\mathbb{X}}$.

the achievement of higher secret key rates with respect to general secrecy, which pays the price for the higher level of secrecy it provides. The gain becomes more evident when the channel becomes noisier and the QBER increases. We also observe that with $Q_{\mathbb{X}} = 4.9\%$ ε_{sec}-GS secure keys are obtained for $p_{\mathbb{Z}} = 16\%$, $p_{\mathbb{Z}} = 28\%$, $p_{\mathbb{Z}} = 40\%$ and $p_{\mathbb{Z}} = 49\%$ and not for $p_{\mathbb{Z}} = 9\%$, whereas, when $Q_{\mathbb{X}} = 8.3\%$, only keys secure against pragmatic secrecy can be extracted with the parameters we used.

We point out that the bounds derived for the general and pragmatic secrecy do take into account statistical fluctuations: if the measured $\hat{Q}_{\mathbb{Z}}$ is greater than $Q^{\mathbb{Z}}_{\text{tol}}$ the protocol aborts, while for $\hat{Q}_{\mathbb{Z}} < Q^{\mathbb{Z}}_{\text{tol}}$ the protocol gives a secure key with security parameter ε_{sec}. As an example, given $Q_X = 4.9\%$, $Q_Z = 6.0\%$, $n = 100000$ and $p_{\mathbb{Z}} = 9\%$, the parameter μ which takes into account these fluctuations for general secrecy (see Eq. (4.19)), is approximately equal to 0.15, a value which, for an experimentally realistic number of bits disclosed during the information reconciliation procedure, and even without the contribution of $Q^{\mathbb{Z}}_{\text{tol}}$, yields the impossibility of producing a secret key.

Moreover, we notice that higher values of $p_{\mathbb{Z}}$ ($\sim 50\%$) better suit lower values of n for both general and pragmatic secrecy in all considered cases: for instance, when $Q_{\mathbb{X}} = 0.3\%$ in the general secrecy case, $p_{\mathbb{Z}} = 49\%$ is optimal for $n < 3 \cdot 10^3$; on the other hand, as n increases, it is possible to decrease $p_{\mathbb{Z}}$ and when $n \simeq 10^5$ the highest rate is obtained with $p_{\mathbb{Z}} = 16\%$. This feature can be understood in the following way: for a short sifted key $\mathbf{X}$, an almost equally long string $\mathbf{Z}$ ($k \sim n$) is needed to reliably detect eavesdropping; when n grows, less bits of $\mathbf{Z}$ (in percentage) are necessary. In fact, in the large n limit, it is possible to choose k so that k/n vanishes as n goes

to infinity and the secret key rate approaches the asymptotic bound, $r = 1 - h_2(Q_{\mathbb{X}}) - h_2(Q_{\mathbb{Z}})$.

It is worth noting that, in the asymptotic limit, a biased choice of the bases gives a higher secure key rate with respect to the BB84 protocol (§3.4.1) whenever $p_{\mathbb{X}} > \sqrt{1/2}$. In fact, in the infinite limit, the fraction of secure over sifted bits is given by $1 - 2H(Q)$ in both cases (for simplicity we here assume $\hat{Q}_{\mathbb{X}} = \hat{Q}_{\mathbb{Z}} = Q$); however, a biased choice of the bases gives a number of sifted bits that is approximately a fraction $p_{\mathbb{X}}^2 > 1/2$ of the sent bits (also in the finite size regime), while for the BB84 protocol the sifted bits are $1/2$ of the sent bits. In particular, by using a large $p_{\mathbb{X}}$, namely $p_{\mathbb{X}} \sim 1$, in the infinite key limit we approach a double secret key rate with respect to BB84. In Fig. 4.8 the asymptotic bound of the secure key rate r, defined as the number of secure bits over number of sent bits, is twice the corresponding asymptotic bound of the BB84 protocol.

With the obtained data we also estimated the minimum number of received qubits M that are needed in order to obtain a key of given length ℓ. In figure 4.9 we show this quantity as a function of the QBER (in this case we assumed that $Q_{\mathbb{X}} = Q_{\mathbb{Z}}$). Solid lines represent the theoretical minimum M necessary to obtain a general secret key for different lengths ℓ. With markers of different colors we indicate the experimental received qubits for the different values of ℓ. Clearly, as the QBER grows, it is necessary to increase the number of exchanged qubits to obtain a given key length ℓ. On the other hand, when the channel is almost noiseless, a secret key of reasonable length can be extracted by using a relatively small number of qubits: for instance, more than 1000 secure key bits can be obtained by exchanging less than 20000 photons (see Fig. 4.9).

4.3.5 Discussion

In conclusion, we have experimentally demonstrated the feasibility of key distillation according to the finite-key analysis proposed in [74] and compared it with a less stringent definition of security, called pragmatic, that protects the protocol against intercept and resend attacks. We compared the two analyses for different amounts of depolarizing noise added to the quantum channel.

With pragmatic security, a significantly secret key rate with finite keys is demonstrated, even in conditions near the theoretical $Q_{\mathbb{X}}$, $Q_{\mathbb{Z}}$ bound of 11%. Its drawback is the insecurity against collective attacks, which however are not presently available. We stress that, when the channel is very noisy ($Q_{\mathbb{X}} = 8.3\%$) no key that is secure against the most general quantum attack could be extracted up to $2 \cdot 10^5$ sifted bits; however, by considering only intercept and resend attacks, in this case a secrect key rate up to 7.5% was obtained. When $Q_{\mathbb{X}}, Q_{\mathbb{Z}} > 11\%$ it is not possible to obtain a secure key even in the asymptotic large n limit. This shows that, for highly noisy channels, the use of pragmatic secrecy is a viable solution to obtain some secret bits for a experimentally realistic number of exchanged photons. We believe that this work can have important applications for free-space quantum communication and for all QKD scenarios in which the number of exchanged qubits is limited by physical constraints, such as in the inter-satellites link scenario.

4.4 Long distance free-space quantum key distribution using turbulence as a resource

The transmission of quantum states to a receiver located far away on the Earth or on some mobile, or even orbiting, station is the frontier of Quantum Communications, aiming to extend the networking currently available through fiber to a planetary scale and beyond.

The protocols devised for such purposes rely on a clean detection of quantum bits - or qubits - that are most conveniently encoded in single light-quanta. In free-space communications, a number of background photons are always present, together with detector non-idealities such as dark counts and dead-time. These effects are detrimental to the quantum protocol accomplishment. In the case

of quantum key distribution (QKD), a threshold on the overall quantum bit error rate (QBER), due to experimental imperfections and noise, limits the possibility of exchanging a secure key, in that, for a higher QBER, the unconditional secrecy of the key cannot be guaranteed.

Current demonstrations of QKD have avoided the condition of normal background by operating in dark nights or by using a very strict filtering that imposes a low key rate already on urban scale (see, e.g., [148, 149, 132, 59, 150, 151]). However, while aiming at QKD over long links in realistic conditions including daylight, a breakthrough in the protocol is needed for enabling key distillation even when the QBER is above the secure threshold. Due to unavoidable background photons, the QBER is below the threshold only in case of high channel transmission. We devise here a solution that, by exploiting the atmospheric turbulence, allows secret key generation in part of the link time, even when the average transmission is below the limit of secure communication.

In a link with fluctuating transmission coefficient and a significant attenuation, due to turbulence and to the combination of optical diffraction and scintillation, respectively, it is possible to devise a solution to the high QBER problem on the base of a sound characterization of the channel transmission. A recent study [152] pointed out that the temporal profile of the transmissivity typically has peaks lasting for a few milliseconds, distributed in a low transmissivity background. Post-selection techniques based on QBER estimation in short time frames are ineffective here, because the QBER value cannot be reliably estimated on this time scale. An additional resource may be introduced to estimate the link transmissivity in its intrinsic time scale with the use of an auxiliary classical laser beam co-propagating with the qubits, but conveniently interleaved in time. In this way the link scintillation is monitored in real-time and the selection of the time intervals of high channel transmissivity corresponding to a viable QBER for a positive key generation is made available.

In the following sections, we present a demonstration of a protocol for adaptive real-time selection implementing this approach, in conditions of losses equivalent to long distance and satellite links, and with a range of scintillation corresponding to moderate to rough weather. A useful criterion for the selection of the low QBER interval is presented. The proposed solution employs a train of intense pulses propagating on the same path as the qubits, with parameters chosen such that its fluctuation in time reproduces that of the quantum communication.

4.4.1 Experimental setup

The link used in our QKD demonstration is the 143 Km free-space channel between La Palma and Tenerife islands shown in figure 4.10. The B92 protocol (see §3.4.3) was chosen for the experiment, because of its simple implementation.

The transmitter (Alice) was located at the JKT observatory in the island of La Palma where two 850 nm attenuated lasers provided the quantum signal and a 808 nm laser was used as atmospheric probe. The polarization of the 850 nm lasers was set to the two different B92 preparation states ($|\updownarrow\rangle$ for bit 0 and $|\swarrow\nearrow\rangle$ for bit 1, respectively) by means of half wave plates and quarter wave plates. The encoding of the quantum signal was then obtained by controlling the lasers with an FPGA. Classical and quantum lasers were coupled into mode fibers and injected into a fiber beam splitter. One of the two beam splitters output was delivered to a suitably designed Galilean telescope whose main characteristic is a singlet aspheric lens with diameter 230 mm and focal length 2200 mm. This lens allowed us to get, after 143 Km of propagation, a beam spot comparable to the dimensions of the primary mirror of the receiving telescope, so that the power transfer between the two parties was maximized. In order to compensate the beam wandering induced by the atmosphere, we implemented a feedback loop for controlling the direction of the transmission: the fiber delivering the signal to the transmitter was mounted on a XYZ movable stage in correspondence of the focal place of the 230mm lens, with computer controlled stepped motors. On this same stage, we

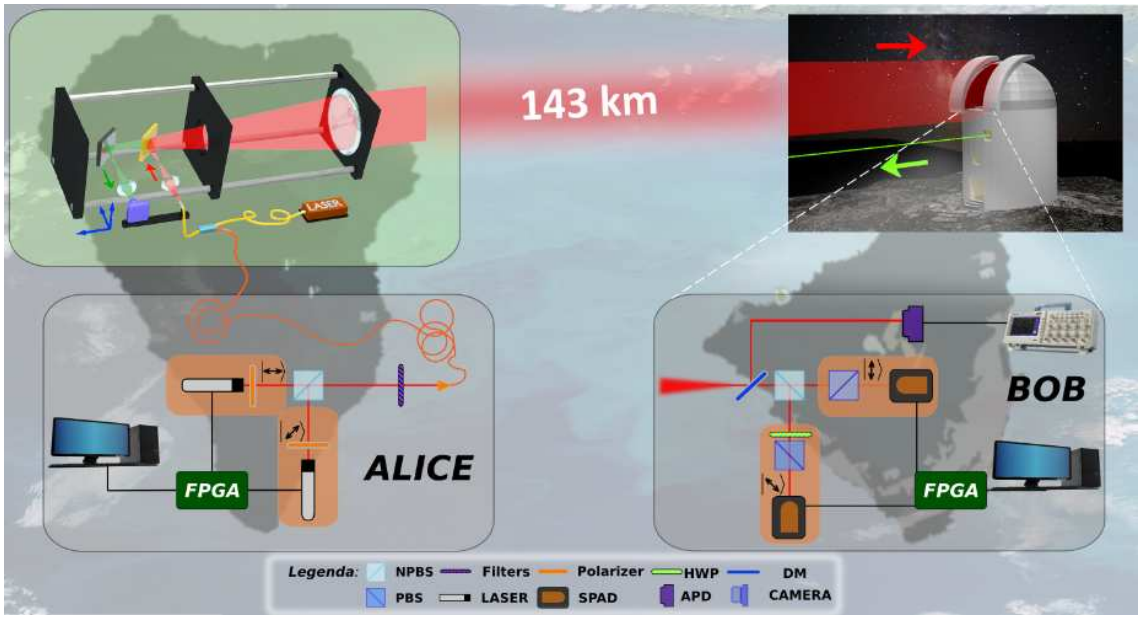


Figura 4.10: Experimental setup: Alice located at JKT observatory in La Palma, sends qubits by firing two 850 nm FPGA-controlled attenuated lasers and an atmospheric laser probe (30mW @ 808 nm). Both the quantum and probe signals are coupled into the same optical path of a custom made telescope. The telescope is also used to track a beacon laser sent by Bob, located at the Optical Ground Station (OGS) in Tenerife. Bob receives through the OGS telescope both the signals: the probe is monitored by an APD and the qubits are detected with two SPADs.

mounted a CCD sensor which acquired a green (532 nm) "beacon" laser sent by Tenerife towards Alice's telescope. The camera recorded an image of the singlet focal plane. The wandering of the beacon on the CCD was then analyzed in real-time by a software that moved the XYZ stage in order to compensate for the movement of the beacon spot on the camera.

At the receiver (Bob), located in Tenerife, we used the 1 m aperture telescope of the ESA Optical Ground Station to receive the signals. After the Coudè path, we collimated the beam and the classical and quantum signals (at different wavelength) were divided by a dichroic mirror. The qubits were measured in two bases, using polarizing beam splitters (PBS) and waveplates. The counts detected by two single-photon avalanche photodiodes (SPAD) were stored on a FPGA. The probe beam was detected by an high-bandwidth APD (Avalanche Photon Detector) and then registered and stored by an oscilloscope.

As for the transmitted qubits, we used the data structure described in §4.2.1. In particular, a raw key is composed by N packets of 1440 bits each, which are divided into 6 frames for the ease of synchronization. In fact, each frame consists of 11 header slots and 240 payload slots with a duration of 800 ns; as regards the payload slots, Alice sends two qubits separated by 200 ns. The two FPGAs are synchronized every second by a pulse-per-second (pps) signal equipped by two GPS receivers located in the two islands.

As a conclusion to this section, we point out that at the transmitter side, we were not working in the single photon regime, thus meaning that the transmitted qubits had, on average, more than one photon per pulse. This choice arises from that the experimental hardware used for this experiment had a maximum transmission rate of 2.5 MHz. Hence, the 30 dB average attenuation of the channel would have made the acquisition too slow for the allocated available time slots at the Observatories of Santa Cruz de La Palma and Tenerife. Our aim, however, was to simulate

a possible, realistic scenario where one could employ a faster free-space QKD system (hundred of MHz to GHz, see, e.g., [151]). In particular, by assuming a transmitter emitting single photon pulses with a repetition rate of 100 MHz, we would observe approximately the same photon rate we registered, given that the key distribution would be affected by the same levels of optical and atmospheric attenuation. Therefore, in the present proof-of-principle, our experimental conditions can be assimilated to a regime of single photon per pulse at the receiver.

Some further details on how the experimental data were prepared for the analysis, can be found in [153].

4.4.2 Preliminary analysis

In order to test the ability of estimating the link transmissivity, we first sent on the same free-space channel, two signals: the classical probe detected with a fast photodiode at the receiver, and a single strongly attenuated laser. The classical signal was made by pulses of 100 μs at 1 kHz repetition rate, while the attenuated laser at 850 nm fired a continuous beam. At the receiver, the quantum signal was detected by a Single Photon Avalanche Diode (SPAD) and acquired in packets with duration of 1 ms.

In order to test the correspondence between the intensity of the received classical beam and the photons received on the quantum channel, we compared the photon counts detected in each packet with the voltage registered by the fast photo-diode. In Figure 4.11 we show these signals for 11 s of acquisition time; as it can be seen in the inset, there is a strong correspondence between them.

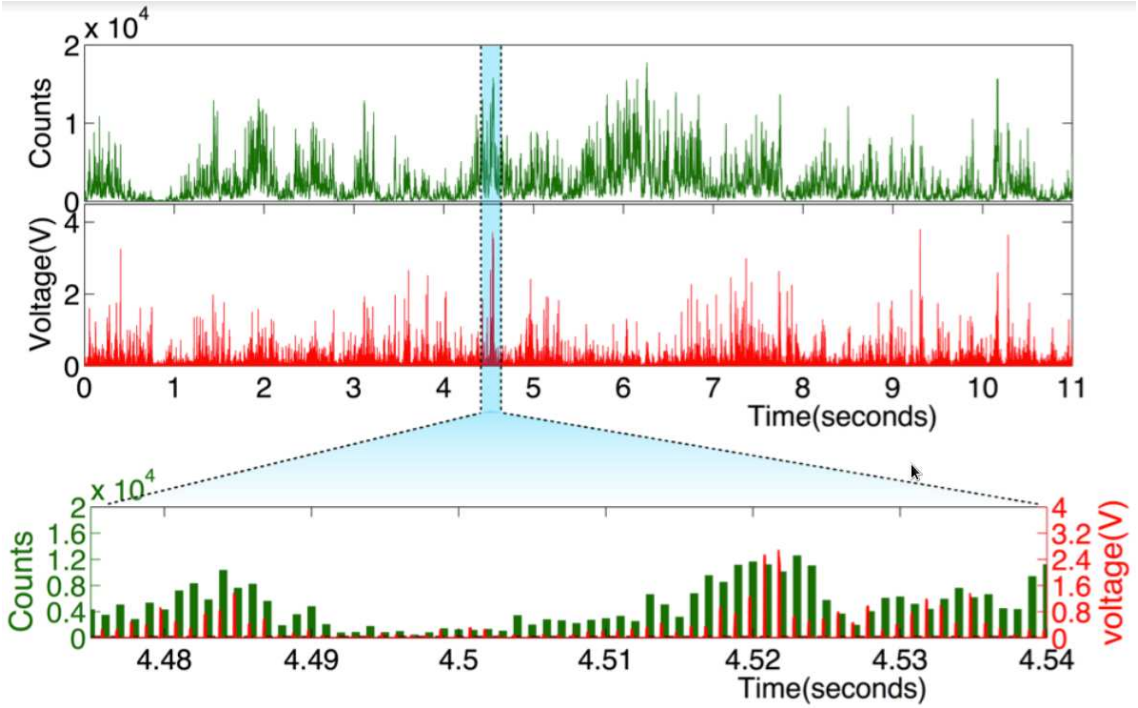


Figura 4.11:

To further demonstrate the correlation, we performed the following analysis. Given a set of L packets (each with a duration of 1 ms), we let V_i be the probe signal amplitude, S_i the number of detected photons in the quantum signal for the i-th packet, respectively. We set a threshold value V_{T} for the probe voltage and select only those packets such that $V_i > V_{\mathrm{T}}$; in particular, we denote by $\mathcal{I}(V_{\mathrm{T}}) = \{i \in [1, L] : V_i > V_{\mathrm{T}}\}$ the indexes of the packets for which the above condition holds

and by $P(V_{\mathrm{T}})$ the corresponding number of packets, that is, $P(V_{\mathrm{T}}) = |\{\mathcal{I}(V_{\mathrm{T}})\}|$. Furthermore, we define the following quantities:

$$S(V_{\mathrm{T}}) \quad = \quad \sum_{i \in \mathcal{I}(V_{\mathrm{T}})} S_i, \quad \overline{\mathcal{S}}(V_{\mathrm{T}}) = \frac{S(V_{\mathrm{T}})}{P(V_{\mathrm{T}})} \tag{4.25}$$

with $S(V_{\mathrm{T}})$ representing the total number of detected bits and $\overline{\mathcal{S}}(V_{\mathrm{T}})$ the mean number of detections per packets after the adaptive real-time selection performed with threshold V_{T}.

The effect of the adaptive real-time selection can be clearly appreciated in Fig. 4.12, where $\overline{\mathcal{S}}(V_{\mathrm{T}})$ is plotted (green line) as a function of the threshold, showing that a larger mean number of counts per packet corresponds to a higher threshold value . This demonstrates that the probe and quantum signals are strongly correlated and one can improve significantly the SNR by thresholding. As a side effect, we have that the adaptive real-time selection decreases the overall number $S(V_{\mathrm{T}})$ of detections in the transmission, as can be noticed by considering the ratio $S(V_{\mathrm{T}})/S(V_{\mathrm{T}} = 0)$ (blue line). Hence, a trade-off between these conflicting effects should be found while looking for the threshold which maximizes the final key rate, as discussed in the following paragraphs.

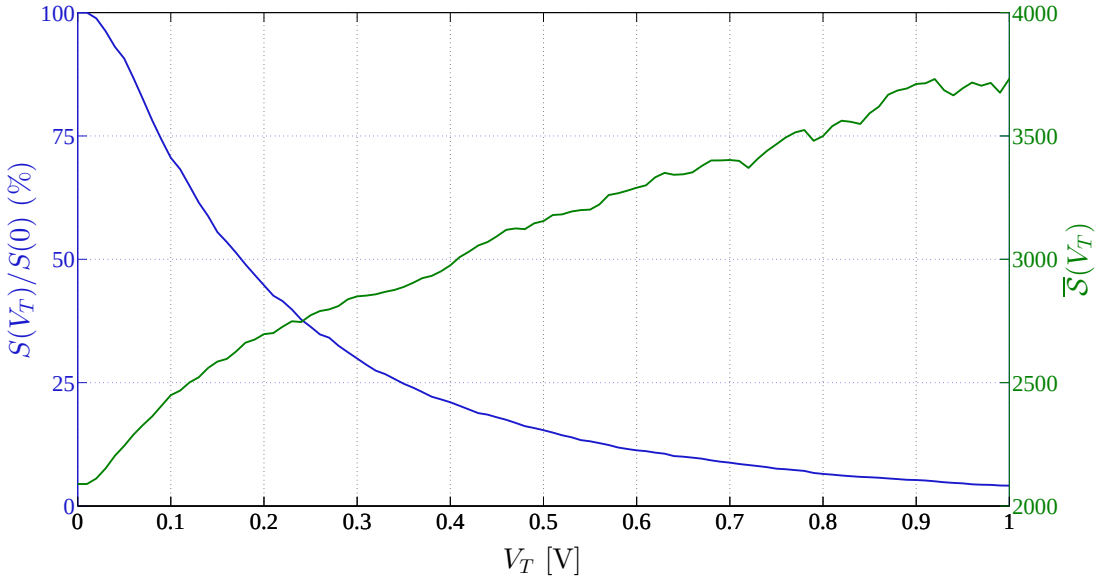


Figura 4.12: Mean counts per packet $\overline{\mathcal{S}}(V_{\mathrm{T}})$ and fraction of total count $S(V_{\mathrm{T}})/S(V_{\mathrm{T}} = 0)$ as a function of the probe threshold.

We now apply the above results to a QKD experiment. In particular, we show that increasing the SNR by thresholding with the adaptive real-time selection protocol gives, in some cases, benefits in terms of the secret key length, even if the total number of sifted bits will decrease. In fact, when the QBER is above 11%, i.e., the maximum QBER tolerable for standard QKD using one-way reconciliation with no noisy pre-processing, adaptive real-time selection reduces the QBER below this limit, thus enabling secure key generation. More details on how the data was stored and prepared for the analysis can be found in [153].

First, given the number of errors E_i in the i-th packet, we define the overall number of errors $E(V_{\mathrm{T}})$ and the quantum bit error rate $Q(V_{\mathrm{T}})$ in the real-time selected packets as

$$E(V_{\mathrm{T}}) = \sum_{i \in \mathcal{I}(V_{\mathrm{T}})} E_i \,, \quad Q(V_{\mathrm{T}}) = \frac{E(V_{\mathrm{T}})}{S(V_{\mathrm{T}})}. \tag{4.26}$$

For evaluating the actual impact of the adaptive real-time selection on the performance of a quantum key distribution system, it is then important to study how the two complementary effects of thresholding, i.e. the increase of mean detected bits per packet $\overline{\mathcal{S}}(V_{\mathrm{T}})$ versus the decrease of total detections $S(V_{\mathrm{T}})$, influence the achievable secret key rate of the system, so that the optimal trade-off can be found. Given these quantities, the asymptotic key rate of a QKD system based on the B92 protocol and with the described probe thresholding mechanism reads as follows:

$$R(V_{\mathrm{T}}) = \frac{S(V_{\mathrm{T}})}{S(0)} \left[1 - 2h_2\left(Q(V_{\mathrm{T}})\right)\right]. \tag{4.27}$$

It is worth noting that evaluating the asymptotic rate instead of the finite-length one [74, J1] may appear as a restrictive approach, especially because the real-time selection further reduces the number of available sifted bits. However, the size of the blocks to be fed as input to the key distillation procedure (i.e., information reconciliation and privacy amplification) can be chosen, without loss of generality, so that the asymptotic bound provides a reasonable approximation of the actual rate.

4.4.3 QBER and secret key rate prediction

As demonstrated in [152], the statistics of the transmission of a long free-space channel follows a log-normal distribution. The measured probe voltage at the receiver, being the transmitted intensity constant in time, follows the same distribution, given by

$$p(V; m_V, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{V} e^{-[(\ln \frac{V}{m_V} + \frac{1}{2}\sigma^2)]^2/(2\sigma^2)}, \tag{4.28}$$

where σ^2 is defined as function of the mean m_V and of the variance v_V of the probe intensities distribution, that is,

$$\sigma^2 = \ln\left(1 + \frac{v_V}{m_V^2}\right). \tag{4.29}$$

As an example, Figure 4.13 shows the distribution of the measured voltages of the data used in figure 4.11, that, according to the theory, follows a log-normal distribution [152].

In the following analysis, we assume that the number of detected photons and the probe intensity have completely correlated log-normal distributions [152]. This trivially implies that both distributions have the same parameter σ^2. By this hypothesis, we can predict the number of packets above threshold, $P_{\mathrm{th}}(V_{\mathrm{T}})$, and the number of sifted bits surviving the thresholding, $S_{\mathrm{th}}(V_{\mathrm{T}})$, in case of null background as

$$\frac{P_{\mathrm{th}}(V_{\mathrm{T}})}{P(0)} = \int_{V_{\mathrm{T}}}^{+\infty} p(V; m_V, \sigma) dV, \tag{4.30}$$

$$\frac{S_{\mathrm{th}}(V_{\mathrm{T}})}{S(0)} = \int_{V_{\mathrm{T}}}^{+\infty} \frac{V}{m_V} p(V; m_V, \sigma) dV \tag{4.31}$$

where $P(0)$ and $S_{\mathrm{th}}(0)$ are the overall experimental number of packets and sifted bits, respectively, when no threshold is used. By taking into account the background clicks we get:

$$P_{\mathrm{th}}(V_{\mathrm{T}}) = P(0)\frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{\ln \frac{V_{\mathrm{T}}}{m_V} + \frac{1}{2}\sigma^2}{\sqrt{2\sigma^2}}\right)\right], \tag{4.32}$$

$$S_{\mathrm{th}}(V_{\mathrm{T}}) = n_b P_{\mathrm{th}}(V_{\mathrm{T}}) + \frac{1}{2}[S(0) - n_b P(0)]\left[1 - \mathrm{erf}\left(\frac{\ln \frac{V_{\mathrm{T}}}{m_V} - \frac{1}{2}\sigma^2}{\sqrt{2\sigma^2}}\right)\right], \tag{4.33}$$

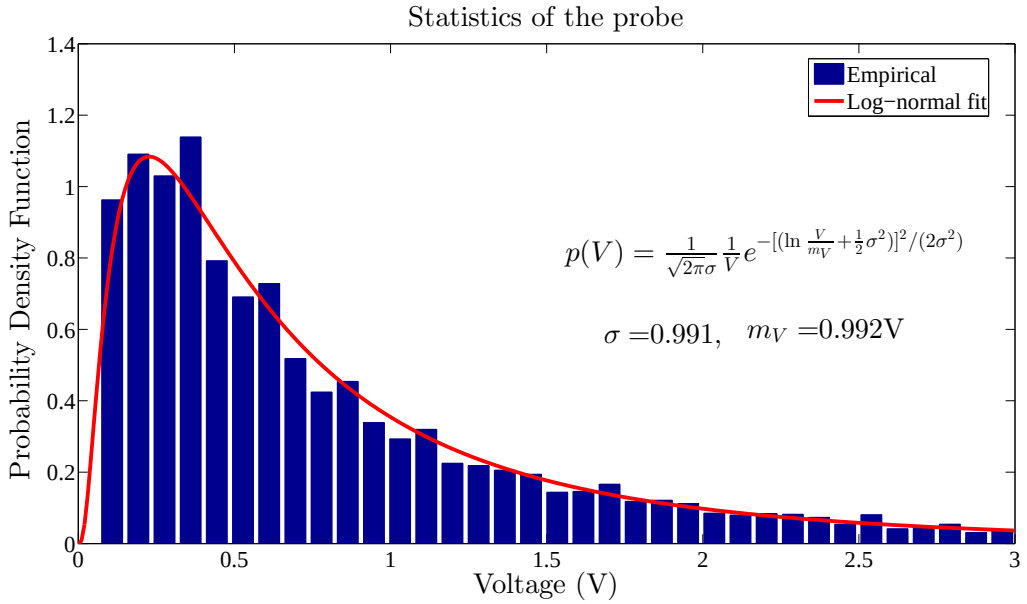


Figura 4.13: Comparison of the empirical probe intensity distribution and of its log-normal fit.

where n_b is the experimental average number of background counts per packet. In fact, the assumption of complete correlation between the quantum and the probe signal is not strictly verified in our experiments and eq. (4.33) turns out to be an approximation of the experimental values. Still, it allows to derive an effective real-time selection threshold, as will be seen in the following (e.g., in figure 4.14).

We now define a further predictive model for estimating the bit error rate on the quantum channel as a function of the probe threshold. Let us assume that the average bit error rate on the quantum channel is m_Q and that the number of counts per packet due to background noise is n_b. Now, since background counts output a random result, the corresponding bit error rate is $1/2$, and we can write the predicted quantum bit error rate Q_{th} as a function of the threshold V_{T}, namely,

$$Q_{\text{th}}(V_{\text{T}}) = m_Q\left(1 - \frac{n_b}{\overline{\mathcal{S}}_{\text{th}}(V_{\text{T}})}\right) + \frac{1}{2}\frac{n_b}{\overline{\mathcal{S}}_{\text{th}}(V_{\text{T}})}, \tag{4.34}$$

where the predicted value for $\overline{\mathcal{S}}_{\text{th}}(V_{\text{T}}) = S_{\text{th}}(V_{\text{T}})/P_{\text{th}}(V_{\text{T}})$ is obtained by using equation (4.33). Hence, we can finally write the predicted secret key rate as follows

$$R_{\text{th}}(V_{\text{T}}) = \frac{S_{\text{th}}(V_{\text{T}})}{S(0)}\left[1 - 2h_2\left(Q_{\text{th}}(V_{\text{T}})\right)\right]. \tag{4.35}$$

4.4.4 Experimental results

In figure 4.14, we compare the expected (solid line) and the experimental values (circles) for both the QBER (red) and the asymptotic key rate (black) as a function of the probe intensity threshold in a QKD run. The curves for the expected QBER and for the key rate were obtained by substituting maximum likelihood estimates for the log-normal parameters m_V and σ^2 in eq. (4.34) and in eq. (4.35). The other two parameters, $S(0)$ and $P(0)$, required for predicting $S(T)$ and $P(T)$, are directly measured (they correspond to the total number of sifted bits and the total number of packets received respectively).

The experimental data refer to an acquisition of $5 \cdot 10^5$ sifted bits in condition of high background, simulated by an external light source. The intensity of the background was chosen in order to obtain a mean QBER larger than 11%. In particular, we measured an average value of $n_b = 35.17$ for the background clicks per packet and we assume $m_Q = 5.6 \cdot 10^{-2}$. As clearly shown in the figure, eq. (4.34) provides a good approximation of the experimental curve.

As one can appreciate from the same Figure, we have a remarkable correspondence between the shape of the theoretical rate, R_{th}, and the measured rate, R_{exp}. The fact that the experimental points do not fit the expected curve can be ascribed to the discrepancy in the empirical joint distribution of probe intensities and counts with respect to the model; in particular, we measured the following fitting parameters for the normalized log-normal distributions: $\sigma_V^2 = 0.967$ for the probe intensities and $\sigma_S^2 = 0.716$. However, the derivation of the optimal threshold for maximizing the secret key length (green dashed line) from the probe distribution yields the optimal V_T also for the experimental data. In particular, the optimal threshold inferred from the probe distribution is $V_{T,\mathrm{opt}}^{(\mathrm{th})} = 375$ mV, and coincides with the one resulting from optimization on the experimental data, yielding a rate of $R(V_{T,\mathrm{opt}}^{(\mathrm{th})}) = 5.55 \cdot 10^{-2}$.

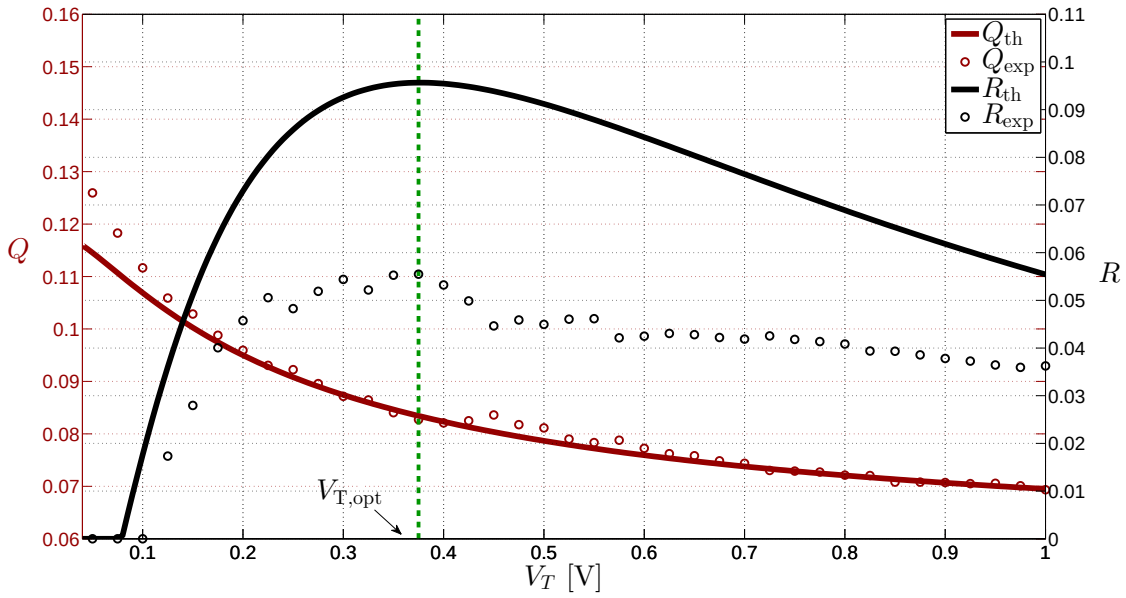


Figura 4.14: Predicted (solid lines) and experimental (markers) values for the QBER and for the secret key rate depending on the real-time threshold value. The optimal threshold, $V_{T,\mathrm{opt}}^{(\mathrm{th})} = 375$ mV, is shown by the magenta line.

Also, we observe that for $V_\mathrm{T} < 100$ mV no key can be extracted, being the QBER higher than the theoretical maximum (i.e., $Q = 11\%$), whereas by increasing the threshold value a non-zero secret key rate is achievable. With the optimal threshold value, the measured QBER is $Q(V_{T,\mathrm{opt}}^{(\mathrm{th})}) = 8.34 \cdot 10^{-2}$; a significant gain with respect to the initial value, $Q(0) = 13.14 \cdot 10^{-2}$ is therefore achieved. Finally, we observe that for increasing values of $V_\mathrm{T} > V_{T,\mathrm{opt}}^{(\mathrm{th})}$ the QBER still decreases, but so does the rate, since the reduction in the residual number of sifted bits does not compensate the advantage obtained thanks to the the lower QBER. This result is of absolute practical relevance, as it shows that leveraging the probe intensity information is an enabling factor for quantum key distribution, since it allows to distill a secret key even when without the real-time selection it would not be possible.

4.4.5 Comment on security and conclusions

As for the security of the proposed adaptive real-time selection protocol as applied to a QKD system, no advantage is delivered to a potential attacker in the true single photon regime, being the thresholding nothing but a further sifting step on the received bits [101, 100]. If the attacker tried to force Alice and Bob to select a particular bit, in fact, she would alter the probe signal *before* the disclosure of the preparation bases on the public channel, and, therefore, before she could actually know if her measured bit is correct. On the other hand, altering the probe statistics or interrupting the probe transmission would not yield any advantage to the attacker, as it would just break the correlation between the quantum and the classical signal and would thus result in a denial of service attack. The security analysis gets more involved if we allow *photon number splitting* (PNS) attacks. In that case, the attacker may force Bob to receive just the qubits for which the PNS attack was successful, i.e., only those pulses with multiple photons. A decoy state protocol may counteract this strategy, but its effectiveness with a turbulent free-space link has to be investigated (see [154] for some considerations).

Summarizing, we showed that the turbulence of the channel can be exploited for allowing secret key distillation even when the average QBER of the transmission is above the critical threshold of 11%. The proposed adaptive real-time selection protocol was demonstrated in a realistic scenario, and, by leveraging the classical information provided by the probe signal, it allowed to reduce the QBER and to enable secret key distillation even in low SNR scenarios; more in general, the protocol could allow to increase the secret key rate. The prediction of the optimal threshold can be inferred by just observing the probe intensity statistics, and the predicted value is consistent with the experimental one. Finally, let us remark that the protocol can be integrated with existing systems.

Capitolo 5

Conclusions

This work has explored two main topics at the cross-road of information theory and quantum physics, that is, randomness extraction as applied to quantum random number generation and quantum key distribution. We moved from the problem of creating uniform randomness to the one of sharing secret random sequences between distant parties, so that a bounded information leakage to the adversary is guaranteed in an information-theoretic security framework. With these problems in mind, classical processing algorithms for quantum information security have been investigated, with a strong focus on their application to realistic scenarios.

A randomness extractor for a commercial quantum random number generator has been developed and the uniformity of its output has been quantitatively evaluated. More in general, a framework for randomness extraction for block-wise randomness extractors was described, so that the implemented solution could be easily applied to other physical random number generators, by properly choosing the extractor parameters according to the collision entropy of the input source.

In the domain of Quantum Key Distribution, a taxonomy of information reconciliation protocols was proposed. Then, different protocols were analyzed and their performance evaluated depending on the input parameters. In particular, a novel, thorough analysis of the Winnow scheme was carried out, and its performance were analytically assessed. An optimization method for its parameters was then provided, so that a given target bit error rate could be ensured at the protocol output.

As for the privacy amplification phase, a security analysis in the scenario of selective individual attacks has been carried out. As a first step, a tight bound on the information leakage while using the class of random binary matrices (and the one of random Toeplitz binary matrices) has been derived, showing that the average bound was rather loose, though much easier to compute. Then, the security analysis for a possibly aborting protocol was proposed in the finite-key scenario, so that the secret key length can be derived according to the required secrecy level.

The obtained theoretical results and the considered classical algorithms were then implemented and exploited for the practical implementation of Quantum Key Distribution systems in free-space. In particular, three experiments were carried out. The first one, performed over a 50 meters link, tested the real-time feasibility of free-space quantum key distillation under selective individual attacks, including intercept-and-resend, unambiguous state discrimination and photon number splitting. The second experiment was based on the efficient BB84 protocol, and extended the adversarial scenario to general quantum attacks. The achievable secret key rates were compared under different noise conditions and according to two distinct security definitions, pragmatic and general secrecy. The results have shown that secret keys can be distilled even when considering finite-key effects in noisy scenarios. In the last experiment, a novel approach to free-space Quantum Key Distribution was proposed, leveraging the turbulence of the channel to enable key distillation

even in harsh conditions. The feasibility of key distillation over a 143 Km free-space link was shown, even when the average bit error rate on the quantum channel would have prevented the key exchange. The application scenarios for this new protocol include daylight Quantum Key Distribution over long distances, and its extensions to satellite communications.

Acknowledgments/Ringraziamenti

First of all, I would like to thank my supervisor, Dr. Nicola Laurenti, for his patient support and his insightful guidance, which I never lacked during these three years. His door was always opened, for both professional and personal discussions. I also thank Dr. Giuseppe Vallone and Prof. Paolo Villoresi, for having shared their passion for research and their brilliant ideas and comments on our activities. Thanks to ID Quantique, and in particular to Damien Stucki, Matthieu Legré, Gregoire Ribordy and Patrick Trinkler, who game me the opportunity to spend six months in the company and to work with them in an exciting environment.

Thanks to all the people who shared their personal and professional experience with me during these years: Paolo Baracca, Nicola Dalla Pozza, Francesco Michielin, Giulia Cisotto, Francesco Renna, Simon Calimani, Ilaria Savorgnan, Davide Bacco, Davide Marangon, Francesca Gerlin.

Thanks to my wife, Giulia, for having always been by my side, with her love and invaluable attentions. Finally, thanks to my parents, who made all this possible.

This work has been carried out within the Strategic Research Project QUANTUMFUTURE of the University of Padova.

Grazie innanzitutto al mio supervisore, il Dr. Nicola Laurenti, per il supporto paziente e per la guida sicura, che non mi ha mai fatto mancare durante questi tre anni. La sua porta è sempre stata aperta, per discutere sull'attivitá di ricerca, ma anche per un confronto personale.

Ringrazio il Dr. Giuseppe Vallone ed il Prof. Paolo Villoresi per aver condiviso con me la loro passione per la ricerca, le loro brillanti idee e i commenti sulle nostre attivitá.

Grazie ad ID Quantique, ed in particolare a Damien Stucki, Matthieu Legré, Gregoire Ribordy e Patrick Trinkler, per avermi dato l'opportunitá di trascorrere sei mesi presso la loro azienda e di lavorare insieme in un ambiente stimolante.

Grazie a tutte le persone che hanno condiviso la loro esperienza personale e professionale con me durante questi anni, rendendo l'esperienza di dottorato così ricca dal punto di vista umano: Davide Bacco, Paolo Baracca, Simon Calimani, Giulia Cisotto, Nicola Dalla Pozza, Francesca Gerlin, Davide Marangon, Francesco Michielin, Francesco Renna, Ilaria Savorgnan,.

Grazie a mia moglie, Giulia, per essermi sempre stata accanto ed avermi incoraggiato lungo tutto questo percorso, con il suo amore e le sue preziose attenzioni.

Infine, grazie ai miei genitori, per aver reso tutto questo possibile.

Questo lavoro è stato svolto all'interno del Progetto Strategico di Ateneo QUANTUMFUTURE dell'Universitá di Padova

Publications

Journal

[J1] Davide Bacco, Matteo Canale, Nicola Laurenti, Giuseppe Vallone, Paolo Villoresi. Experimental Quantum Key Distribution with finite-key security analysis for noisy channels.
Nature Communications 4, 2363 (2013) DOI:10.1038/ncomms3363.

Conference

[C1] Matteo Canale, Francesco Renna, and Nicola Laurenti, "QKD secrecy for privacy amplifaction matrices with selective individual attacks," in Annual conference on Quantum Cryptography, QCRYPT, 2011.
`http://www.qcrypt2011.ethz.ch/programme/posters`.
`http://quantumfuture.dei.unipd.it/attachments/article/76/CanaleEtAl_Qcrypt2011B.pdf`

[C2] Matteo Canale, Davide Bacco, Simon Calimani, Francesco Renna, Nicola Laurenti, Giuseppe Vallone, and Paolo Villoresi, "Performance analysis of a low-cost, low-complexity, free-space QKD scheme based on the B92 protocol," in Annual conference in Quantum Cryptography, QCRYPT, 2011.
`http://www.qcrypt2011.ethz.ch/programme/posters`.
`http://quantumfuture.dei.unipd.it/attachments/article/76/CanaleEtAl_Qcrypt2011A.pdf`

[C3] Matteo Canale, Davide Bacco, Simon Calimani, Francesco Renna, Nicola Laurenti, Giuseppe Vallone, and Paolo Villoresi, "A prototype of a free-space QKD scheme based on the B92 protocol," in International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL, 2011.
`http://dl.acm.org/citation.cfm?doid=2093698.2093884`.

[C4] Davide Bacco, Matteo Canale, Nicola Laurenti, Giuseppe Vallone, and Paolo Villoresi, "Experimental quantum key distribution with finite-key security analysis for noisy channels.," in Annual conference on Quantum Cryptography, QCRYPT, 2013.
`http://2013.qcrypt.net/posters/`.

[C5] Davide Bacco, Matteo Canale, Nicola Laurenti, Giuseppe Vallone, and Paolo Villoresi, "Extracting secure keys from a limited number of qubits and high noise," to appear in Open Systems and Information Dynamics Journal, 5th LFPPI Symposium seQre2014 - Progress on Quantum Cryptography.

In preparation

[P1] Giuseppe Vallone, Davide Marangon, Matteo Canale , Ilaria Savorgnan, Davide Bacco, Mauro Barbieri, Simon Calimani, Cesare Barbieri, Nicola Laurenti, and Paolo Villoresi. Daylight Quantum Key Distribution in Long Distance Free-Space Links by Noise Mitigation using Turbulence as a Resource.

Bibliografia

[1] M. Troyer and R. Renner, "A randomness extractor for the Quantis device," tech. rep., 2012.

[2] T. M. Cover and A. J. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2nd ed., 2006.

[3] M. Nielsen and I. L. Chuang, *Quantum Computation And Quantum Information*. Oress Syndacate of the University of Cambridge, 2000.

[4] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden, "Optical Quantum Random Number Generator," p. 3, July 1999.

[5] A. Rukhin, J. Soto, J. Nechvatal, E. Barker, S. Leigh, M. Levenson, D. Banks, A. Heckert, J. Dray, S. Vo, M. Smid, M. Vangel, and L. E. Bassham Iii, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., 2001.

[6] R. Renner, *Security of Quantum Key Distribution*. Ph.d. thesis, ETH Zurich, 2005.

[7] A. Barvinok, "Matrices with prescribed row and column sums," *arXiv:1010.5706*, p. 30, Oct. 2010.

[8] T. Holenstein and R. Renner, "On the Randomness of Independent Experiments," *IEEE Transactions on Information Theory*, vol. 57, pp. 1865–1871, Apr. 2011.

[9] A. De, C. Portmann, T. Vidick, and R. Renner, "Trevisan's Extractor in the Presence of Quantum Side Information," *SIAM Journal on Computing*, vol. 41, pp. 915–940, Jan. 2012.

[10] C. Cachin, *Entropy measures and unconditional security in cryptography*. PhD thesis, ETH Zurich, 1997.

[11] R. Shaltiel, *Explicit construction of pseudo-random generators and extractors*. PhD thesis, 2001.

[12] R. Shaltiel, "An introduction to randomness extractors," *http://cs.haifa.ac.il/~ronen/online_papers/ICALPinvited.pdf*, 2011.

[13] M. Santha and U. V. Vazirani, "Generating quasi-random sequences from semi-random sources," *Journal of Computer and System Sciences*, vol. 33, pp. 75–87, Aug. 1986.

[14] S. P. Vadhan, "Randomness Extractors," in *Foundations and Trends in Theoretical Computer Science (Draft)*, 2011.

[15] R. Shaltiel, "Weak Derandomization of Weak Algorithms: Explicit Versions of Yao's Lemma," in *2009 24th Annual IEEE Conference on Computational Complexity*, pp. 114–125, IEEE, July 2009.

[16] J. Radhakrishnan and A. Ta-Shma, "Bounds for Dispersers, Extractors, and Depth-Two Superconcentrators," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 1, p. 2, 2000.

[17] R. T. Konig and B. M. Terhal, "The Bounded-Storage Model in the Presence of a Quantum Adversary," *IEEE Transactions on Information Theory*, vol. 54, pp. 749–762, Feb. 2008.

[18] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf, "Exponential separations for one-way quantum communication complexity, with applications to cryptography," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*, (New York, New York, USA), p. 516, ACM Press, 2007.

[19] R. Renner, "Quantum-Resilient Randomness Extraction," in *International Conference on Information Theoretic Security (ICITS) 2011*, vol. 17, pp. 52–57, Springer Berlin Heidelberg, Jan. 2011.

[20] A. Rao, "An Exposition of Bourgain's 2-Source Extractor," *Electronic Colloquim on Computational Complexity (ECCC)*, 2007.

[21] C. H. Bennett, G. Brassard, C. Crepeau, and U. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.

[22] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *Proceedings of the twenty-first annual ACM symposium on Theory of computing - STOC '89*, (New York, New York, USA), pp. 12–24, ACM Press, 1989.

[23] R. Konig, U. Maurer, and R. Renner, "On the Power of Quantum Memory," *IEEE Transactions on Information Theory*, vol. 51, pp. 2391–2401, July 2005.

[24] R. Renner and R. König, "Universally composable privacy amplification against quantum adversaries," *arXiv:quant-ph/0403133*, p. 14, Mar. 2004.

[25] D. R. Stinson, "Universal hash families and the leftover hash lemma, and applications to cryptography and computing," *Journal of Combinatorial Mathematics and Combinatorial Computing*, 2002.

[26] M. Tomamichel, C. Schaffner, A. Smith, and R. Renner, "Leftover Hashing Against Quantum Side Information," *IEEE Transactions on Information Theory*, vol. 57, pp. 5524–5535, Aug. 2011.

[27] J. Naor and M. Naor, *Small-bias probability spaces: efficient constructions and applications*. New York, New York, USA: ACM Press, 1990.

[28] S. Fehr and C. Schaffner, "Randomness extraction via δ-biasedmasking in the presence of a quantum attacker," in *TCC'08 Proceedings of the 5th conference on Theory of Cryptography* (LNCS: Lecture Notes In Computer Science, ed.), vol. 4948, pp. 465–481, Springer-Verlag Berlin, Heidelberg 2008, 2008.

[29] L. Trevisan, "Extractors and pseudorandom generators," *Journal of the ACM*, vol. 48, pp. 860–879, July 2001.

[30] A. De and T. Vidick, "Near-optimal extractors against quantum storage," in *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, (New York, New York, USA), p. 161, ACM Press, 2010.

[31] A. Ta-Shma, "Short seed extractors against quantum storage," in *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09*, (New York, New York, USA), p. 401, ACM Press, 2009.

[32] S. P. Vadhan, "On Constructing Locally Computable Extractors and Cryptosystems in the Bounded Storage Model," *Applied Sciences*, 2003.

[33] R. Konig and R. Renner, "Sampling of Min-Entropy Relative to Quantum Knowledge," *IEEE Transactions on Information Theory*, vol. 57, pp. 4760–4787, July 2011.

[34] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, pp. 143–154, Apr. 1979.

[35] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.

[36] H. Krawczyk, "New Hash Functions for Message Authentication," *Advances in Cryptology - EUROCRYPT '95*, vol. 921, pp. 301–310, 1995.

[37] G. Golub and C. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd ed., 1996.

[38] B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," in *Cryptographic Hardware and Embedded Systems - CHES 2003* (C. D. Walter, c. K. Koç, and C. Paar, eds.), vol. 2779 of *Lecture Notes in Computer Science*, (Berlin, Heidelberg), Springer Berlin Heidelberg, 2003.

[39] N. Nisan and D. Zuckerman, "Randomness is linear in space," *Journal of Computer and System Sciences*, vol. vol. 52, no. 1, pp. 43–52, 1996.

[40] R. Raz, O. Reingold, and S. Vadhan, "Extracting all the Randomness and Reducing the Error in Trevisan's Extractors," *Journal of Computer and System Sciences*, vol. 65, pp. 97–128, Aug. 2002.

[41] W. Mauerer, C. Portmann, and V. B. Scholz, "A modular framework for randomness extraction based on Trevisan's construction," *arXiv:1212.0520*, p. 21, Dec. 2012.

[42] ID Quantique, "Random Number Generation using Quantum Physics," tech. rep., 2010.

[43] ID Quantique, "Randomness extraction for the Quantis true random number generator," tech. rep., 2012.

[44] D. G. Marangon, "Study and analysis of a fast quantum random number generator for the Quantum Key Distribution," *Master thesis, University of Padova*, 2011.

[45] D. Frauchiger, R. Renner, and M. Troyer, "True randomness from realistic quantum devices," *arXiv:1311.4547v1*, pp. 1–20.

[46] J. Von Neumann, "Various techniques used in connection with random digits," *Journal of Research of NIST*, vol. 12, pp. 36–38, 1951.

[47] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[48] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, Nov. 1976.

[49] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[50] V. Scarani and C. Kurtsiefer, "The black paper of quantum cryptography: real implementation problems," *arXiv:0906.4547*, pp. 1–6, 2009.

[51] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of Modern Physics*, vol. 81, pp. 1301–1350, Sept. 2009.

[52] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical Review Letters*, vol. 68, no. 21, pp. 3121–3124, 1992.

[53] C. H. Bennett and G. Brassard, "Quantum cryptography: Public-key distribution and coin tossing," in *IEEE International Conference on Computers, Systems and Signal Processing*, (Bangalore), pp. 175–179, IEEE Computer Society, Dec. 1984.

[54] A. Ekert, "Quantum cryptography based on Bell's theorem," *Physical Review Letters*, vol. 67, no. 6, pp. 661–663, 1991.

[55] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without Bell's theorem," *Physical Review Letters*, vol. 68, no. 5, pp. 557–559, 1992.

[56] L. Masanes, S. Pironio, and A. Acín, "Secure device-independent quantum key distribution with causally independent measurement devices.," *Nature communications*, vol. 2, p. 238, Jan. 2011.

[57] C. C. W. Lim, C. Portmann, M. Tomamichel, R. Renner, and N. Gisin, "Device-Independent Quantum Key Distribution with Local Bell Test," *Physical Review X*, vol. 3, p. 031006, July 2013.

[58] S. Pironio, A. Acín, N. Brunner, N. Gisin, S. Massar, and V. Scarani, "Device-independent quantum key distribution secure against collective attacks," *New Journal of Physics*, vol. 11, p. 045021, Apr. 2009.

[59] D. Elser, T. Bartley, B. Heim, C. Wittmann, D. Sych, and G. Leuchs, "Feasibility of free space quantum key distribution with coherent polarization states," *New Journal of Physics*, vol. 11, p. 045014, Apr. 2009.

[60] F. Grosshans, G. Van Assche, J. Wenger, R. Brouri, N. J. Cerf, and P. Grangier, "Quantum key distribution using gaussian-modulated coherent states.," *Nature*, vol. 421, pp. 238–41, Jan. 2003.

[61] K. Inoue, E. Waks, and Y. Yamamoto, "Differential-phase-shift quantum key distribution using coherent light," *Physical Review A*, vol. 68, p. 022317, Aug. 2003.

[62] D. Stucki, N. Brunner, N. Gisin, V. Scarani, and H. Zbinden, "Fast and simple one-way quantum key distribution," *Applied Physics Letters*, vol. 87, no. 19, p. 194108, 2005.

[63] M. Westmoreland and B. Schumacher, "Capacities of Quantum Channels and Quantum Coherent Information," in *Lecture Notes in Computer Science*, vol. 1509, pp. 285–295, Springer Berlin Heidelberg, 1998.

[64] G. Cariolaro, *Comunicazioni quantistiche*. Cartoleria Portello, 2009.

[65] B. A. Slutsky, R. Rao, P.-C. Sun, L. Tancevski, and Y. Fainman, "Defense frontier analysis of quantum cryptographic systems.," *Applied optics*, vol. 37, pp. 2869–78, May 1998.

[66] W. K. Wooters and W. Zurek, "A single quantum cannot be cloned," *Nature*, no. 299, pp. 802–803, 1982.

[67] N. Lütkenhaus, "Security against individual attacks for realistic quantum key distribution," *Physical Review A*, vol. 61, no. 5, 2000.

[68] E. Biham and T. Mor, "Security of Quantum Cryptography against Collective Attacks," *Physical Review Letters*, vol. 78, no. 11, pp. 2256–2259, 1997.

[69] J. Müller-Quade and R. Renner, "Composability in quantum cryptography," *New Journal of Physics*, vol. 11, p. 085006, Aug. 2009.

[70] R. König, R. Renner, A. Bariska, and U. M. Maurer, "Small Accessible Quantum Information Does Not Imply Security," *Physical Review Letters*, vol. 98, p. 140502, Apr. 2007.

[71] R. Renner and S. Wolf, "Simple and Tight Bounds for Information Reconciliation and Privacy Amplification," *International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology, ASIACRYPT*, vol. 3788, pp. 199–216, 2005.

[72] S. Abruzzo, H. Kampermann, M. Mertz, and D. Bruß, "Quantum key distribution with finite resources: Secret key rates via Rényi entropies," *Physical Review A*, vol. 84, p. 032321, Sept. 2011.

[73] S. Bratzik, M. Mertz, H. Kampermann, and D. Bruß, "Min-entropy and quantum key distribution: Nonzero key rates for "small" numbers of signals," *Physical Review A*, vol. 83, pp. 1–9, Feb. 2011.

[74] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, "Tight finite-key analysis for quantum cryptography," *Nature Communications*, vol. 3, p. 634, Jan. 2012.

[75] O. Hirota, "Incompleteness and Limit of Quantum Key Distribution Theory," *arXiv:1208.2106*, pp. 1–11, Aug. 2012.

[76] H. P. Yuen, "Unconditional Security In Quantum Key Distribution," *arXiv:1205.5065*, pp. 1–13, May 2012.

[77] R. Renner, "Reply to recent scepticism about the foundations of quantum cryptography," *arXiv:1209.2423*, vol. 2, p. 1, Sept. 2012.

[78] O. Hirota, "Misconception in Theory of Quantum Key Distribution - Reply to Renner," *arXiv:1306.1277*, pp. 1–7, June 2013.

[79] M. Wilde, *Quantum Information Theory*. Cambridge University Press, 2013.

[80] D. Bruss, "Optimal Eavesdropping in Quantum Cryptography with Six States," *Physical Review Letters*, vol. 81, pp. 3018–3021, Oct. 1998.

[81] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, "Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations," *Physical Review Letters*, vol. 92, p. 057901, Feb. 2004.

[82] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, 2002.

[83] P. W. Shor and J. Preskill, "Simple Proof of Security of the BB84 Quantum Key Distribution Protocol," *Physical Review Letters*, vol. 85, pp. 441–444, July 2000.

[84] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 1–28, 1992.

[85] M. Dušek, M. Jahma, and N. Lütkenhaus, "Unambiguous state discrimination in quantum cryptography with weak coherent states," *Physical Review A*, vol. 62, pp. 1–9, July 2000.

[86] M. Dušek, N. Lütkenhaus, and M. Hendrych, "Quantum Cryptography," in *Progress in Optics*, vol. 49, p. 381, 2006.

[87] M. Lucamarini, G. Di Giuseppe, and K. Tamaki, "Robust unconditionally secure quantum key distribution with two nonorthogonal and uninformative states," *Physical Review A*, vol. 80, pp. 1–7, Sept. 2009.

[88] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, "Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations," *Physical Review Letters*, vol. 92, p. 057901, Feb. 2004.

[89] W.-Y. Hwang, "Quantum Key Distribution with High Loss: Toward Global Secure Communication," *Physical Review Letters*, vol. 91, Aug. 2003.

[90] H.-K. Lo, X. Ma, and K. Chen, "Decoy State Quantum Key Distribution," *Physical Review Letters*, vol. 94, no. 23, 2005.

[91] T. Schmitt-Manderbach, H. Weier, M. Fürst, R. Ursin, F. Tiefenbacher, T. Scheidl, J. Perdigues, Z. Sodnik, C. Kurtsiefer, J. Rarity, A. Zeilinger, and H. Weinfurter, "Experimental Demonstration of Free-Space Decoy-State Quantum Key Distribution over 144Â km," *Physical Review Letters*, vol. 98, pp. 1–4, Jan. 2007.

[92] E. Meyer-Scott, Z. Yan, A. MacDonald, J.-p. Bourgoin, H. Hübel, and T. Jennewein, "How to implement decoy-state quantum key distribution for a satellite uplink with 50 dB channel loss," *System*, p. 8, Nov. 2011.

[93] D. Rosenberg, C. G. Peterson, J. W. Harrington, P. R. Rice, N. Dallmann, K. T. Tyagi, K. P. McCabe, S. Nam, B. Baek, R. H. Hadfield, R. J. Hughes, and J. E. Nordholt, "Practical long-distance quantum key distribution system using decoy levels," *New Journal of Physics*, vol. 11, p. 045009, Apr. 2009.

[94] J. Hasegawa, M. Hayashi, T. Hiroshima, A. Tanaka, and A. Tomita, "Experimental Decoy State Quantum Key Distribution with Unconditional Security Incorporating Finite Statistics," *arXiv:0705.3081*, p. 5, May 2007.

[95] P. Rice and J. Harrington, "Numerical analysis of decoy state quantum key distribution protocols," *arXiv:0901.0013v2*, pp. 1–30, 2009.

[96] G. Brassard and L. Salvail, "Secret-Key Reconciliation by Public Discussion," *International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology, EUROCRYPT*, pp. 410–423, 1993.

[97] 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*, vol. 67, p. 052303, May 2003.

[98] D. Elkouss, A. Leverrier, R. Alléaume, and J. J. Boutros, "Efficient reconciliation protocol for discrete-variable quantum key distribution," in *IEEE International Symposium on Information Theory, ISIT*, pp. 1879–1883, IEEE, 2009.

[99] J. Lodewyck, M. Bloch, R. García-Patrón, S. Fossier, E. Karpov, E. Diamanti, T. Debuisschert, N. J. Cerf, R. Tualle-Brouri, S. McLaughlin, and P. Grangier, "Quantum key distribution over 25km with an all-fiber continuous-variable system," *Physical Review A*, vol. 76, pp. 1–10, Oct. 2007.

[100] S. Watanabe, R. Matsumoto, T. Uyematsu, and Y. Kawano, "Key rate of quantum key distribution with hashed two-way classical communication," *Physical Review A*, vol. 76, p. 032312, Sept. 2007.

[101] J. Bae and A. Acín, "Key distillation from quantum channels using two-way communication protocols," *Physical Review A*, vol. 75, p. 012334, Jan. 2007.

[102] G. V. Assche, *Quantum Cryptography and Secret-Key Distillation*. Cambridge University Press, 2006.

[103] N. Benvenuto, R. Corvaja, T. Erseghe, and N. Laurenti, *Communication systems: fundamentals and design methods*. John Wiley & Sons, 2007.

[104] M. Mondin, M. Delgado, F. Mesiti, and F. Daneshgaran, "Soft-processing for Information Reconciliation in QKD Applications," *International Journal of Quantum Information*, 2010.

[105] W. Traisilanun, K. Sripimanwat, and O. Sangaroon, "Secret key reconciliation using BCH code in quantum key distribution," in *International Symposium on Communications and Information Technologies, ISCIT*, pp. 1482–1485, IEEE, 2007.

[106] D. Elkouss, J. Martinez, D. Lancho, and V. Martin, "Rate compatible protocol for information reconciliation: An application to QKD," in *IEEE Information Theory Workshop 2010 (ITW 2010)*, pp. 1–5, IEEE, Jan. 2010.

[107] D. Elkouss, J. Martinez-Mateo, and V. Martin, "Efficient Reconciliation with Rate Adaptive Codes in Quantum Key Distribution," *Quantum Information and Computation*, vol. 0, pp. 1–14, July 2010.

[108] D. S. Kim, "Weight Distributions of Hamming Codes," pp. 1–3, Oct. 2007.

[109] R. G. Gallager, "Low-Density Parity-Check Codes," *PhD Thesis*, 1963.

[110] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. McLaughlin, and J.-M. Merolla, "Applications of LDPC Codes to the Wiretap Channel," p. 30, 2004.

[111] J. a. Barros, M. Bloch, M. R. D. Rodrigues, and S. W. McLaughlin, "LDPC-Based Secure Wireless Communication with Imperfect Knowledge of the Eavesdropper's Channel," *IEEE Information Theory Workshop, ITW*, pp. 155–159, Oct. 2006.

[112] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, *LDPC-Based Secure Wireless Communication with Imperfect Knowledge of the Eavesdropper's Channel*. IEEE, Oct. 2006.

[113] R. Matsumoto, "Problems in application of LDPC codes to information reconciliation in quantum key distribution protocols," *arXiv:0908.2042*, p. 10, 2009.

[114] R. Storn and K. Price, "Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," in *Technical Report TR-95-012*, 1995.

[115] M. Mondin, M. Delgado, and F. Mesiti, "Novel Techniques for Information Reconciliation, Quantum Channel Probing and Link Design for Quantum Key Distribution," in *Personal Satellite Services, PSATS*, pp. 1–12, 2010.

[116] M. Tomamichel, J. Martinez-Mateo, C. Pacher, and D. Elkouss, "Fundamental Finite Key Limits for Information Reconciliation in Quantum Key Distribution," *arXiv:1401.5194*, pp. 1–11, Jan. 2014.

[117] Y. Polyanskiy, *Channel coding : non-asymptotic fundamental limits*. PhD thesis, Princeton University, 2010.

[118] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," *IEEE Transactions on Information Theory*, vol. 56, pp. 2307–2359, May 2010.

[119] D. Elkouss, J. Martinez-Mateo, and V. Martin, "Untainted Puncturing for Irregular Low-Density Parity-Check Codes," p. 3, Mar. 2011.

[120] E. Eleftheriou and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, pp. 386–398, Jan. 2005.

[121] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Blind reconciliation," *Quantum Information and Computation*, vol. 12, no. 9-10, pp. 791–812, 2012.

[122] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, "Tight Finite-Key Analysis for Quantum Cryptography," *arXiv:1103.4130*, Mar. 2011.

[123] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy Amplification by Public Discussion," *SIAM Journal on Computing*, vol. 17, no. 2, p. 210, 1988.

[124] C.-H. F. Fung, X. Ma, and H. F. Chau, "Practical issues in quantum-key-distribution postprocessing," *Physical Review A*, vol. 81, no. 1, 2010.

[125] P. Villoresi, T. Jennewein, F. Tamburini, M. Aspelmeyer, C. Bonato, R. Ursin, C. Pernechele, V. Luceri, G. Bianco, a. Zeilinger, and C. Barbieri, "Experimental verification of the feasibility of a quantum channel between space and Earth," *New Journal of Physics*, vol. 10, p. 033038, Mar. 2008.

[126] C. Bonato, A. Tomaello, V. Da Deppo, G. Naletto, and P. Villoresi, "Feasibility of satellite quantum key distribution," *New Journal of Physics*, vol. 11, p. 045017, Apr. 2009.

[127] A. Tomaello, C. Bonato, V. Da Deppo, G. Naletto, and P. Villoresi, "Link budget and background noise for satellite quantum key distribution," *Advances in Space Research*, vol. 47, pp. 802–810, Mar. 2011.

[128] V. Scarani and R. Renner, "Quantum Cryptography with Finite Resources: Unconditional Security Bound for Discrete-Variable Protocols with One-Way Postprocessing," *Physical Review Letters*, vol. 100, pp. 200501.1–5, May 2008.

[129] R. Y. Q. Cai and V. Scarani, "Finite-key analysis for practical implementations of quantum key distribution," *New Journal of Physics*, vol. 11, p. 045024, Apr. 2009.

[130] V. Scarani, "QKD: a million signal task," in *NATO Advanced Research Workshop on Quantum Cryptography and Computing*, (Gdansk), p. 7, Oct. 2010.

[131] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications, 1964.

[132] R. J. Hughes, J. E. Nordholt, D. Derkacs, and C. G. Peterson, "Practical free-space quantum key distribution over 10 km in daylight and at night," *New Journal of Physics*, vol. 4, pp. 43–43, July 2002.

[133] S. Nauerth, F. Moll, M. Rau, C. Fuchs, J. Horwath, S. Frick, and H. Weinfurter, "Air-to-ground quantum communication," *Nature Photonics*, vol. 7, pp. 382–386, Mar. 2013.

[134] "European Quantum Information Processing and Communication Roadmap. Revision of February 2013.."

[135] "Japanese Quantum Information Roadmap (2010)."

[136] "USA roadmap for the free-space links (2004)."

[137] M. Aspelmeyer, T. Jennewein, M. Pfennigbauer, W. Leeb, and A. Zeilinger, "Long-distance quantum communication with entangled photons using satellites," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 9, pp. 1541–1551, Nov. 2003.

[138] J. E. Nordholt, R. J. Hughes, G. L. Morgan, C. G. Peterson, and C. C. Wipf, "Present and future free-space quantum key distribution," in *roc. SPIE 4635* (G. S. Mecherle, ed.), pp. 116–126, Apr. 2002.

[139] C.-Z. Peng, T. Yang, X.-H. Bao, J. Zhang, X.-M. Jin, F.-Y. Feng, B. Yang, J. Yang, J. Yin, Q. Zhang, N. Li, B.-L. Tian, and J.-W. Pan, "Experimental Free-Space Distribution of Entangled Photon Pairs Over 13 km: Towards Satellite-Based Global Quantum Communication," *Physical Review Letters*, vol. 94, p. 150501, Apr. 2005.

[140] F. Gerlin, N. Laurenti, G. Naletto, G. Vallone, P. Villoresi, L. Bonino, S. Mottini, and Z. Sodnik, "Design optimization for quantum communications in a GNSS intersatellite network," in *2013 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–6, IEEE, June 2013.

[141] S. Calimani, "Unconditionally secure authentication for quantum key distribution," *Master thesis, University of Padova*, 2011.

[142] J. Bierbrauer, T. Johansson, G. A. Kabatianskii, and B. J. M. Smeets, "On families of hash functions via geometric codes and concatenation," in *Advances in Cryptology, CRYPTO '93*, pp. 331–342, 1993.

[143] D. R. Stinson, "Universal hashing and authentication codes," *Designs, Codes and Cryptography*, vol. 4, pp. 369–380, July 1994.

[144] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, pp. 265–279, June 1981.

[145] A. Abidin, "Weaknesses of Authentication in Quantum Cryptography and Strongly Universal Hash Functions," *Master thesis, Linköping University*, 2010.

[146] B. Huttner, N. Imoto, N. Gisin, and T. Mor, "Quantum cryptography with coherent states," *Physical Review A*, vol. 51, pp. 1863–1869, Mar. 1995.

[147] C. Elliott, D. Pearson, and G. Troxel, "Quantum cryptography in practice," *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, p. 227, 2003.

[148] B. C. Jacobs and J. D. Franson, "Quantum cryptography in free space.," *Optics letters*, vol. 21, pp. 1854–6, Nov. 1996.

[149] W. T. Buttler, R. J. Hughes, P. Kwiat, S. K. Lamoreaux, G. Luther, G. Morgan, J. E. Nordholt, C. G. Peterson, and C. Simmons, "Practical Free-Space Quantum Key Distribution over 1 km," *Physical Review Letters*, vol. 81, pp. 3283–3286, Oct. 1998.

[150] M. Peev, A. Poppe, O. Maurhart, T. Lorünser, T. Länger, and C. Pacher, "The SECOQC quantum key distribution network in Vienna," in *European Conference on Optical Communication, ECOC*, vol. 11, pp. 1–4,20–24, July 2009.

[151] M. J. García-Martínez, N. Denisenko, D. Soto, D. Arroyo, a. B. Orue, and V. Fernandez, "High-speed free-space quantum key distribution system for urban daylight applications.," *Applied optics*, vol. 52, pp. 3311–7, May 2013.

[152] I. Capraro, A. Tomaello, A. Dall'Arche, F. Gerlin, R. Ursin, G. Vallone, and P. Villoresi, "Impact of Turbulence in Long Range Quantum and Classical Communications," *Physical Review Letters*, vol. 109, p. 200502, Nov. 2012.

[153] I. Savorgnan, "Exploiting Turbulence to increase Quantum Key Distribution feasibility over free-space channels," *Master thesis, University of Padova*, 2013.

[154] C. Erven, B. Heim, E. Meyer-Scott, J.-P. Bourgoin, R. Laflamme, G. Weihs, and T. Jennewein, "Studying free-space transmission statistics and improving free-space quantum key distribution in the turbulent atmosphere," *New Journal of Physics*, vol. 14, p. 123018, Dec. 2012.