UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di laurea Magistrale in Ingegneria delle Telecomunicazioni

Exploiting Turbulence to increase Quantum Key Distribution feasibility over free-space channels

Laureanda:
Ilaria Savorgnan

Relatore:
Prof. Nicola Laurenti
Correlatore:
Dott. Matteo Canale

Anno Accademico 2012-2013

Contents

1 Introduction

The establishment of a *secret transmission* is one of the most relevant issues in modern communications. The growing request and research for increasingly sophisticated security mechanisms to protect the information exchanges gave birth to a variety of key agreement mechanisms. The expanding technological innovations will bring, in the near future, the necessity to guarantee the establishment of a secure system between devices experiencing outstanding spatial separations: the first one being the satellite communications. These new developing horizons required to merge two distinct disciplines: the **Classical** and **Quantum Information Theory**. Thanks to the exploitation of Quantum Mechanics laws, and of the concept of photon -an information bearer with amazing properties- the realization of a secure communication establishment is possible, and the detection of a malicious entity can be properly realized. Neverthless, we cannot lose consciousness of the fact that, by increasing the distance between devices, we have a consequential enlargement of the complications that the trasmission itself has to dam up: among them, the **atmospherical turbulence** problem. For years, the Communication community acted against this bothering component, with the aim of realizing solutions to combat its effects: over time, significant results have been presented, nevertheless the problem -and the turbulence itself- still persists. In this work we will expose an innovative and refined way to allow secure communications even in malevolent situations: the turbulence will be *exploited*, thanks to the following criterion: the presence of atmospherical turbulence implies that the channel experienced by the signal exhibits occasional high gains rather than a constant amplitude. This in turn means that such transmettivity peaks could be exploited with the aim of improving the quality of our transmission. This can be done by using a suitable probing signal that will provide the receiver with the required information about the channel status: we will choose to acquire the key signal in those instants for which the probe signal lies above a given threshold. Therefore, we will obtain a preselection of packets, that will be numerically lower, but at the same time we will experience an improvement on the *SNR* and on the quality of the transmission itself: the idea is that the *SNR* increase should outpace the decrease of the overall counts in a given time. We will show that this expedient provides some impressive results, in particular for acquisitions with high background noise conditions: that is, transmissions for which we previously had considerable limitations regarding the possibility of completing a secret key exchange successfully.

In **Chapter 2** we will introduce the basis of Quantum Information Theory: the first step is an overview of the Information Theory basics, briefly focusing on source coding and channel coding theorems, and on cardinal concepts such as entropy, mutual information and perfect secrecy. In the second part we will familiarize with the fundamentals of quantum mechanics, and with the laws that allowed a growing development of the field of Quantum Cryptography, making it essentially the only practical tool for physical layer information theoretic security. To close the chapter, we will focus on Cryptographic Security Mechanisms such as encryption, unconditionally secure authentication and key agreement; all these topics will be analyzed in details.

Chapter 3 is devoted to Quantum Key Distribution, that is, a technique that uses the main principles of quantum mechanics in order to guarantee a secure communication between two parties, generally called Alice and Bob. We will present key exchange and key distillation, including the crucial phases of sifting, reconciliation, privacy amplification. This is done with the purpose of agreeing on a common yet secret key, thus ensuring the confidentiality of communications.

Chapters 4,5 and **6** concern experimental realization of quantum key exchange across a turbulent free space quantum channel at the Canary Islands and the data processing and analysis that provide the final results. Specifically, Chapter 4 introduces the theoretical background needed to fully understand the innovative idea of *exploiting the turbulence*; we will provide a sound motivation to the intuitions and the theoretical model underlying this experiment. Chapter 5 is dedicated to the exposition of the functions implementing the processing of acquired data; a focus on each function, attempting to provide an overall comprehension to the reader, is thus proposed. Chapter 6 finally presents the remarkable results that allow us to obtain some impressive conclusions and to suggest the *turbulence exploiting approach* as a possible way to enhance the quality of long-distance communications.

2 | Quantum Information Theory

2.1 Motivation

The outstanding development of network technologies, the persistent growth of scientific and technical knowledge and the increasing demand for systems able to guarantee the confidentiality and integrity of informations carried across the network, gave rise to the need for establish new foundation for cryptography science.

This led to the development of *Quantum Cryptography*, which lies at the intersection of two cardinal branches of science, as it combines the processing techniques of *Information Theory* with the inviolable laws of *Quantum Mechanics*.

2.2 Basics of Information Theory

Information Theory, the mathematical theory of communication, has two predominant intents:

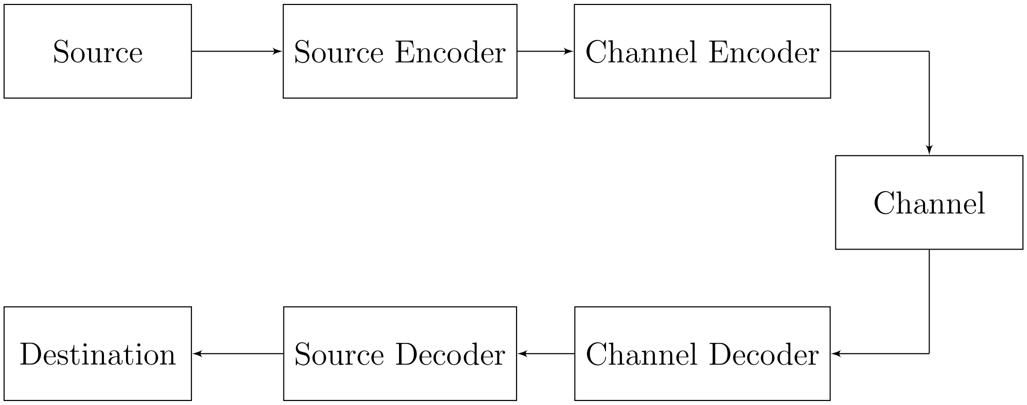
- the derivation of the fundamental theoretical limits on the achievable performance, when communicating information from a given source over a given channel with the use of coding schemes from within a prescribed class.

- The development of coding schemes providing performance that is reasonably good in comparison with the optimal performance given by the theory [1].

This remarkable discipline was developed by Claude Shannon at Bell Labs in 1940. With Shannon, the information is represented by signals, which in turn are carriers of informative contents.

The aim of Shannon was to characterize and define the quantity of information emitted by a source and to discover a way to represent the informative content by the signals, so that the information remains undistorted even if the transmitted signals are corrupted by noise.

A communication system as described by *Information Theory* is depicted below [2]:



In the following, the block diagram will be exploited in order to present the basic results of Information Theory: *The Source Coding Theorem* and *The Channel Coding Theorem*.

2.2.1 Source and Information

The aim of describing mathematically the process under investigation gives rise to the necessity of defining a suitable quantity as well as a unit to measure it: so Shannon introduced the quantity of **Information** in a message. For a discrete source with finite number of messages he defined the information content of the symbol x as the logarithm of the inverse of its probability:

$$i(x) = \log\left(\frac{1}{p(x)}\right) = -\log(p(x)) \tag{2.1}$$

The convention of using a base-two logarithm implies that the units of this measure are bits.

Another fundamental quantity is **Entropy**, that can be thought as a measure of our uncertainty about which symbol will be chosen and emitted by the source: for a source with higher entropy, this uncertainty is higher [2].

The **Entropy** of the discrete random variable X is defined as:

$$H(X) = -\sum_{x} p_X(x) \log_2(p_X(x)) \tag{2.2}$$

where $p_X(x)$ is the probability mass function associated with random variable X [3].

2.2.2 Source coding

The task of the *Source encoder* block is to represent (encode) the information by signals in an proficient way; the aim is to compress data in order to use the system more efficiently.

In source coding, the *Entropy* of a random variable tells us precisely about its compressibility. If we assume a finite range $\mathcal{X}$, we could use $\lceil \log |X| \rceil$ bits per symbol to encode the random variable X: nevertheless, this is not optimal if some symbols are more frequent than others. A way to solve this problem is to encode more frequent symbols using fewer bits and vice versa. We define the encoding rate R as the average number of bits per symbol. With this concept in mind, we can express one of the fundamental results of Shannon [4]:

Theorem 1 (Source Coding Theorem)**.**

It is not possible to compress a source with a Rate R lower than its Entropy:

$$H(X) \leq R \tag{2.3}$$

The cardinal concept of the **Source Coding Theorem** is that the average number of bits needed to encode the source symbol can be made as small as *Entropy*, but not smaller.

The source encoding is accomplished by giving the shorter code words to the symbols with higher probabilities. One of the drawbacks of source encoding is that when one received symbol is in error, the synchronisation between encoder and decoder will be lost for some time, resulting in a series of erroneously decoded symbols. A lot of efforts have been done to find the suitable code words so as that resynchronisation is obtained as fast as possible [2].

2.2.3 Channel coding

The task of the *Channel encoder* block is to represent the information by the channel signals is such a way that no information is lost if the signals are distorted and even if some finite error probability exists. The target is to find the most efficient way to transmit information over a potentially noisy channel. Therefore, it's necessary to take into consideration the tradeoff between:

- optimizing the *transmission rate*
- maintaining the *transmission reliability*

In order to meet the first requirement, we want to transmit as many symbols as possible per channel use, while for the second one we would like to have a vanishingly small probability that the message arrives inconsistently at the output of the channel [4].

The main result showed by Shannon is that the error probability can be made as small as we wish if the information flow is smaller then the **Channel Capacity**, a quantity that is characteristic of the channel.

For the additive white noise Gaussian channel, the capacity only depends on the bandwidth and the signal-to-noise ratio. Denoting the signal power by P, the noise power by N, and the bandwidth by B, the resulting capacity C is [2]:

$$C = B \log_2 \left(1 + \frac{P}{N}\right) \tag{2.4}$$

In principle, a channel with Capacity C is capable, with suitable coding, of transmitting at any rate less than C bits per symbol with vanishingly small probability of error. For rates greater than C the probability of error cannot be made arbitrarily small.

Another crucial notion of information used by Shannon was **Mutual Information**. *Entropy* is in fact a notion of self information: the information that a random process provides by itself. **Mutual information** is instead a measure of the information contained in one process about another process. While entropy is sufficient to study the reproduction of a single process through a noiseless environment, more often one has two or more distinct random processes, e.g., one random process representing an information source and another representing the output of a communication medium wherein the coded source has been corrupted by another random process called *noise*. In such cases observations are made on one process in order to make decisions on another [1]. Therefore, Shannon introduced the notion of **Mutual Information** between two processes:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{2.5}$$

that is, the sum of the two self entropies minus the entropy of the pair.

The quantity $H(X,Y)$ is called *Joint Entropy* and in the case of two random variables X and Y is defined as:

$$H(X,Y) = -\sum_{x,y} P_{XY}(x,y) \log P_{XY}(x,y) \tag{2.6}$$

where $P_{XY}(x,y)$ is the probability distribution of the joint random variable XY.

The **Mutual Information** can also be defined in terms of *Conditional Entropy* (or *equivocation*) $H(X|Y) = H(X,Y) - H(Y)$, which characterizes the compressibility of the variable X if Y is known to both the encoder and the decoder. Hence:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{2.7}$$

In this form the **Mutual Information** can be interpreted as the information born by one process minus the additional information it bears when the other process is already known [1].

Finally, Shannon defined the input distribution as $P_X(x)$ and showed this fundamental result [4]:

Theorem 2 (Channel Coding Theorem)**.**

The basic upper bound on the rate of transmission is $I(X,Y)$ bits per channel use. This maximization defines the Channel Capacity:

$$C = \max_{P_X(x)} I(X,Y) \tag{2.8}$$

2.2.4 Information Theoretic notion of secrecy

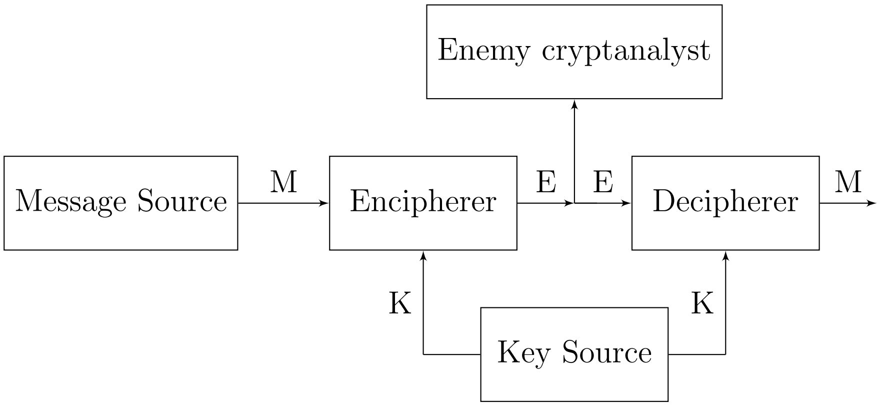
In 1949, Claude Shannon exploited the previous results in order to provide the information theory basis for secrecy: one of the main result is that the amount of secret information that can be introduced into an encoded message can't be greater than that of the *cryptographic key* used to encode it. In order to analyze in depth this idea, we have to introduce the concept of **Secrecy System**.

A **Secrecy System** is defined abstractly as a set of transformations of one space (the set of possible messages) into a second space (the set of possible cryptograms). Each particular transformation of the set corresponds to enciphering with a particular key. The transformations are supposed to be reversible (non-singular), so that unique deciphering is possible when the key is known.

Each key and therefore each transformation is assumed to have an *a priori probability* associated with it: the probability of choosing that key.

Similarly, each possible message is assumed to have an associated *a priori probability*, determined by the underlying stochastic process representative of the discrete sequence of symbols under consideration. These probabilities for the various keys and messages are also any attacking cryptanalyst's a priori probabilities for the choices in question, and represent his *a priori knowledge* of the situation.

A schematic diagram of a general secrecy system is shown below, where M is the message, E is the cryptogram, K is the key [5]:



At the transmitting end there are two information sources: a *message source* and a *key source*. The key source produces a particular key among those that are possible in the system. The selected key is sent to the receiver: the choice of a key determines a particular transformation in the set forming the system. Then a message is selected and the particular transformation corresponding to the chosen key is applied to the message to produce a cryptogram. So if M is the message, K

the key, and E the enciphered message, or cryptogram, we have that E is function of the other parameters:

$$E = f(M, K) = T_i M \tag{2.9}$$

where the second mathematical structure is introduced to emphatize the fact that the cryptogram E is the result of the transformation T_i applied to message M; the index i corresponds to the particular key being used.

The cryptogram is transmitted to the receiver and may be intercepted by an adversary; the receiver applies to the cryptogram the inverse of the chosen transformation, in order to recover the original message. So at the receiving end it must be possible to recover M, just knowing E and K. Thus the transformations T_i in the family must have unique inverses T_i^{-1} such that $T_i^{-1} T_i = I$, the identity transformation. In this manner:

$$M = T_i^{-1} E \tag{2.10}$$

If the adversary intercepts the cryptogram, he can calculate from it the *a posteriori probabilities* of the various possible messages and keys which might have produced it. This set of *a posteriori probabilities* constitutes the adversary's knowledge of the key and message after the interception.

Perfect Secrecy

A fundamental question could be how immune is a system to cryptanalysis when the cryptanalyst has unlimited time and power available for the analysis of cryptograms. Does a way to guarantee the system security -no matter what- actually exist?

Let's consider a finite number of possible messages $M_1 ... M_n$ and their correspondent *a priori probabilities* $P(M_1) ... P(M_n)$: this are enciphered into the possible cryptograms $E_1 ... E_m$ thanks to the relation 2.10. The cryptanalyst intercepts a particular E and can then calculate, in principle at least, the *a posteriori probabilities* for the various messages, $P_E(M)$.

The definition of **Perfect Secrecy** is given by the condition that, for all E the *a posteriori probabilities* are equal to the *a priori probabilities* independently of the values of these. This means that the cryptanalyst obtains no additional information by intercepting the message. The drawback is that, if this condition is not satisfied, there will exist situations in which the adversary has certain *a priori probabilities*, and certain key and message choices may occur for which the enemy's probabilities do change. This in turn may affect his actions and thus **Perfect Secrecy** has not been obtained [5].

In order to formalize this concept, let's define:

- $P_E(M) =$ *a posteriori* probability of message M if cryptogram E is intercepted.
- $P(M) =$ *a priori* probability of message M.
- $P_M(E) =$ *conditional* probability of cryptogram E if message M is chosen. This is the sum of the probabilities of all keys which produce cryptogram E from message M.
- $P(E) =$ probability of obtaining cryptogram E *from any message.*

By using the *Bayes' Theorem*, we have:

$$P_E(M) = \frac{P(M)P_M(E)}{P(E)} \tag{2.11}$$

For **Perfect Secrecy** we have said that $P_E(M)$ must equal $P(M)$ for all E and all M. Hence:

Theorem 3.

A necessary and sufficient condition for **Perfect Secrecy** is that:

$$P_M(E) = P(E) \tag{2.12}$$

for all M and E. This means that $P_M(E)$ must be independent of M.

This Theorem implies that an encryption scheme achieves *Perfect Secrecy* if, for any two messages M_i and M_j, any cryptogram E has the same probability of being the encryption of M_i as being the encryption of M_j.

Another fundamental assessment is that, since for a fixed i T_i gives a one-to-one correspondence between all the M's and some of the E's, there must be as many E's as there are M's. Hence there is at least one key transforming any M into any of these E's, but all the keys from a fixed M to different E's must be different, and therefore:

Lemma 4.

In order to obtain **Perfect Secrecy**, the number of different keys should be at least as great as the number of messages.

Furthermore, **Perfect Secrecy** can be achieved only when the secret key is at least as long as the plaintext message. Shannon proved this basic lemma, but he didn't say how to obtain such a long secret key. Anyway, this concept can be restrictive if we take into account that the secret key needs to be transmitted confidentially: if we have a private line to transmit it securely, and the key is of the same length of the message itself, it would be better to use that resource to forward directly the confidential information. For this reason, the accepted convention is to use a smaller secret key; this makes confidentiality achievable in practice but at the cost of a security's performance leak, as from Shannon's theory the security of this scheme cannot be perfect.

Leaving aside the problem of sending confidential information for now (we will come back to all these aspects in *Section 2.4*), another important argument concerns the physical carrying of information: a basic result is that a piece of information must be stored or written on a medium and hence must follow the laws of physics. In his theory, Shannon essentially assumes a classical physical support. When the medium is of atomic scale, such as photons, the carried information behaves quite differently, so it's necessary to introduce the basic law of *Quantum Mechanics* involved in this approach: this will be done in the following section [4].

2.3 Fundamentals of Quantum Mechanics

The idea of exploiting the fundamental principles of Quantum Mechanics and apply them to the rising field of **Quantum Cryptography** was first proposed by Stephen Wiesner (1983) and by Charles H. Bennett and Gilles Brassard (1984) [6].

At that time, the Information Security field was finally growing enough that physicists were ready to consider *Quantum Mechanics* as a tool for new engineering. The amazing theory governing this discipline is based on apparently negative rules about things that cannot be done: nevertheless, these rules opened an incredibly wide ranging area of applications and allowed a substantial increasing in the system performance.

We introduce this set of negative rules, that will be analyzed further:

- One cannot take a measurement without perturbing the system.
- One cannot determine simultaneously a couple of characteristics of a quantum object with arbitrarily high accurancy.
- One cannot duplicate an unknown quantum state.

In the following, we will investigate the fundamental principles of *Quantum Mechanics* that play a role in the developing of the new **Quantum Cryptography** field. But before that, the first thing that should be done when approaching the Quantum Physics discipline, is to focus on the preeminent definitions that could help our understanding of this challenging field of study.

2.3.1 Basic definitions

Quantum states

We can refer to a *quantum state* as the state of a quantum system. Substantially, a *quantum state* is a linear superposition of other quantum states, which means that a particle in one quantum state is also simultaneously in other quantum states. With the aim of examining in depth this concept, let's introduce a *complex Hilbert space* $\mathcal{H}$, which in quantum mechanics is used to describe a physical system. The state of this physical system is in turn any unit-sized vector denoted $|\psi\rangle$. With this notation in mind, we can describe the Hilbert space $\mathcal{H}$ as spanned by some orthonormal basis $\{|b\rangle\}$: the state $|\psi\rangle$ can then be written as a complex linear combination of the basis vectors [4]:

$$|\psi\rangle = \sum_b c_b |b\rangle \tag{2.13}$$

with: $c_b \in \mathbf{C}$ and $\sum_b |c_b|^2 = 1$. The decomposition coefficients c_b can be expressed as the inner product between $|b\rangle$ and $|\psi\rangle$, and denoted as: $c_b = \langle b|\psi\rangle$. The linear combination of *equation (2.13)* is then a *superposition*: this refers to the fact that any given quantum state can be decomposed into a linear *sum*, i.e. a superposition of eigenstates that are in a physical sense mutually incompatible, which in turn allows them to be mathematically modeled as orthogonal basis vectors in Hilbert space [7].

Qubits

We already know that the *bit* is used to model the smallest classical information unit; the physical representation of a classical bit is generally a system containing many atoms, and is described by one or more continuous parameters (such as voltages). In this circumstance two well-separated regions are chosen to represent 0 and 1, and signals are periodically restored toward these regions in order to avoid drifting caused by perturbations or imperfect conditions [8].

The quantum counterpart of the bit is the *qubit*, which models the smallest quantum information unit: it is typically a microscopic system, such as an atom or nuclear spin, or polarized photon. A *qubit* can be defined as a quantum system that lies in a two-dimensional Hilbert space: the basis is called the *computational basis* and denoted as $\{|0\rangle, |1\rangle\}$, in such a way that a qubit state can be described as $c_0|0\rangle + c_1|1\rangle$ with [4]:

$$|c_0|^2 + |c_1|^2 = 1 \tag{2.14}$$

This means that, according to quantum information theory, a qubit can be a linear superposition of the two classical states, with complex coefficients; anyway, from any given qubit we can extract no more than one classical bit of information, and the more information we obtain about it, the more we disturb it irreversibly [9].

The Boolean states 0 and 1 are then represented by a fixed pair of reliably distinguishable states of the qubit: as an example we use horizontal and vertical polarizations:

$$|0\rangle = \leftrightarrow \quad and \quad |1\rangle = \updownarrow \tag{2.15}$$

while intermediate states, or *superposition*, correspond to other polarizations, and they can be depicted by:

$$\nearrow = \sqrt{1/2}(|0\rangle + |1\rangle) \quad and \quad \searrow = \sqrt{1/2}(|0\rangle - |1\rangle) \tag{2.16}$$

The main difference with the intermediate states of a classical bit is that these intermediate states cannot be reliably distinguished, even in principle, from the basis states. If we take a measurement which distinguishes the states $|0\rangle$ and $|1\rangle$, the superposition: $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ behaves like $|0\rangle$ with probability ${c_0}^2$ and like $|1\rangle$ with probability ${c_1}^2$. This means that *two quantum states are reliably distinguishable if and only if their vector representations are orthogonal*:

- $\leftrightarrow$ and $\updownarrow$ are reliably distinguishable by one type of measurement;
- $\nearrow$ and $\searrow$ are reliably distinguishable by another type of measurement;
- there's no way to reliably distinguish $\leftrightarrow$ from $\nearrow$.

The previous result is an interesting property of quantum mechanics, that will be suitable for the subsequent dissertations [8].

Von Neumann Entropy

We have previously introduced the concept of *Shannon Entropy*, and we have seen that it measures the uncertainty of a classical random variable; we would like to generalize this notion to quantum information, and that's why we introduce the *von Neumann Entropy*, which measures some form of uncertainty in a quantum state.

In quantum mechanics, physical quantities are associated with linear operators called **observables**: these are linear applications from the Hilbert space $\mathcal{H}$ to $\mathcal{H}$ itself. We can denote as $\mathbf{O}|\psi\rangle$ the application of an operator $\mathbf{O}$ to a state $|\psi\rangle$. Subsequently, we can define the **trace of an operator** (for some orthonormal basis $\{|b\rangle\}$) as:

$$\operatorname{Tr}\mathbf{O} = \sum_b \langle b|\mathbf{O}|b\rangle \tag{2.17}$$

Another useful representation is the **density matrix**, which is a positive linear operator that represents a quantum state. A classical random variable, yielding state $|\psi_i\rangle$ with probability p_i, gives the density matrix:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{2.18}$$

We can then give the definition of *pure quantum state*: ρ is pure whenever it can be written as a density operator (called *projector*): $\rho = |\psi\rangle\langle\psi|$. Otherwise, it is called *mixed* [4].

- Examples of *pure* states: $|0\rangle\langle 0|$, $|1\rangle\langle 1|$, $|\nearrow\rangle\langle\nearrow|$, $|\searrow\rangle\langle\searrow|$
- An example of *mixed* state:

$$\rho = \frac{1}{4}(|0\rangle\langle 0| + |1\rangle\langle 1| + |\nearrow\rangle\langle\nearrow| + |\searrow\rangle\langle\searrow|) \tag{2.19}$$

Exploiting this tools we can finally introduce the **von Neumann entropy** of ρ:

$$H(\rho) = -\operatorname{Tr}(\rho\log\rho) \tag{2.20}$$

The **von Neumann entropy** admits an intuitive interpretation. Suppose that Alice generates a random quantum state $|\psi_x\rangle$ according to some probability density $p_X(x)$ of a random variable X. Suppose further that Bob has not yet received the state from Alice and does not know which one she sent. The expected density operator from Bob's point of view is the ρ of formulation 2.20. We can then exploit this result to get an interpretation on $H(\rho)$: it quantifies Bob's uncertainty about the state sent by Alice, as his expected information gain is $H(\rho)$ qubits upon receiving and measuring the state that Alice sends [10].

To sum up, we have now all the essential elements required as foundations in order to fully understand the main concepts and principles of *Quantum Mechanics*, that will be analyzed in the following.

2.3.2 The Quantum Superposition principle

One of the main aspect that uniquely distinguishes quantum theory from any classical theories in physics is the **Quantum Superposition Principle**; it is one of the fundamental axiom of quantum mechanics and expresses the idea that a system can exist simultaneously in two or more mutually exclusive states.

Axiom 5 (The Quantum Superposition principle)**.**

A quantum state is a linear superposition of other quantum states, which means that a particle in one quantum state is also simultaneously in other quantum states.

The point is that quantum systems subsist in a superposition of mutually incompatible states, because they are expressions of entities that intrinsically 'lack' something until a measurement transforms them into an actual system [7].

2.3.3 The Observer Effect principle

This principle refers to all the modifications that the act of observation can make on an observed event.

Axiom 6 (The Observer Effect principle)**.**

Observing a phenomenon -or measuring it- will perturb the system.

If we adress the eavesdropper with the name *Eve*, in order to indentify a malicious entity with the role of the adversary in cryptology, we can exploit this apparently negative law to our advantage. In fact, thanks to the **Observer Effect principle**, Eve cannot get any information about the communication without introducing perturbations that would in principle reveal her presence.

Let's presume that Alice (the traditional sender side of the communication) codes information in individual photons, with the aim of sending them to Bob (the conventional receiver). If Bob receives the photons unperturbed, then, according to the Observer Effect's axiom, the photons were not measured; this implies that Eve did not get any information about the photons (acquiring information is synonymous with carrying out measurements).

After the photons-exchange, Alice and Bob check for the presence of Eve to establish if she was listenting: they just have to compare a randomly chosen subset of their data using a pubblic channel. If the received subset of Bob turns out to be unperturbed, then we establish that the adversary didn't get information, since the absence of perturbation implies no measurement, thus no eavesdropping.

2.3.4 The Heisenberg Uncertainty principle

The precision with which pairs of physical properties of a particle can be known simultaneously, has a fundamental limit which is depicted by the **Heisenberg Uncertainty principle**.

In fact, we can realize two different measures over a quantum system if and only if the two observables **A** and **B** (see *Section 2.3*) commute, that is: **AB** = **BA**. If they don't, then a simultaneous measurement implies uncertainty on both results. In order to quantify these equivocalities, we can consider the variances on both measures:

$$\sigma_m{}^2 = \langle \mathbf{A}^2 \rangle - (\langle \mathbf{A} \rangle)^2 \tag{2.21}$$

and

$$\sigma_n{}^2 = \langle \mathbf{B}^2 \rangle - (\langle \mathbf{B} \rangle)^2 \tag{2.22}$$

where $\langle \mathbf{X} \rangle$ and $\langle \mathbf{X}^2 \rangle$ are the mean value and the root mean square value of the measure, respectively.

These formulations give rise to an important result, known as the **Heisenberg Uncertainty principle**. It can be shown that [11]:

Axiom 7 (The Uncertainty principle)**.**

The simultaneous measurement over two non-commutative observable implies equivocality on both results. The variances follow this inequality:

$$\sigma_m \sigma_n \geq \frac{1}{2} \left| \langle \psi | [\mathbf{A} - \langle \mathbf{A} \rangle, \mathbf{B} - \langle \mathbf{B} \rangle] | \psi \rangle \right| \tag{2.23}$$

2.3.5 The No-cloning theorem

Suppose we have a quantum state $|s\rangle$ of a particular system which we would like to copy. This is the case of a malicious adversary whose primary intent is to get information about the communication without being detected, for example by making a copy of the message and sending the other one to the authorized receiver. But is it effectively possible? The answer was given in 1982 by Wootters and Zurek: quantum-mechanical states cannot be cloned.

Theorem 8 (The No-cloning theorem)**.**

It is impossible to make a perfect copy of an unknown quantum state.

This impossibility of cloning may seem at first an annoying restriction, but, if used with consciousness, it turns out to be a useful advantage, since we can exploit this property against the eavesdropper [12]. Eve could try to create a perfect copy of the quantum state sent by Alice with the aim of keeping the copy for herself and let the original move onward unperturbated; thanks to this principle, Eve cannot create a perfect copy of the state, thus she will face some gap of information.

2.4 Cryptographic Security Mechanisms

The term *Cryptography* is used to specify the techniques designed to ensure the confidentiality of information during the transmission, but it encompasses other fundamental functions such as authentication, signature and secret sharing. In this chapter, we will deal with *encryption*, *unconditionally secure authentication* and *key agreement*; all these topics will be analyzed in details.

2.4.1 Encryption

The aim of encryption is to ensure the **confidentiality** of the transmission, by rendering a message unintelligible to any unauthorized entity; this target can be achieved by using an algorithm (also called a cryptosystem or cipher) to combine a message with some additional information-known as the *encryption key*- and produce a cryptogram. In order for the cryptosystem to be secure in the sense of *Section 2.2.4*, it should be impossible to unlock the cryptogram without the key; anyway in practice, this requirement is mitigated so that the system is just extremely difficult to crack: the main goal is to protect the message at least as long as the informative content is valuable [6]. Essentially, there are two basic types of encryption schemes, depending on whether Alice and Bob use the same key:

- Asymmetrical (public-key) cryptosystems
- Symmetrical (secret-key) cryptosystems

Asymmetrical (public-key) cryptosystems

In an *Asymmetrical (public-key) cryptosystem*, different keys are used for encryption and decryption. This class of cryptosystems was first proposed in 1976 by Whitfield Diffie and Martin Hellman, while the first actual implementation was then developed by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978: it is known as *RSA* and is still widely used. The main principle is that, in order for Bob to be able to receive messages encrypted with a public-key cryptosystem, he must first choose a *private key*, and keeps it secret; after that, he computes a *public key* from this *private key*, and discloses it to any interested entity. Alice can then use this *public key* to encrypt her message: She transmits the encrypted message to Bob, who decrypts it with the *private key*.

The foundation of the security of public-key cryptosystems is the *computational complexity*: the idea is to use mathematical objects called one-way functions. This means that it is easy to compute the function $f(x)$ given the variable x, but difficult to reverse the calculation and deduce x from $f(x)$, where with "difficult" we mean that the time required to perform a task grows *superpolynomially* with the number of bits in the input, while "easy" stands for a performing time growing *polynomially* [6].

Strengths and Weaknesses of Asymmetrical cryptosystems

1. The first advantage of the **Asymmetrical** technique is that no key agreement is required in advance, since the only key that needs to be shared with the other entity is a *public key* that can be safely shared with everyone.

2. The second benefit is that the asymmetrical algorithm requires that only the member that has generated the key has the task to successfully keep it secret, while, as we will see in the next section, the security of a symmetric algorithm depends on two parties.

3. The third strength is that the issue of trusting the other party disappears in many scenarios, since without knowledge of your secret key, that entity cannot do malicious actions, such as digitally sign a document with your private key or divulge your secret key to others.

Nevertheless, the **Asymmetrical** technique has also some disadvantages, such as [13]:

1. all public-key cryptosystems rely for their security on unproven assumptions, which could themselves be weakened or suppressed by theoretical or practical advances. This increases the risks related to that technique.

2. An Asymmetric algorithm tends to be slower than a symmetric one.

3. They provide lower security than symmetric algorithms for a given key size.

Symmetrical (secret-key) cryptosystems

Symmetrical cryptosystems require the use of a *single key* for both encryption and decryption. Thanks to section 2.2.4 we are already familiar with *perfect secrecy*, and now we recall these concepts in order to introduce *Perfectly secret ciphers*, for which the knowledge of the ciphertext (encrypted data) does not change the statistical distribution of the plaintext messages (original data).

The most important and famous example of perfectly secret cipher is the **One-time pad**: in this scheme, Alice encrypts her message, a string of bits denoted by the binary number m_1, using a randomly generated key k, which in turn consists of a string of bits of length at least equal to the length of the plaintext message. Then, she simply adds modulo 2 each bit of the message to the corresponding bit of the key to obtain the scrambled text s:

$$s = m_1 \oplus k \tag{2.24}$$

The scrambled text is then sent to Bob, who decrypts the message by subtracting the key:

$$s \ominus k = m_1 \oplus k \ominus k = m_1 \tag{2.25}$$

The bits of the scrambled text do not contain any information, as they are as random as those of the key. This means that this cryptosystem is provably secure according to Information theory; actually, this is the only provably perfectly secure cryptosystem known today. Although perfectly secure, this scheme has a huge problem: Alice and Bob need to possess a common secret key, which must be at least as long as the message itself, and that key can only be used for a *single* encryption (hence the name ”One-time pad”). If they used the key more than once, Eve could record all of the scrambled messages and start to build up a picture of the plain texts and thus also of the key; furthermore, the key has to be transmitted with some trusted methods. Because of all these problems, the One-time pad is used only for critical applications [6].

Strengths and Weaknesses of Symmetrical cryptosystems

1. This type of encryption is simple and does not require a lot of computer resources.

2. Since a different secret key is used for communication with every different entity, if a key is compromised only the messages between a particular pair of sender and receiver are affected. The other communications are still secure.

3. Symmetric algorithms are much faster than asymmetric algorithms, especially for bulk data encryption.

4. Symmetric algorithms provide a higher level of security than asymmetric algorithms for a given key size.

For what concerns the limitations of this techniques, we have to keep in mind that:

1. Symmetric cryptography needs a secure channel for secret key exchange, in order to share a secret key before any secure communication can be initiated, and both parties must then ensure that the key remains secret.
2. In a large network using symmetric encryption there will be many key pairs, all of which must be securely managed. Because the secret key is exchanged and stored in more than one place, the symmetric key must be changed frequently [13].

We have pointed out advantages and disadvantages of both techniques, and for this reason it is important to notice that one type of cryptography does not replace the other: they should be used appropriately and in a complementary manner, exploiting the distinctive benefits depending on the situation.

2.4.2 Authentication

Authentication is a set of techniques with the aim of verifying that a transmitted message arrives unmodified at the receiver: this implies that if the message has been modified of created by another entity which is not the sender, it can be detected and discarded [14]. The main attacks against an authentication scheme are the following:

- *Impersonation*: after seeing n messages $m_1....m_n$, the malicious opponent creates a message m'_{n+1} that he wants to be accepted by the receiver as legitimate.
- *Substitution*: after seeing n messages $m_1....m_n$, the malicious opponent modifies the message m_n and replace it with his own message m'_n that he wants to be accepted by the receiver as legitimate [4].

So the main issue that authentication wants to solve is: how can Bob be sure that the message he has received was actually sent by Alice? In addition, how can he be sure that the message was not corrupted by Eve during the transmission?

As for encryption, authentication schemes are classified as being **Computationally secure** or **Unconditionally secure**. *Computationally secure* means that the security of the scheme relies on the computational difficulty to crack it; while *Unconditionally secure* denote a scheme whose security is indipendent of the computing power or time applied by an opponent attempting to encroach the system.

In general, the authentication depends on the message content: as an example, authentication is achieved by adding some form of redundancy -called *Message Authentication Code* (MAC)- to the original message. The secret key K shared by Alice and Bob is used to generate and verify the MAC: before sending the message, Alice attaches a MAC, calculated as a function of the key and the message itself. When Bob receives the message, he calculates his MAC and compares it with the attached MAC. If they match (and provided that the shared key is not compromised), Bob can be confident that the message is legitimate [4].

A good example of a *computationally secure* authentication scheme is a MAC built upon **block ciphers**, while a model of *unconditionally secure* MAC is the one calculated from **universal families of hash functions**.

Computationally secure MAC: built upon block cipher

Let m be the message to be authenticated, composed of β blocks (denoted with m_i), each of length b. Let F_K be a **block cipher** with key K: that is, a cryptographic primitive which encrypts a block of b bits at a time using a key of n bits. We start by encrypting the first block:

$$a_1 = F_K(m_1) \tag{2.26}$$

After that, the second block is bitwise modulo-2 added to a_1 before it is encrypted. That is:

$$a_2 = F_K(m_2 \oplus a_1) \tag{2.27}$$

For each of the next blocks, one computes:

$$a_i = F_K(m_i \oplus a_{i-1}) \tag{2.28}$$

Finally, the value of the last encryption a_β is used as a MAC.

The security of this scheme depends on the strength of the underlying block cipher, in the sense that if the block cipher is secure, this construction gives a strong MAC. Anyway, in the scope of Quantum Cryptography, the idea is to step away from computational assumption, and that's why we need to define *unconditionally secure MACs*.

Unconditionally secure MAC: built upon universal families of hash function

A well-studied technique for designing *Unconditionally secure MACs* is based on strongly universal families of hash functions, as proposed by Wegman and Carter [Universal classes of hash functions]. The basic idea behind universal hash-function families based MACs is to compress the message to be authenticated (using a universal hash function) and then encrypt the compressed image (e.g., using one-time pad ciphers, stream ciphers, or pseudorandom functions). In order to fully understand these concepts, is useful to introduce the core principle of this technique:

Universal families of hash function In general, a *hash function* is a function that maps a larger set to a smaller one, with the property that any two different inputs are likely to yield different outputs. In our technique we will use a *family* of hash function: the choice of the specific hash function that will be employed is kept secret by the legitimate parties. The security of this construction comes from the properties of the *family* itself, rather than from those of the *individual* hash functions, which can be very simple: that's why we're going to introduce a fundamental property for families, the **Strong Universality** [4].

Definition 9 ($\frac{\epsilon}{|\mathcal{B}|}$-almost strongly 2-universal)**.**

Given two sets $\mathcal{A}$ and $\mathcal{B}$, a class $\mathcal{H}$ of functions $\mathcal{A} \to \mathcal{B}$ is $\frac{\epsilon}{|\mathcal{B}|}$-almost strongly 2-universal if the following two conditions are satisfied:

- *for any $x_1 \in \mathcal{A}$ and any $y_1 \in \mathcal{B}$,we have:*

$$|\{h \in \mathcal{H} : h(x_1) = y_1\}| \leq \frac{|\mathcal{H}|}{|\mathcal{B}|} \tag{2.29}$$

- *for any $x_1 \neq x_2 \in \mathcal{A}$ and any $y_1, y_2 \in \mathcal{B}$, we have:*

$$|\{h \in \mathcal{H} : h(x_1) = y_1 \wedge h(x_2) = y_2\}| \leq \frac{\epsilon|\mathcal{H}|}{|\mathcal{B}|^2} \tag{2.30}$$

If the last condition in satisfied for $\epsilon = 1$, the class is simply called ***strongly 2-universal***

Where if $\mathcal{S}$ is a set, $|\mathcal{S}|$ will denote its *size*, that is, the number of elements in $\mathcal{S}$.

To sum up, the idea of a universal class of hash functions is to define a collection $\mathcal{H}$ of hash functions in such a way that a random choice of a function $h \in \mathcal{H}$ yields a low probability that any two distinct inputs x and y will collide when their hashed values are computed using the function h [15]. Then, given a $\frac{\epsilon}{|\mathcal{B}|}$-almost strongly 2-universal family, we can compute the MAC as:

$$MAC = h_K(m) \tag{2.31}$$

where m is the message to authenticate and K the shared secret key.

Regarding the main attacks against an authentication scheme we've disclosed before, the use of an $\frac{\epsilon}{|\mathcal{B}|}$-almost strongly 2-universal family of hash functions allows the MAC to have:

- *Impersonation probability* $= \frac{1}{|\mathcal{B}|}$
- *Substitution probability* $\leq \frac{\epsilon}{|\mathcal{B}|}$

No matter how powerful the opponent's computer is and no matter how cryptanalysis evolves over the year: Eve won't misleave the legitimate parties, except with an arbitrarily small probability.

2.4.3 Key Agreement

Key Agreement is the problem of generating a shared secret key K by two entities knowing dependent random variables X and Y, respectively. So *Key Agreement* solves the following problem: two entities i and j wish to agree on keying information in secret over a distributed network; each party desires an assurance that no party other than i and j can possibly compute the keying information agreed.

In general, the main desirable attributes for a *key agreement protocols* are the subsequents [16]:

1. *Resilience against known past keys*: the protocol still achieves its goal even if the opponent has learned some previous keys.

2. *Forward secrecy*: if long-term secrets of one or more entities are compromised, the secrecy of previous session keys is not affected.

3. *Loss of information*: Compromise of other information that would not ordinarily be available to an adversary does not affect the security of the protocol.

4. *Message indipendence*: If the protocol runs between two honest entities, individual flows are unrelated.

In the following, we will deal with specific method of exchanging cryptographic keys: **the Diffie Hellman key agreement**, which is one of the earliest practical examples of key exchange implemented within the field of cryptography and is *Computationally secure*, and the **Information Theoretic key agreement**, which on the contrary is guaranteed to be *Unconditionally secure.*

Diffie Hellman Key Agreement

Diffie-Hellman key agreement provided the first practical solution to the key sharing problem, allowing two parties, never having met in advance or sharing keying material, to establish a shared secret by exchanging messages over an open channel. The security rests on the intractability of the Diffie-Hellman problem and the related problem of computing discrete logarithms. We will analyze the protocol step by step:

- The first thing is to select and publish an appropriate *prime* p and a number α, called the *generator* of the multiplicative group of integers modulo p (where an element α is called a generator of a group $\mathcal{A}$ if every element in $\mathcal{A}$ can be expressed as the product of finitely many powers of α);

- After that, Alice picks a random number x and evaluates:

  $$\alpha^x \bmod p \tag{2.32}$$

  and sends it to Bob through the public channel.

- Bob chooses another random number y, then computes and sends to Alice the value:

  $$\alpha^y \bmod p \tag{2.33}$$

- Bob receives α^x and computes the shared key as:

  $$(\alpha^x)^y \bmod p \tag{2.34}$$

- Alice receives α^y and computes the shared key as:

$$(\alpha^y)^x \bmod p \tag{2.35}$$

we see that the resulting secret key are the same, since:

$$(\alpha^y)^x \bmod p = (\alpha^x)^y \bmod p = k \tag{2.36}$$

- If Eve wants to compute k, then she would need either x or y. Otherwise, Eve would need to solve a Discrete Logarithm Problem, and there is no known algorithm able to accomplish this with a polynomial complexity in the number of digits of p.

Information Theoretic Key Agreement

In 1993, Maurer introduced a general *Information theoretic model* for key agreement [17]. It takes place in a scenario where Alice and Bob are connected by an insecure channel to which a passive eavesdropper Eve has perfect access; Alice and Bob, who want to establish a mutual secret key have access to realizations of random variables X and Y, respectively, whereas the adversary Eve knows a random variable Z: these variables are distributed according to some joint probability distribution P_{XYZ}. The context is the following one [18,19]:

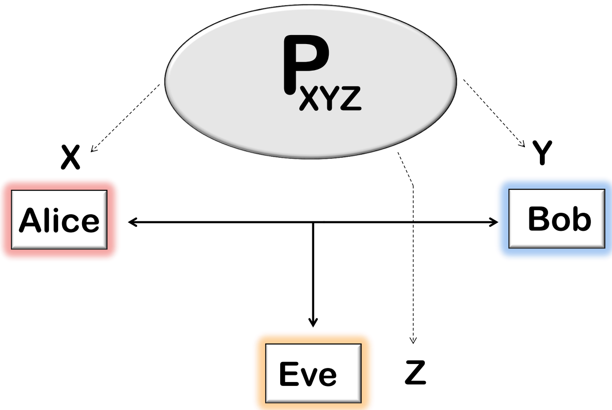


Figure 2.1: Secret key agreement by public discussion from common information

A surprising fact that was demonstrated by Maurer was that, even if Eve's channel is much superior compared to that of Alice and Bob, *unconditionally secure secret key agreement is possible*, provided Alice and Bob know P_{XYZ}. In the following, we will consider the case of a source that sends out random bits at very low signal power and Alice, Bob and Eve receive these bits over indipendent binary-symmetric channels with error probabilities α, β and ϵ respectively. A key agreement protocol for such a scenario generally consists of three phases:

1. *Advantage distillation*
2. *Information reconciliation*
3. *Privacy ampification*

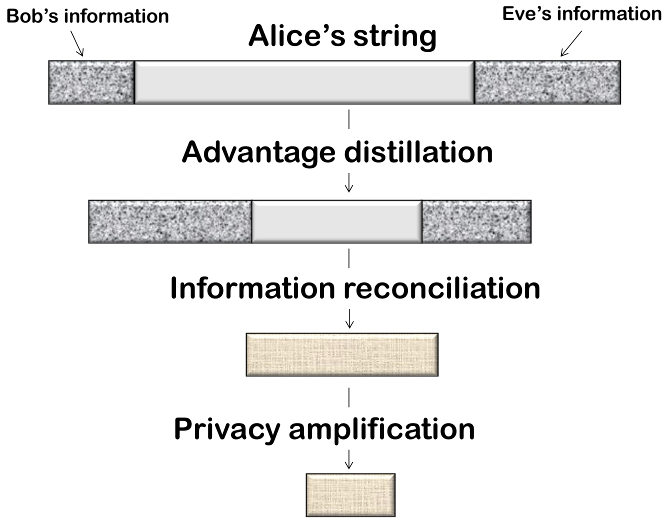


Figure 2.2: Phases of a Secret-Key-Agreement Protocol

Advantage distillation: this phase is needed when neither Alice nor Bob has an advantage compared to Eve with respect to the information about each other's random variables. The objective of this phase is hence to generate an advantage over the opponent by exploiting the authenticity of the public channel. After this phase, involving a sequence of messages summarized in a random variable C, Alice can compute a string W from X and C about which Bob has less uncertainty than Eve: $H(W|YC) < H(W|ZC)$. Essentially, the basic idea of the advantage distillation phase is that Alice and Bob use the noiseless discussion channel for exchanging information about their bits in an insicure but authentic way, with the objective of identifying bits that are correct with a higher probability than others.

Information reconciliation: this second phase consists of interactive error correction; after advantage distillation Bob has more information about Alice's string than Eve, and after information reconciliation, Bob should exactly know Alice's string. This means that Alice and Bob exchange redundant information and apply error-correction techniques in order for Bob to be able to learn W with very high probability but such that Eve is left only with incomplete information about it.

Privacy ampification: in this last step, Alice and Bob distill from W a shorter string K about which Eve has only a negligible amount of information. For istance, Alice and Bob publicly agree on a function g that becomes known to Eve, and they let $K = g(W)$. This is related with the concept of *hash function* introduced in *Section 2.4.3*, and will be analyzed in details in Chapter 3 in the *Privacy Amplification* section [18,19].

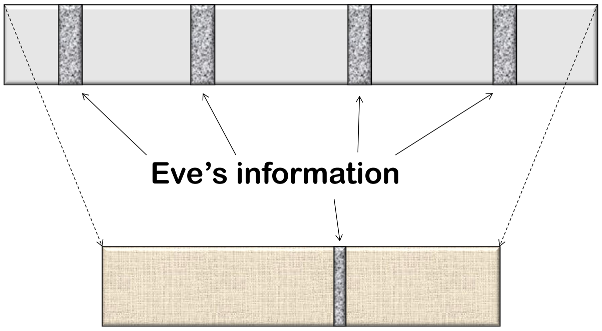


Figure 2.3: Eliminating Eve's knowledge by Privacy Amplification

3 | Quantum Key Distribution

In the previous chapter we have briefly reviewed the foundations of *Quantum Information Theory*, with the aim to present the main laws and principles that can be used to ensure the confidentiality of information, by exploiting the fascinating behaviour of photons; now we are ready to introduce the security mechanism that is dealt with in this dissertation: *Quantum Key Distribution* (*QKD*).

QKD uses the main principles of quantum mechanics in order to guarantee a secure communication between two parties, conventionally called Alice and Bob. To ensure the confidentiality of communications, they agree on a common yet secret key, that is, a piece of information with whom they can perform an *encryption* of the message, so that the result is incomprehensible to an observer who doesn't know the key. Alice anb Bob wish to exchange a key having access to a *Quantum Communication Channel*, on which quantum carriers -i.e. photons- are transmitted, and to a *Public Classical Authenticated Channel*, on which the two parties will exchange side information in order to detect the presence of an eavesdropper Eve. In the following, we will refer to the scheme depicted in *Figure 3.1*:

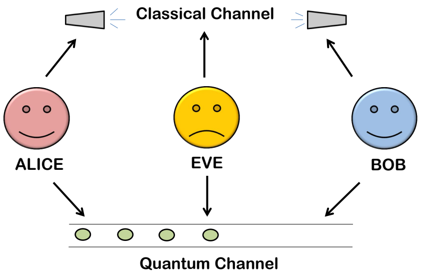


Figure 3.1: QKD model

Alice encodes random pieces of information -i.e. random bits- in the quantum carriers, in order to build the key; she then sends them to Bob through the *quantum channel.* Eve might eavesdrop this transmission and therefore spy on potential secret key bits; anyway, this does not pose a fundamental problem, since the eavesdropping is detectable thanks to the arise of transmission errors. Furthermore, as we can deduce from the presence of a *Public Classical Authenticated Channel*, Quantum Key Distribution is not only based on the main principles of quantum mechanics, but it also relies on *Classical Information Theory*: in order to obtain a distributed key which is common and secret, the transmission errors detected previously must be corrected, and after that, is fundamental to ensure that Eve knows nothing about the key. These two targets are achieved by exploiting techniques from classical information theory, denoted as *secret-key distillation* techniques.

One of the remarkable aspect of Quantum Key Distribution is that Shannon's condition on the secret key length needed to achieve perfect secrecy is no longer a problem, since we can use *QKD* to obtain a long secret key, and then use it classically to encrypt a message of the same length. Furthermore, the security of *QKD* is guaranteed by the laws of quantum mechanics, enabling the long-term secrecy of confidential data transmission [4].

Quantum Key Distribution is such a prosperous field that it gave rise to several different approaches, that can be divided into two main categories, depending on the property they exploit:

- *Entanglement based protocols*: these protocols are based on *entanglement*, which is a quantum phenomenon that implies that a measurement performed on one physical system affects the other, even if they are spatially separated. This means that the quantum states of two or more distinct objects are correlated, and they can be described as a *combined quantum state* rather than as individual entities. This principle can be exploited in the subsequent way: if Alice and Bob share an entangled pair of objects, the overall system will be altered if an eavesdropper intercepts either object, and this will reveal the presence of Eve.

- *Prepare and measure protocols*: we already know from the Observer Effect principle (*Section 2.3.3*), that measuring an unknown quantum state will change that state in some way. We can exploit this peculiarity with the aim of detecting an eavesdropper and calculating the amount of information intercepted.

In the following, we will deal with the QKD schemes of *Prepare and measure* category.

With the purpose of efficiently introducing this disquisition, we will introduce the subsequent concepts by simultaneously proposing an overview of the first *QKD* protocol, created in 1984 by Bennett and Brassard and called **BB84**. This protocol is an exquisite model that can be exploited in order to focus on the main aspects of the *QKD* technique and understand it as broadly as possible.

3.1 Key Exchange

The main purpose of *QKD* is to distribute a secret key between Alice and Bob, who share no information initially, and protect it from an adversary called Eve, who eavesdrops on their communications. We have broadly enlightened that the fundamental component of this process is the qubit, an elementary quantum system -i.e. a photon- used to encode digital information, and that, following the *Uncertainty principle* (see *Section 2.3.4*), there exist pairs of properties such that measuring one property necessarily randomizes the value of the other. An example of this phenomenon is measuring a single photon's linear polarization: this, for example, randomizes its circular polarization, and vice versa; in general, any pair of polarization states will be referred to as a **basis** if they corrispond to a reliably measurable property of a single photon [20].

But what are the main tools in order to deal with polarization? We know from [21] that we can produce polarized light by sending an ordinary light beam through a polarizing apparatus -such as a Polaroid filter or calcite crystal-; the beam's polarization axis will be determined by the orientation of the polarizing apparatus in which the beam originates. This allows Alice to choose a random bit string and a random sequence of polarization bases; each base is made by two orthogonal quantum states that are reliably distinguishable (see *Section 2.3.1*), as for example:

- **the rectilinear base**, depicted as ✛:
  - ↔ represents random bit 0
  - ↕ represents random bit 1
- **the diagonal base**, depicted as ✕:
  - ↗ represents random bit 0
  - ↘ represents random bit 1

Alice selects a base for each random bit, represents this bit in the chosen base, and sends Bob a train of photons. For each received photon, Bob decides randomly whether to measure the photon's rectilinear polarization or its diagonal polarization, and interprets the result of the measurement as a binary 0 or 1. This process goes as follows:

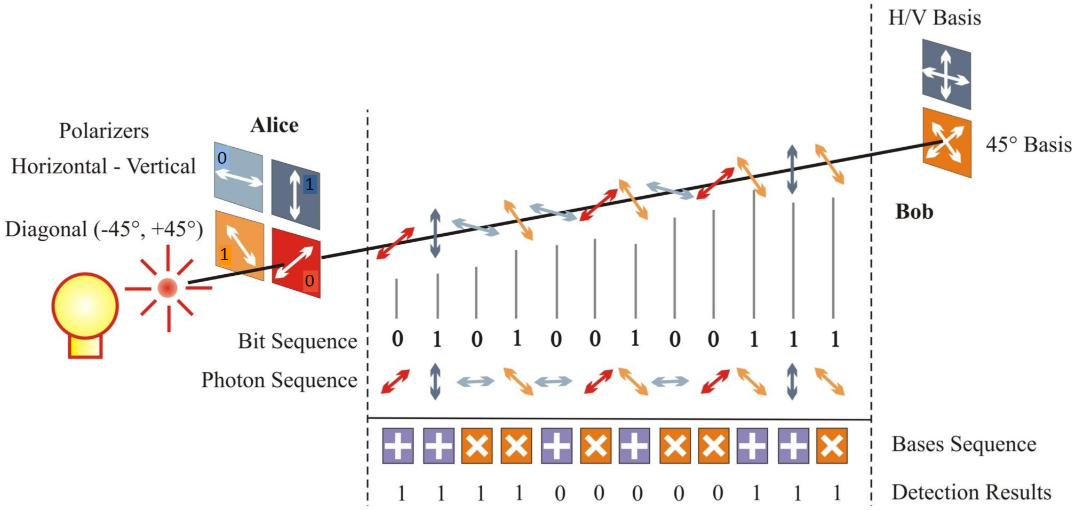


Figure 3.2: Key Exchange scheme [22]

At the end of this procedure, the *raw bit list of Alice* corresponds to the list of bit values she has sent through the quantum channel, while the *raw bit list of Bob* is composed of the list of bit values he has measured. These two lists can be different, as qubits do not always reach the receiver due to loss in the quantum channel. In order to extract a **raw key** from these raw bit lists, Bob announces to Alice -by using classical communication- the index of the qubits for which he had a detection, so Alice can make the related selection in her list. After that, they both have generated a **raw key**.

3.2 Key Distillation

Key Distillation is the technique used to convert some random information, shared among the legitimate parties and a potential adversary Eve, into a **secret key**. The protocol is made of three steps:

1. *Sifting*

2. *Reconciliation*
3. *Privacy Amplification*

We have closed the previous section with the creation of a *raw key*; in the following we will fully analyze how to exploit this result in order to obtain a **secret key** shared by Alice and Bob, and unknown to Eve. The following figure schematically shows the result of each procedure:

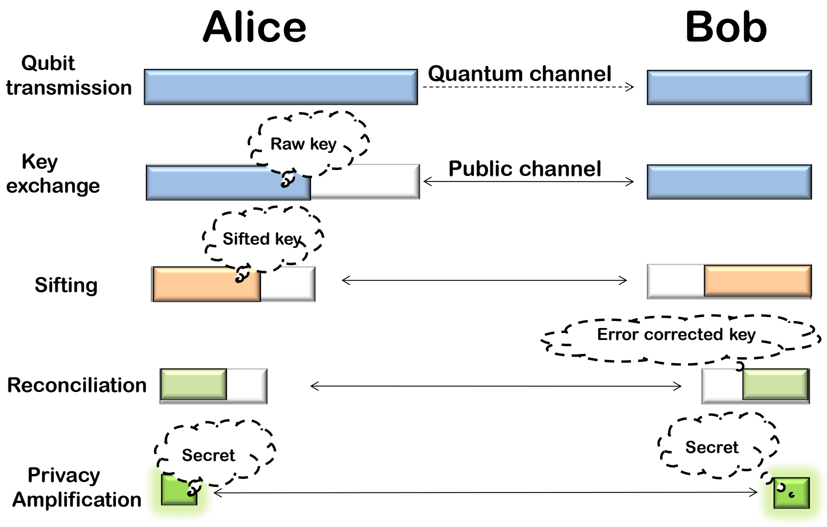


Figure 3.3: The Quantum Key Distribution scheme

3.2.1 Sifting

After raw key exchange, the key is *sifted*, that is, Bob reveals over the classical channel some information on the sequence of detections he gets: he reports bases of received bits, without disclosing the actual results of his measurements, while Alice says which bases were correct. During this phase the bits that do not have a perfect correlation between the bits of the emitter and those of the receiver are discarded. The effect of *Sifting* is that sender and receiver share a correlated key, called **sifted key**, with the same length; furthermore, what is revealed during the sifting does not allow an eavesdropper to get any information on the key. Recalling *Figure 3.2*, we can enlight the results of sifting:

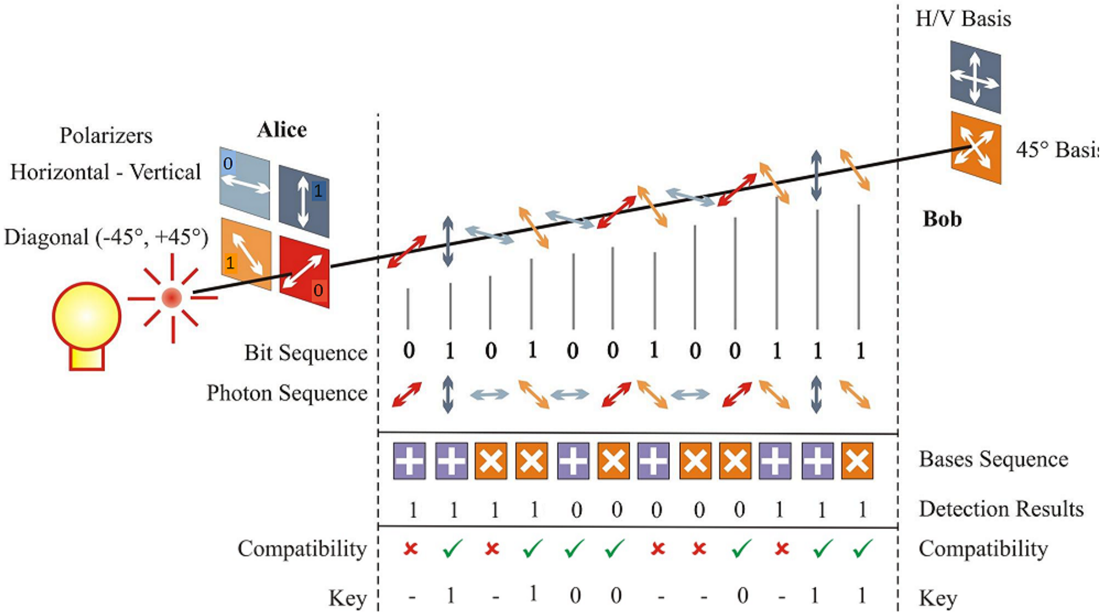


Figure 3.4: Sifting scheme [22]

Effect of eavesdropping

Thanks to the random mix of rectilinear and diagonal photons in the quantum transmission, it is possible to detect the presence of a malicious entity: an eavesdropper intercepting the photons will carry the risk of use the wrong basis, thus modifying the state of these photons and producing disagreement between Bob and Alice on some of the bits on which they expected to agree. As a consequence, Alice and Bob simply have to check the presence of errors in the sequence of shared bits, by comparing over the classical channel a sample of the bits, to verify the integrity of the key; this of course will sacrifice the secrecy of the disclosed bits, that will be discarded, as they may have been intercepted by Eve.

The following step is to remove errors and information that the adversary could have collected, by processing the *sifted key*. This can be done by means of *Reconciliation* and *Privacy Amplification* processes: the resulting key is secure and can be used as a **secret key** material [21].

3.2.2 Reconciliation

Reconciliation is a type of error correction carried out between Alice and Bob's key, with the aim of ensure both keys are identical; this procedure is done over the Public Classical Authenticated Channel, thus it is important to minimize the disclosed information, as this can be monitored by the eavesdropper. In the case of *BB84*, *Reconciliation* takes the form of an *interactive* error correction protocol, that is, Alice and Bob exchange information both ways: they alternatively disclose parities of subsets of their key elements. If they detect a diverging parity, it means that there is at least one error in the corresponding subset: by dividing the block and by means of a check for each sub-block, they can locate the error and correct it. The two parties repeat the process a sufficient number of times and the result is that they both share equal bits.

BBBSS: The first *binary interactive error correction* protocol used for QKD is called **BBBSS** and was designed by Bennett, Brassard et al. 1991; this can be used to find and correct errors introduced into the quantum channel from noise and from eavesdroppers. Alice and Bob need to exchange parities of subsets of their bits: by detecting the presence of diverging paritites, the transmitter and receiver can focus on errors and correct them by mean of a bisection. This protocol uses several iterations, between which the bit positions are permuted in a pseudo-random way [4]. A schematic overview of the main criteria of **BBBSS** is given in the following:

1. **In each iteration**, Alice and Bob divide the shared binary string into sub-blocks with the same length. That is:

$$l = nw \tag{3.1}$$

   with l = length of the overall string, n = number of subblocks and w = number of bits for each subblock.

2. **First iteration:** in this first phase, the subblocks are the sets of indices:

$$B_j^{(i)} = \{(j-1)w + 1.......jw\} \qquad for \quad 1 \leq j \leq n \tag{3.2}$$

3. **Subsequent iterations:** the two parties agree on a randomly-chosen permutation $\pi^{(i)}$. The purpose of the permutation is to attempt to spread out the error bits randomly and separate consecutive errors from each other. The subblocks contain the indices:

$$B_j^{(i)} = \{\pi_t{}^{(i)} : (j-1)w + 1 \leq t \leq jw\} \tag{3.3}$$

4. Alice and Bob then share the parities of these subblocks by using the classical channel to compare them; for iteration i, they disclose the parities:

$$\mathcal{P}_{X,j}^{(i)} = \sum_{t \in B_j^{(i)}} X_t \tag{3.4}$$

   and:

$$\mathcal{P}_{Y,j}^{(i)} = \sum_{t \in B_j^{(i)}} Y_t \tag{3.5}$$

   with X and Y = ℓ-bit strings of Alice and Bob respectively.

5. When a subblock has mismatching parities, it means that there is an *odd* number of error in that subblock, since an *even* number of errors would mask each other; that is, at least one. If this is the case, a *bisection* is used: Alice and Bob exchange the parity of *half* of the subblock, and they check if the parity is right or wrong.

   - **if wrong**: the two parties have to go on with the bisection in that half;
   - **if right**: this means that at least one error exists in the other half of the subblock; thus the bisection takes place in the other half.

   The bisection ends when an erroneous bit is disclosed; Bob knows the position of this bit and can correct it by simply flip it. For what concern secrecy, it is assumed that the parity bits give information to the eavesdropper about the secret key, since the exchange of parity bits occurs on the classical channel that can be monitored by Eve. In order to prevent and circumscribe the information gained by the adversary during this procedure, the last bit of each block and subblock involved in the parity check is discarded [23].

The Cascade Protocol : Another common protocol used for information reconciliation is the **Cascade Protocol**, proposed in 1994; this is an improvement of *BBBSS* in terms of bits leaked during the reconciliation stages and it will be discussed in the following.

1. **First iteration:** is identical to the first iteration of *BBBSS*.

2. **Second iteration:** starting from the second iteration, and unlike *BBBSS*, Cascade keeps track of all investigated subblocks, by taking advantage of this information. The protocol keeps two sets of subblocks, and each subblock for which a parity was disclosed is listed in one of these:

   - $\mathcal{B}_0$: subblocks for which the parity is equal between Alice and Bob;
   - $\mathcal{B}_1$: subblocks for which the parity is diverging.

3. When the bisection ends and an error is corrected -i.e. the concerned bit b is flipped- Alice and Bob go through the list of all the subblocks for which they have already calculated the parity in the current and previous interations. For a subblock containing the flipped bit b, the parity is flipped as a consequence, due to the correction of that bit. This means that there may be subblocks for which the parity was equal between the two parties, and which is now different. Let $\mathcal{S}_i \subseteq \mathcal{B}_i$, with $i = 0, 1$, be the set of subblocks in $\mathcal{B}_i$ such that they contain the bit b; the sets $\mathcal{B}_0$ and $\mathcal{B}_1$ are updated by considering the complement in the following way:

$$\mathcal{B}_0 \leftarrow \mathcal{B}_0 \backslash \mathcal{S}_0 \cup \mathcal{S}_1 = \{x \in \mathcal{B}_0 | x \notin \mathcal{S}_0 \cup \mathcal{S}_1\} \tag{3.6}$$

   and:

$$\mathcal{B}_1 \leftarrow \mathcal{B}_1 \backslash \mathcal{S}_1 \cup \mathcal{S}_0 = \{x \in \mathcal{B}_1 | x \notin \mathcal{S}_1 \cup \mathcal{S}_0\} \tag{3.7}$$

4. Before the end of an iteration, Alice and Bob correct all the known diverging parities; among all the subblocks in $\mathcal{B}_1$, they proceed with a bisection in the smallest of such subblocks, and when they find an error to correct they update again the parities of all the previous subblocks, update $\mathcal{B}_0$ and $\mathcal{B}_1$, repeating the process until $\mathcal{B}_1 = \oslash$, that is until no subblocks presents a diverging parity [4].

3.2.3 Privacy Amplification

Privacy Amplification is a fundamental tool implemented in order to extinguish Eve's information at the cost of a reduced key length. The aim is to exploit what the eavesdropper does not know about the key: so, if r is the string after reconciliation,

n is its length, and s is an arbitrary security parameter, we can deduce, thanks to [24], that if Eve's knowledge about r is $\leq l$ deterministic bits, then a **hash function h** randomly and publicly chosen from an appropriate class of functions:

$$\{0,1\}^n \rightarrow \{0,1\}^{n-l-s} \tag{3.8}$$

will map r into a value $h(r)$ about which Eve's expected information -I_{Eve}- is upper bounded by:

$$I_{Eve} \leq \frac{2^{-s}}{ln2}\ bit \tag{3.9}$$

this means that by choosing a suitable arbitrary security parameter s, Eve knows nothing at all about the final secret key $h(r)$ shared between Alice and Bob except with probability at most $\frac{2^{-s}}{ln2}$, in which case she knows at least one deterministic bit [20].

The concept of *Universal families of hash function* was fully introduced in Section 2.4.2; in the following we will deal with some requirements for these families in the scope of **privacy amplification**:

- **the family should be Universal**: this is because the proximity to universality directly affects the quality in terms of secrecy of the resulting key.
- **The number of bits used to represent a particular hash function within its family should be resonably low**: in fact, the choice of the specific hash function has to be transmitted between Alice and Bob, so it is important to subdue that transmission as much as possible.
- **The evaluation of a hash function within the family should be efficient**: in a real-time application of *QKD*, the secret key distillation should not take too much time, and since the evaluation of the hash function is one of the critical issues, this has to be handle with care.

Thanks to the processes described in the previous sections, the *Key Distribution problem* is solved by allowing the exchange of a cryptographic key between two remote parties with absolute security. In the following, we introduce the main concepts of another *QKD protocol*, i.e. the **B92 protocol**, as it was used in the experiment at the basis of this thesis.

The B92 Protocol In 1992, a simplified version of the *BB84 protocol* was proposed by Bennett in his paper: "*Quantum cryptography using any two non-orthogonal states*" [25]. The main difference between the *BB84* and the *B92 protocol* is that only two states are necessary in the latter, rather than the possible four polarization states of the former. The protocol exploits the following polarization directions: 0 can be encoded as 0 degrees in the *rectilinear* basis and 1 can be encoded by 45 degrees in the *diagonal* basis. This situation is depicted in *Figure 3.5*:

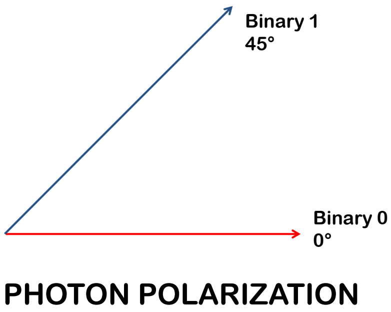


Figure 3.5: B92 2-state encoding

In the **B92 protocol** we deal with two *distinct nonorthogonal states*: $|u_0\rangle$ and $|u_1\rangle$. Alice prepares and transmits to Bob a sequence of photons, using states $|u_0\rangle$ and $|u_1\rangle$ to represent the bits 0 and 1 respectively; Bob still randomly chooses a basis for the measurement but if he chooses the wrong basis, he will not measure anything. After that, Bob publicly tell Alice in which case his measurement had a positive result, without unveil which measurement he made; in this way the two parties are able to agree on the discarding of all the other instances. The main difference between *BB84* and *B92* protocols is that, even if the former exhibits better performance in the distribution of the secure communication key over long distance, *B92* is easier and can be implemented with less resources [26].

USD attack Although the two non-orthogonal photon-polarization states of the B92 protocol cannot in general be distinguished unambiguously by an eavesdropping probe without being disturbed, they can be unambiguously discriminated at least some of the time. Thus a possible eavesdropping strategy against this protocol is the *unambiguous state discrimination attack* (USD attack): it allows Eve to identify a fraction of the signals without error. Unambiguous state discrimination is possible whenever the N states in question are linearly independent (for our case $N = 2$).

More generally, the problem can be described by a measurement which can give the results *state* 0, *state* 1, .. *state* $N-1$, and the result *don't know*. Eve can perform an unambiguous state discrimination (USD) measurement on the signal states, thereby distinguishing between cases where she got a deterministic result or a random one. After that, she applies the following strategy: in those cases where she knows the state with certainty, she forwards it to Bob; in all other cases, she sends no signal at all, mimicking a lossy channel. B92 protocol can therefore be affected by such an attack: furthermore, under a given threshold no secure key transmission is possible. This threshold, which depends on the state non-orthogonality, is defined as the transmissivity where the probability of success of an USD measurement equals Bob's detection probability on the lossy channel [27, 28].

In the Canary Islands experiment we used the B92 protocol; in the hypothesis of infinite length the asymptotic bound on the *secret key rate* will be:

$$R_k = R_{raw}[A\eta(1-2h(\epsilon))] \tag{3.10}$$

wherr R_{raw} is the raw key rate, A is the channel attenuation, related to the concept of turbulence, η is the protocol efficiency, h stands for the binary entropy and ϵ represents the *QBER* value. When ϵ is greater than the 11%, then the secret key rate drops to zero. Our exploiting-turbulence proposal selects only the transmissions with low *QBER*; the aim is to guarantee that the *QBER* decreasing overcomes the sifted bits loss, in order to obtain an overall enhancement of the secret key rate.

3.2.4 Conclusion

In this chapter, we have analyzed the main aspects of **Quantum Key Distribution**, highlighting the fundamental techniques of the process, in order to understand how the creation of a secret key is possible exploiting the laws of quantum physics. At this point we have all the theoretical basis to introduce the innovative approach developed in the Canary Island Experiment.

4 | The Canary Island Experiment: an innovative approach

4.1 Motivation and application scenarios

In the previous chapters we have seen how the domain of Quantum Communications could be useful and prosperous: we have studied its basic principles, evaluated the fundamental implementative features, and underlined that the introduction of a *Quantum approach* enables us to develop high-performance application fields that have been unexplored until recently. Effectively, the technological evolution is strongly encouraging to allow more robust communications, even when remarkable distances are introduced between the devices involved. The growing interest for the free-space propagation and the evaluation of new methods allowing long space transmissions have incentivized the need to apply Quantum Communication to great challenges related to the connection between remote devices: the aim is to explore new satellite employments and to accomplish ambitious scientific projects by pressing the application of Quantum Communications toward unexplored horizons.

The main obstacle related to free-space propagation is the presence of **Atmospherical turbulence**, which in fact introduces thermal fluctuations - thence reflaction index variations - producing perturbations in the wavefront of the object we are inspecting. As a consequence the light, instead of focusing in a definite image, is distributed in a larger and irregular surface in which the desired informations and details are substantially lost. This is an extremely limiting problem for free-space communications, and it implies critical issues which are really difficult to solve in modern systems. The incentive connected to the problem of turbulence is thus to determine an innovative technique allowing to exceed this dilemma in an elegant and ingenious way.

4.2 The intuition behind the experiment

So far, the approach that was used to stem the atmospherical turbulence problem required to correct and remove its detrimental effects, attempting to leverage the long distance communication systems. To this purpose, in the last decates, a rose of investigative discipline has been developed beside the classical astrophysics, attempting to study the nature of optical turbulence and how it acts on wavefronts: the intent is to measure the perturbations and correct them. These techniques are called *Adaptive Optics* and their aim is made of two steps: in the first one the wavefront perturbations are evaluated grounding on dedicated sensors, while in the second one an equally but opposite deviation is imposed with the purpose of rebuild the original configuration. Nevertheless, these methods have a lot of limitations: first at all, they can only *compete* -even if with accurate techniques- against a troublesome entity as the turbulence, that can easily annihilate the effectiveness of transmission.

Therefore, following the more brilliant principle for which: *the best way to conquer an enemy is to become good friends*, a new approach gets ahead thanks to the intuition of Capraro, Tomaello, Dall'Arche, Gerlin, Ursin, Vallone and Villoresi in the paper "Impact of Turbulence in Long Range Quantum and Classical Communications" [29]: in this work an hardy proposal is presented, that is, the turbulence is no more struggled or limited, but rather it is *exploited*. The amazing result is that potentially it is possible to complete successfully a key exchange even in loath conditions, in which by convention it would not be possible to complete a secure communication successfully.

The ingenious idea behind this new approach is connected to the fact that, when we have a turbulent channel, the statistics is transformed from *Poissonian* to **Log-normal**. The latter presents intensity peaks that are much more prominent and noticeable than the former. The intuition is thus to exploit this transmissivity peaks for cryptography: a channel with constant transmissivity presents a poissonian statistics; this is not the case of a free-space channel, in which the transmissivity is variable, and for this reason recurrently we have high peaks of intensity, because the presence of turbulence contribute to create a sort of *channel-effectiveness concentration phenomena*, that can be exploited in order to obtain secure key even in noisy conditions. In the following we will analyze the detail and theoretical fundamental of this approach, and we will expose the experiment built up with the purpose of evaluating in practical term if and when the application of this innovative approach allows the creation of secure key even in situation considered unusable until now.

4.3 Theoretical model and realization

When the channel experiences a strong turbulence, it undergoes an increase of trasmitted photons losses and hence an increase of the malicious consequences of noise: with the purpose of evaluating the quality of communications and the possibility to approach unlocked transmission distances, it is required to understand the effects of turbulence on propagation, and the statistics of the received information.

We already know from theory that the probability $p(n)$ of counting n photons in a time interval $[t_0, t_0 + T]$ -where T is the counting interval- follows the poissonian expression [30]:

$$p(n) = \frac{q^n}{n!} e^{-q} \tag{4.1}$$

In fact, we can experimentally see that the photon counting distribution at the laser transmitter is found to be accurately described by the poisson distribution.

Nevertheless, at the receiver the situation changes radically: the presence of **turbulence** induces fluctuations on q, the mean photon number in a counting interval at the receiver. Furthermore, q will decrease due to Rayleigh scattering; because of all this effects, we disclose the formula describing the probability distribution for q:

$$P(p) = \frac{1}{q\sqrt{2\pi\sigma^2}} e^{-\frac{\left[ln\left(\frac{q}{\langle q\rangle}\right)+\frac{1}{2}\sigma^2\right]^2}{2\sigma^2}} \tag{4.2}$$

where $\langle q \rangle$ is the average, and $\sigma^2 = ln(1 + SI) = ln\left(1 + \frac{\Delta q^2}{\langle q\rangle^2}\right)$, SI being the scintillation index.

We can obtain this formula by partitioning the length-L propagation path into a large number of sections N, each of which is of length Δz, and assuming that the effect of consecutive sections on the intensity is multiplicative, so we can write:

$$I = I_0 \prod_{j=1}^{N} e^{-\alpha \Delta z} T_j = a I_0 \prod_{j=1}^{N} T_j \tag{4.3}$$

where:

- I_0 is the intensity at $z = 0$
- I is the intensity at $z = L$
- $a = e^{-\alpha L}$ is the transmission coefficient associated with scattering and absorption.

- T_j Characterizes the turbulence effect on intensity in the $j-th$ section.

Similarly, we can rewrite:

$$\ln\left[\frac{I}{aI_0}\right] = \sum_{j=1}^{N} \ln(T_j) \tag{4.4}$$

in such a way that, reversing the previous formula, we get:

$$I = aI_0 e^{\chi} \tag{4.5}$$

where χ is the value $\sum_{j=1}^{N} \ln(T_j)$, i.e. a quantity containing the intensity fluctuations; from the assumed statistical indipendence of T_j and from the central limit theorem, it follows that the probability distribution of χ is **Gaussian**:

$$P(\chi) = \frac{1}{\sqrt{2\pi{\sigma_\chi}^2}} e^{-\frac{[\chi-\langle\chi\rangle]^2}{2{\sigma_\chi}^2}} \tag{4.6}$$

As a consequence, recalling the relation (4.5) between I and χ, we obtain:

$$P(I) = \frac{1}{I\sqrt{2\pi{\sigma_\chi}^2}} e^{-\frac{\left[\ln\left(\frac{I}{\langle I\rangle}\right)+\frac{1}{2}{\sigma_\chi}^2\right]^2}{2{\sigma_\chi}^2}} \tag{4.7}$$

From this result we can deduce the evolution on $P(q)$, which in fact follows a **log-normal** distribution.

The main conclusion is therefore that the initial equation (4.1) represents the probability of counting n photons in the interval T if q -that is, the mean photon number in a counting interval- is fixed. Nevertheless, in realistic situations q fluctuates due to the presence of atmospheric turbulence, and as a consequence the probability of detecting n photons in each interval does not follows the poissonian statistics, but instead it matches the Mandel distribution:

$$p(n) = \int_0^{\infty} dq \frac{q^n}{n!} e^{-q} P(q) \tag{4.8}$$

The previous argument and results, introduced in [30] have been strongly supported by the experimental foundings formalized in [29]. This experiment, realized in September 2011, is the fundamental basis for the subsequent investigation of June 2012, which data have been post-processed and analyzed in this thesis, in order to get an experimental proof on the feasibility of the *exploiting-turbulence* approach. Both experiments have been carried out in the Canary archipelagos, by studing the propagation on a free-space optical link between La Palma island (the transmitter were positioned at the Jacobus Kapteyn Telescope) and Tenerife, which lodged the receiver at the Optical Ground Station (OGS). The distance covered by the optical link was 143 km.

The data acquired at the receiver in the 2011 experiment have been narrowly worked out, with the aim of evaluate the scintillation of the beam and analyze the statistics of arrival of the incoming photons: an illustration of the results is shown in the following, where the counting occurrences and the correspondent log-normal distribution are plotted [29]:

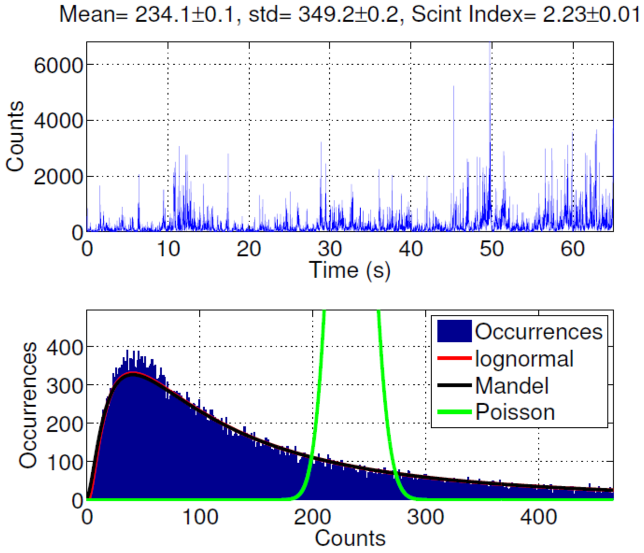


Figure 4.1: Temporal distribution of count occurrences and corresponding *lognormal* and Mandel curves. The green line corresponds to the Poissonian distribution that would be obtained without turbulence [29].

By comparing the log-normal distribution with the poissonian, which should have been obtained if there was no turbulence, we stress what we have previously argued: in a free-space channel the presence of turbulence implies a transformation of the statistics from *Poissonian* to **Log-normal**. This means, as can be seen from *Figure* 4.1, that the channel unveils a huge number of intensity peaks, that could be suitably exploited in order to obtain secure key even in critical situations, that generally would inhibit the correct realization of this process. This is the case of strong turbulence and high noise status, when we want to take advantage of considering only the particular moments in which the turbulence increase the transmettivity. This should be done in order to exploit turbulence to improve the *Signal to Noise Ratio*: following the principle proposed in [29], it is possible to enhance the *SNR* of a long distance free-space transmission by probing the channel using a classical signal (that in the following we will call **Probe signal**). This will give some information about the instantaneous transmission of the channel: if and only if this signal is above a given threshold, the photon signal is acquired. So it is essential to find a suitable **Thresold** to be used in order to realize this implementation. Moreover, it was shown in [29] that, if we take under consideration only the events in which the transmission is above a given threshold, that is $T > T_0$, then the mean number of detected photons becomes:

$$\frac{\langle n \rangle_{thres}}{\langle n \rangle} = \frac{1 - erf\left[\frac{ln\frac{T_0}{\langle T \rangle} - \frac{1}{2}\sigma^2}{\sqrt{2\sigma^2}}\right]}{1 - erf\left[\frac{ln\frac{T_0}{\langle T \rangle} + \frac{1}{2}\sigma^2}{\sqrt{2\sigma^2}}\right]} > 1 \tag{4.9}$$

where $erf(x)$ is the Gaussian error function.

The previous result implies that, by means of a suitable threshold, we have an increase of the SNR, but at the same time we have a decrease of the overall counts in a given time, as can be seen by the following figure:

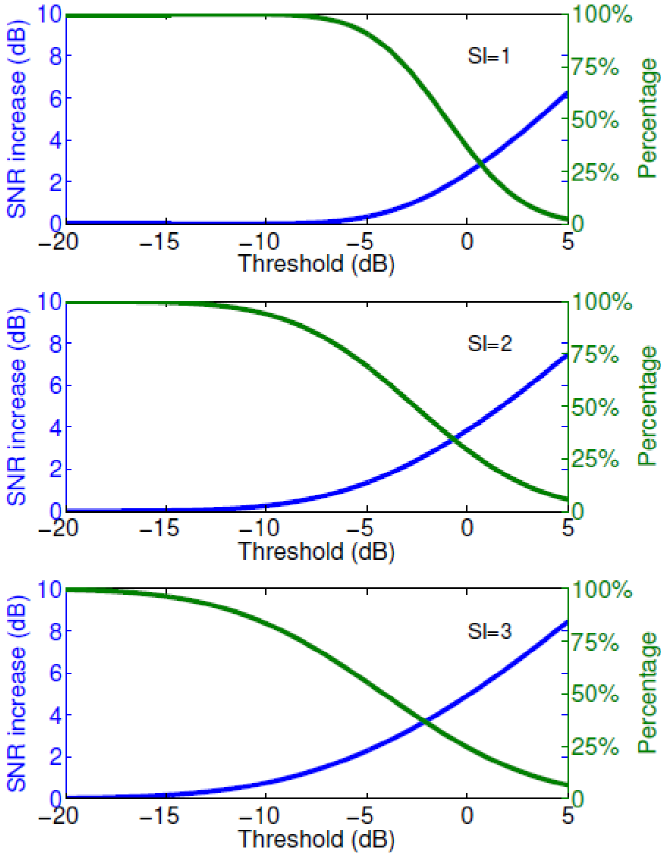


Figure 4.2: SNR and percentage of the overall counts that will be detected as a function of the threshold [29].

From this we can see that, in cases of strong turbulence and high noise, the introduction of a suitable threshold can be a successful techniques in order to support the qubit transmission by exploiting turbulence.

This theoretical model and the correspondent experimentation at the Canary islands are the indispensable basis as well as the starting point for this thesis: the data acquired in the experiment have been fully processed and analyzed by means of suitable Matlab code (*Chapter 5*), and the amazing results are presented (*Chapter 6*).

5 The analysis and design of the system

In this chapter we will illustrate the building blocks that have been implemented with the aim to process the data acquired during the experiment: the treatment are divided into topics in order to clarify the main steps. Therefore, we will deal with *Probe Signal processing* in Section 5.1, and with *Key Signal processing* in Section 5.2, while Section 5.3 is dedicated to the *Alignment of data* and Section 5.4 is for *Data processing and Optimal Threshold evaluation.*

The main relevant elements for this study are the **Probe signal** and the **Key signal**: we stress that the goal is to enlight the possibility of reducing the final key rate at the receiving side by setting a so called *Probe Threshold* (that is, the value of the probe signal whereunder we do not acquire the key signal) and that this can be done in two different manners:

- **Hardware**, that is, by imposing a *Probe Physical Threshold* in advance;

- **Software**, that is, by evaluating an *Optimal Threshold* based on the analysis of the available data.

The QBER at the receiver side can benefit of this mechanism in the peculiar cases in which the background is particularly high. In *Figure 5.1* we can see the setup of the experiment, with the transmitter, located at the Jacobus Kapteyn Telescope at La Palma, and the receiver, positioned at the Optical Ground Station of Tenerife. The transmitter uses two infrared attenuated diode lasers at a wavelength of 850 nm in order to send 0 and 1, where 0 is encoded in the vertical polarization of the photon, $(|\updownarrow\rangle)$ and 1 is encoded in the $+45^o$ linear polarization $(|\nearrow\rangle)$. We also transmit the *probe signal* at a wavelength of 808 nm; at the receiver side, a dichroic mirror is needed to separate the probe signal from the information qubits: the former is detected by an avalanche photodiode, while the latter encroaches onto a 50/50 beam splitter. At either output of the beam splitter, we have a polarizer and a single photon avalanche photodiode ($SPAD$) for the detection of the -45^o linear ($\langle\nwarrow|$) or horizontal ($\langle\leftrightarrow|$) polarization photons. Each click on either $SPAD$ bears the reception of a sifted 0 or 1, respectively [31].

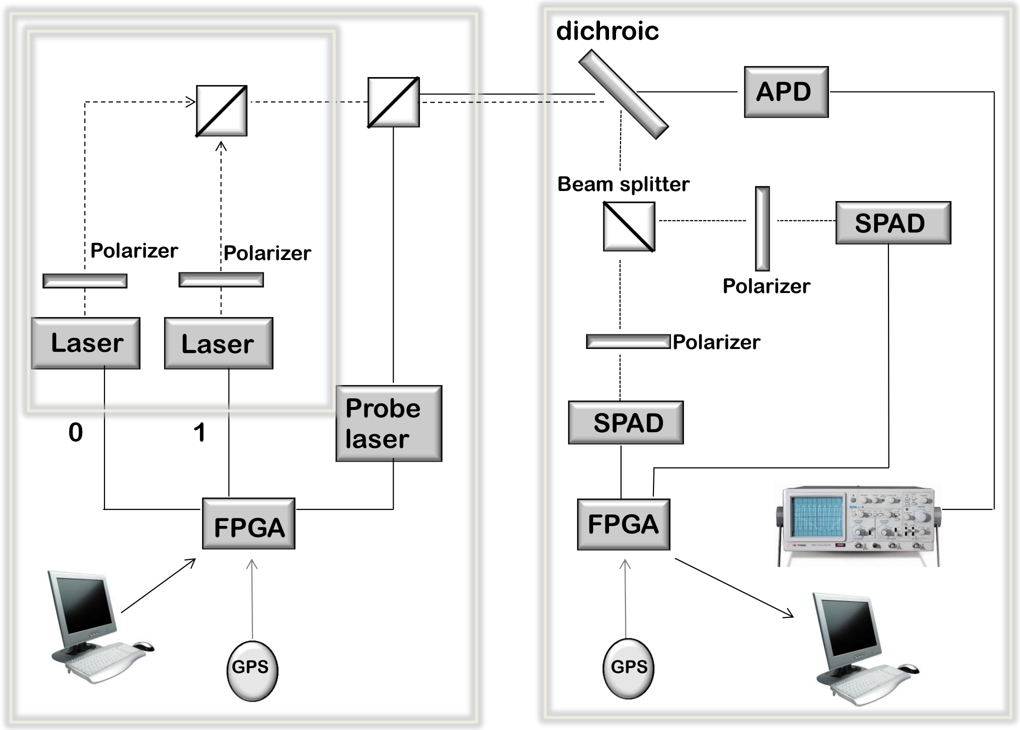


Figure 5.1: Setup of the experiment.

In order to fully clarify the role of each signal in the following elaborations, we need to introduce the main elements of this system, and the peculiarities that distinguish them.

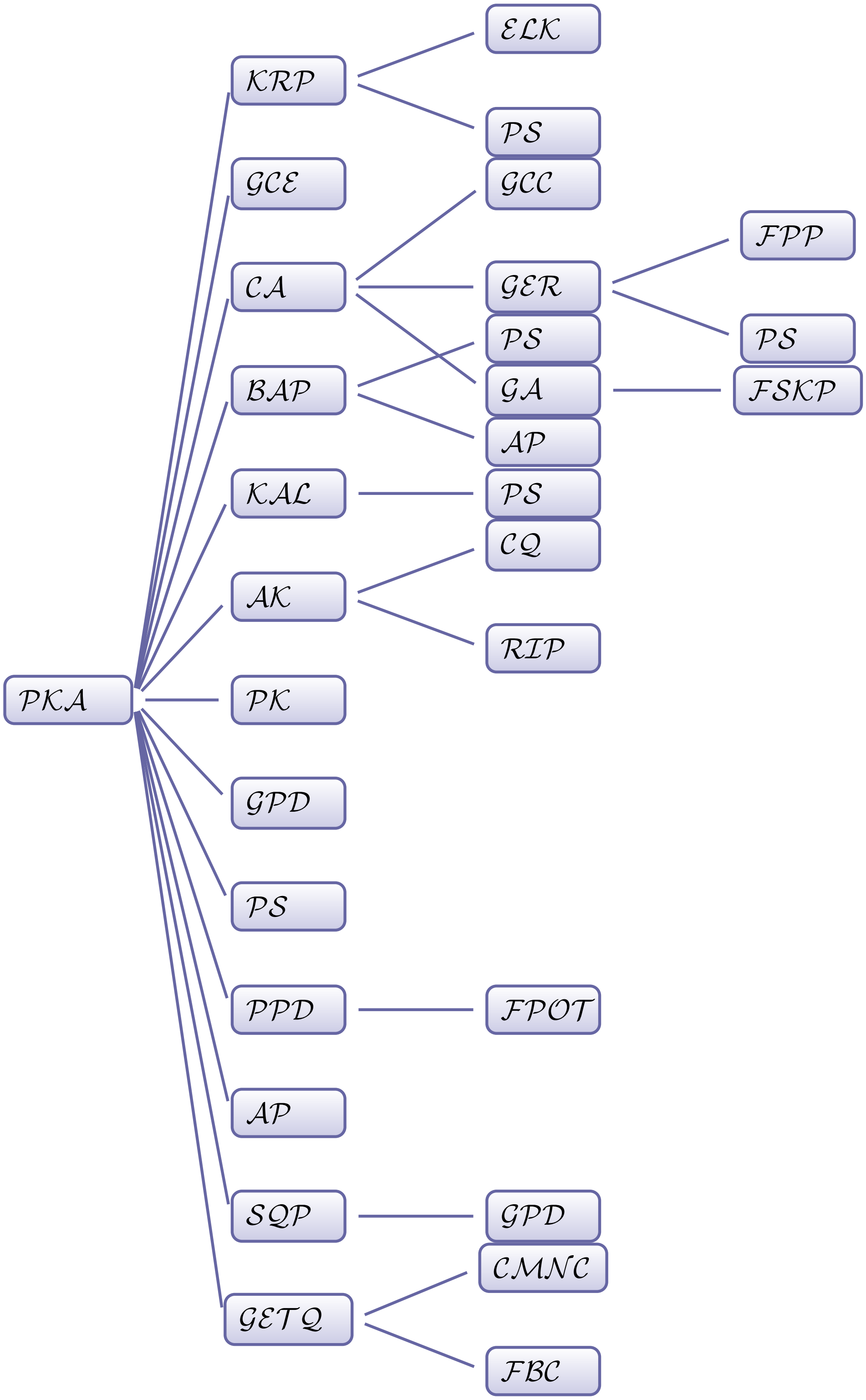
- **The Probe signal:** as was disclosed in the previous chapter, the probe signal is a *classical signal* with a wavelength of 808 nm and is sampled at the r3eceiver with a $T_{interval} = 26\ \mu s$. Thus, it is a one minute-sequence of impulses for each acquisition, and it is used as a reference to evaluate the state of the channel: the idea is to probe the channel and exploit that information, mainly in conditions of strong turbulence and high noise status, when we want to take advantage of considering only the specific moments in which the turbulence increase the transmettivity. When acquiring the signal, the majority of transmissions an hardware threshold -the so called *Probe Physical Threshold*- has been established in advance. This threshold has a variable value depending on the acquisition, and it is a sort of *physical* limit on the intensity of the probe signal, in the sense that under that value the keys have not been saved during the acquisition.

- **The Key signal:** each transmission consists of a sequence of raw keys, each one transmitted during an interval $T_k = 1s$. Each key is composed by 625 raw packets, and the packet period is $Tp = 1.6\ ms$; neverthless, in order to overcome some $FPGA$ limitations, we have to take only one packet out of ten, thus the actual maximum number of sifted packets per key will be 63. The signal wavelength is 850 nm: this value is different enough from the probe wavelength so that we don't need to deal with interference problems, but sufficiently close to allow us to assume that the behaviour of the channel observed by using the probe signal is coherent with the conditions experienced by the key signal. Such assumption will be confirmed by the results in *Chapter 6*.

The following sections are dedicated to the fundamental functions that have been implemented in order to process and analyse the data gathered during the Canary Island experiment. In order to facilitate a complete comprehension of the subsequent dissertations, we propose a diagram in which all the main functions implemented for the analysis are presented. For the sake of clarity, *Table 5.1* defines the function acronyms, so that the diagram can specify the function dependences by using them.

Acronym	**Function name**
$\mathcal{AK}$	AnalyseKey
$\mathcal{AP}$	AcquisitionParameters
$\mathcal{BAP}$	BeforeAfterAlignmentPlots
$\mathcal{CA}$	CheckAlignment
$\mathcal{CMNC}$	ComputeMeanNumberOfClicks
$\mathcal{CQ}$	ComputeQBER
$\mathcal{ELK}$	expLkey
$\mathcal{FBC}$	findBackgroundClicks
$\mathcal{FPOT}$	fixProbeOffThreshold
$\mathcal{FPP}$	FalsePositivePlots
$\mathcal{FSKP}$	findStarting_key_pkt
$\mathcal{GA}$	getAlignment
$\mathcal{GCC}$	getCorrCoef
$\mathcal{GCE}$	getChannelEstimation
$\mathcal{GER}$	getErrorRate
$\mathcal{GETQ}$	getExpectedTheoreticalQBER
$\mathcal{GPD}$	getProbeDistribution
$\mathcal{KAL}$	KEY_alignment
$\mathcal{KRP}$	keyRatePrediction
$\mathcal{PKA}$	ProbeKeyAnalysis
$\mathcal{PK}$	PrepareKey
$\mathcal{PPD}$	pruneProbeData
$\mathcal{PS}$	ProbeSelection
$\mathcal{RIP}$	ReshapeIntoPKT
$\mathcal{SQP}$	Secret_QBERPlots

Table 5.1: Function acronyms


ELK
KRP
PS
GCE
GCC
FPP
CA
GER
PS
PS
BAP
GA
FSKP
AP
KAL
PS
CQ
AK
RIP
PKA
PK
GPD
PS
PPD
FPOT
AP
SQP
GPD
CMNC
GETQ
FBC

5.1 Probe Signal processing

The first result needed for the analysis is a suitable elaboration of the probe signal. This element is indispensable because it gives informations about the state of the channel, intending to exploit this knowledge with regard to the key signal and the possibility to enhance the quality of our final secret transmission. The probe signal needs to be aligned with the key signal, in order to evaluate the correlation between probe samples and the number of sifted bits. That's why the probe signal has to be processed conscientiously, and the functions in charge for that are presented in the subsequent sections.

5.1.1 pruneProbeData

This function loads the probe signal of the acquisition under consideration and elaborates it in the following way:

- identifies the instants for which a suitable *GPS signal*, used as an initial reference and plotted in *Figure 5.2*, is greater than 3 Volts, and derives the step needed in order to pursue its rising edges. The value of $T_{interval}$ is then adjusted according to that GPS reference.

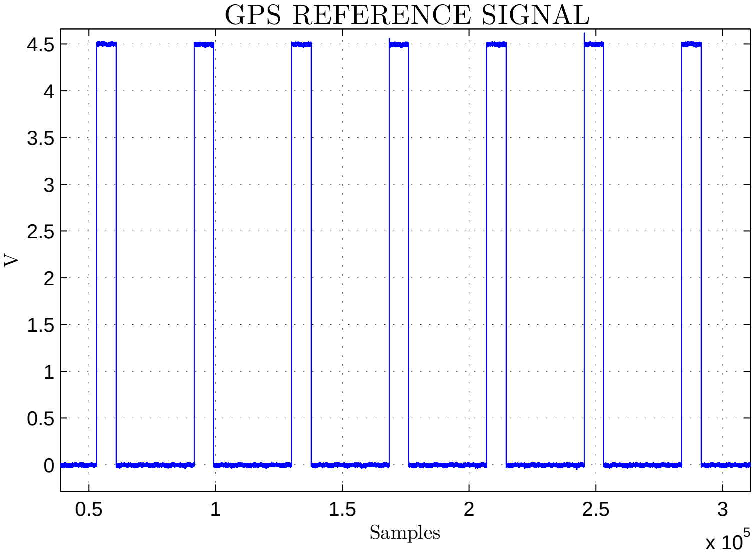


Figure 5.2: GPS reference signal

- pruneProbeData also defines the *probeOffThreshold*, that is, the threshold whereunder the probe values are esteemed to be not denotative and are thus discarded.

- The probe signal is suitably sampled with a sampling period of 1.6 ms, which is the period of the laser pulses, for subsequent comparisons with the key signal. The result is depicted in *Figure 5.3*.

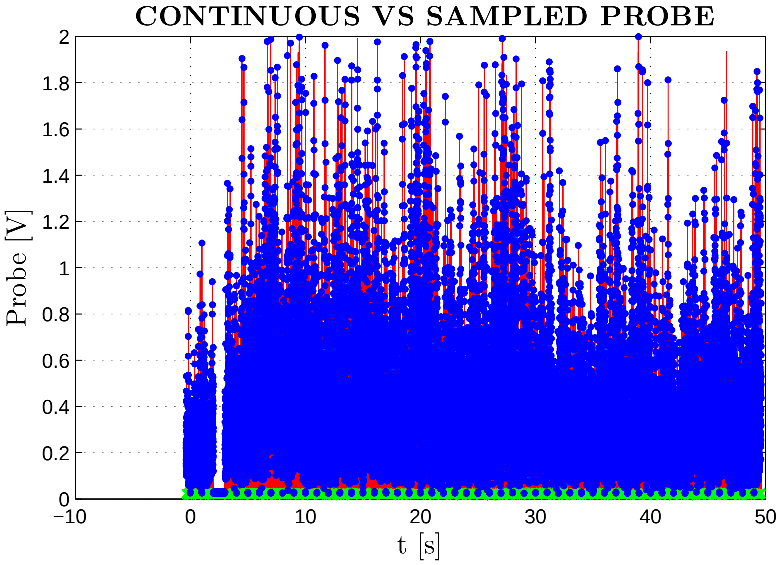


Figure 5.3: Continuous and sampled probe comparison.

5.1.2 ProbeSelection

This function allows different elaborations of the probe signal, depending on the more suitable version required by each calling function. Unless otherwise needed, the saturation value is set to the maximum value of probe detected in the current acquisition. If a physical threshold is imposed for the recovery of the correspondent key packets, the probe values under this threshold can be discarded. With reference to this circumstance, an opportune *tolerance interval* can be set to deal with the nonidealities of measurement instrumentations. Furthermore, we have the possibility to neglect the probe samples for which the probe signal is lower than *probeOffThreshold* for at least one second. In fact, we have forcedly introduced some one-second recovery holes, with the purpose of simplify the alignment between probe signal and key signal. This expedient permits to guarantee a meticulous alignment, as it enables to precisely align the probe sample and its corresponding key packet.

To clarify the functionalities, these elaborations have been divided into five different probe-processing modes:

- **Mode n.1**
  - delete probe signal if lower than *probeOffThreshold* for one second
- **Mode n.2**
  - delete probe signal if lower than *probeOffThreshold* for one second
  - Discard all the values lower than *probePhyThreshold*, by taking into account the instrumental tolerance.
- **Mode n.3**
  - Set the saturated probe values to a suitable *Saturation value*
- **Mode n.4**
  - delete probe signal if lower than *probeOffThreshold* for one second
  - Set the saturated probe values to a suitable *Saturation value*
- **Mode n.5**
  - delete probe signal if lower than *probeOffThreshold* for one second
  - Set the saturated probe values to a suitable *Saturation value*
  - Discard all the values lower than *probePhyThreshold*, by taking into account the instrumental tolerance.

For the sake of clarity, in *Figure 5.4* we show a comparison between **Mode n.3** (where there is no elaboration except the imposition of a saturation value) and **Mode n.5**, in which, additionally, the probe values lower than probeOffThreshold for at least one second are deleted, and the values lower than probePhyThreshold are discarded.

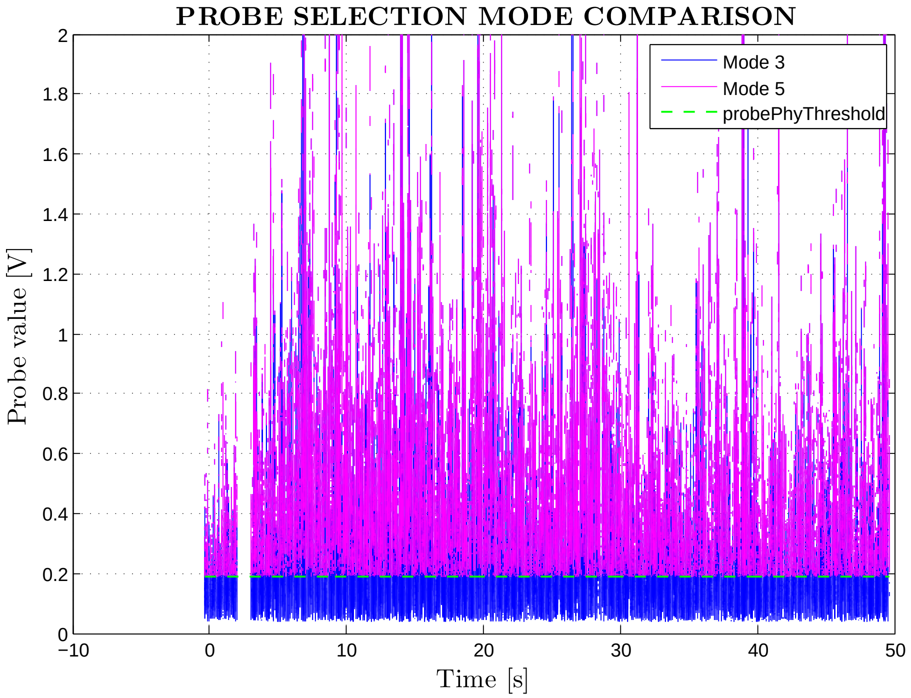


Figure 5.4: Probe selection mode comparison.

5.1.3 getProbeDistribution

The function *getProbeDistribution* performs the statistical analysis of the overall probe data and plots the corresponding empirical distribution. The *blue line* represents the empirical computation of the probe original data distribution, while the *red line* is the lognormal curve best fitting the data, obtained by finding - via maximum likelihood estimation - the parameters μ and σ of the lognormal probability density function:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\log x-\mu)^2}{2\sigma^2}} \tag{5.1}$$

The *green line* is obtained by evaluating the mean $\mathcal{M}$ and variance $\mathcal{V}$ directly from data. As μ and σ depend on the mean and variance of the lognormal distribution, we can compute these parameters by using the following formula:

$$\mathcal{M} = e^{\left(\mu+\frac{\sigma^2}{2}\right)}, \qquad \mathcal{V} = e^{\left(2\mu+\sigma^2\right)}(e^{\sigma^2} - 1) \tag{5.2}$$

Exploiting the mean and variance computed from the original data, we can then reverse the previous formula and find μ and σ, that in turn could be used to process the lognormal distribution, as in (5.1). The comparison between the curves is presented below. As a further validation of what has already been pointed out in the previous chapter, we can see that the channel statistics follows the **log-normal distribution** with high fidelity.

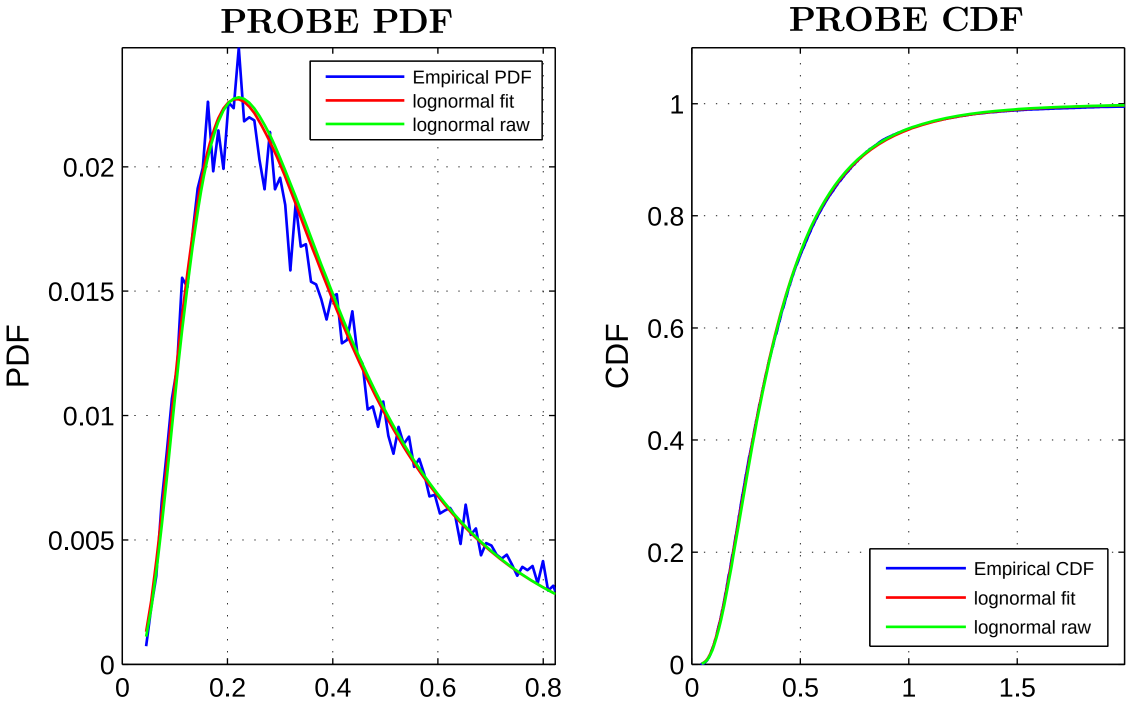


Figure 5.5: Probability distribution function and cumulative distribution function for the probe signal.

5.2 Key Signal processing

We have already underlined that one of the fundamental elements for this experiment is, obviously, the **Key signal**. This component has been conveniently processed, thanks to the functions introduced below, in order to extrapolate the main results of this analysis.

5.2.1 PrepareKey

This function represents the first component of the transmission processing procedure: for each acquisition, prepareKey provides the number of sifted Key and give them a corresponding identification code; these elements are mandatory for subsequent elaborations.

5.2.2 AnalyseKey

This function performs the analysis and processing of the key, by calling different sub-functions, such as:

- ReshapeIntoPKT
- ComputeQBER

ReshapeIntoPKT

Each key is stored in a single file where we have recorded the correspondent sifted bit sequence. We thus need to process the sifted key bits sequence of the overall transmission, and to reorganize them into the corresponding packets. In order to do that, we choose to create a matrix for each *Sifted Key* of the analysed transmission, in which each column identifies a single packet, and its elements are the correspondent packet bits. This function acts as follow:

- the function takes the *sifted* key bits at the receiver and reshapes them into a matrix ($y_{cell\{i\}}$), where i is the index of the Sifted Key under consideration, with one column for each packet.
- n_{sift} is a vector with the number of sifted bits in each packet of the current Sifted Key. The matrix $y_{cell\{i\}}$ has a number of rows which is equal to the maximum value between the element of n_{sift}; if a packet has less than $max(n_{sift})$ sifted bits, the corresponding column is padded with $NaNs$
- $x_{cell\{i\}}$ is a matrix with the same size as $y_{cell\{i\}}$, which elements are the corresponding sifted bits at the transmitter. This is done with the purpose of allowing a subsequent comparison between *transmitted* and *received* sifted bits.

ComputeQBER

This function finds the *number of errors* and *error rate* for each packet and for the whole key. The inputs are $x_{cell\{i\}}$ and $y_{cell\{i\}}$, the matrices of sifted bits by Alice and Bob respectively, for the key under consideration, and n_{sift}, the vector with the number of sifted bits in each packet. With this information, the function computes:

- n_{err}, a vector for the current key with the number of bit errors in each packet.
- $Qber_{pkt}$, a vector with the errors rate in each packet, for each key.

According to the elements defined in these sections, and going back to the calling function *AnalyseKey*, we can formalize the global variables related to the keys composing the overall transmission. Therefore we introduce:

- QBP, a matrix in which each row represents a key, and each column a packet. It was chosen to consider the maximum number of packets per keys as the overall number of columns, and to pad with $NaNs$ the possibly void spots corresponding to keys for whom the number of packets was lower that that maximum value. The element (i, j) corresponds to the *error rate* for the *j–th* packet of the *i–th* key, for the current transmission.

- N_{sift} is again a matrix in which every row represents a key, and every column a single packet, following a criterion similar to that of QBP; in this case the element (i, j) represents the *number of sifted bits* for the *j–th* packet of the *i–th* key, in the current transmission.

- N_{err} is a matrix which dimensions are analogous to the previous, where the element (i, j) corresponds to the *number of bit errors* for the *j–th* packet of the *i–th* key.

Additionally, the function *AnalyseKey* processes the elaboration of two multidimensional matrices, $ALICE_{sifted\ bits}$ and $BOB_{sifted\ bits}$: these are tridimensional matrices, corresponding to the sifted bits of the overall transmission, that is, they are related to the global set of keys for the current acquisition. The former is associated with the transmitter side, while the latter identifies the receiver side; the third dimension is necessary in order to represent each key. Therefore, each key can be seen as a distinguished bidimensional matrix, in which each column represents a packet, and each row represents the *i–th* bit of the *j–th* packet, correspondent to the *k–th* key. In *Figure 5.6* we can see a schematic representation of this criterion:

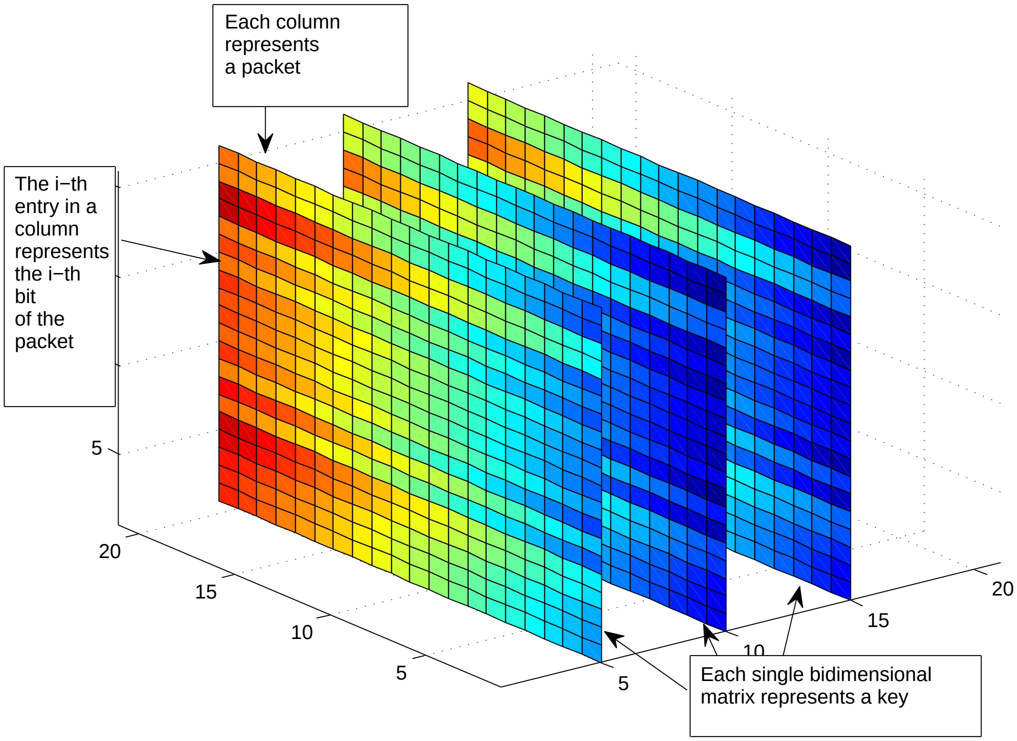


Figure 5.6: Multidimensional representation of $ALICE_{sifted\ bits}$ structure.

5.3 Alignment of data

In this section we will describe the procedure for the alignment of the probe and key signal, in order to realize a meticulous comparison between matching probe samples and packets.

5.3.1 KEY_alignment

This function aligns *sifted data* and *probe signal* basing on transmission one second-holes. It has been already emphasized that this alignment needs to be done with a precision below 1 ms, as each second of probe has to fit in extremely well with the correspondent key, in order to evaluate their consistency. The outputs of this function are:

- t_{ok}, which is the alignment shift;
- the matrix t_p, where the element (i, j) represents the starting time of the *j–th* packet of the *i–th* sifted key;
- t_{sift}, which is the alignment time axis for the sifted key, realized by adding to the first reference of t_p -and for each key- the correct shift value t_{ok}. Therefore, t_{sift} is a column vector, which *i–th* element corresponds to the time instant related to the beginning of the *i–th* key of the current transmission, after the suitable shift.

We need the following parameters:

- t_s, which is the time axis for the sampled probe signal D_s;
- N_{sift}, the matrix already presented in the previous section, in which the element (i, j) corresponds to the number of sifted bits in the *j–th* packet of the *i–th* sifted key;
- $Jump = 10$, which is the packet decimation factor due to FPGA problems;
- T_p, the packet interval;
- $Npkt_{raw}$, correspondent to the number of packets per raw key;
- max_{delay} which is the maximum alignment error that can be corrected during implementation.

Aligning is performed automatically basing on the *minimum of the correlation* between the sifted bit data and a binary mask based on missing blocks of probe data. In *Figure 5.7* we can clearly see the results of this alignment procedure:

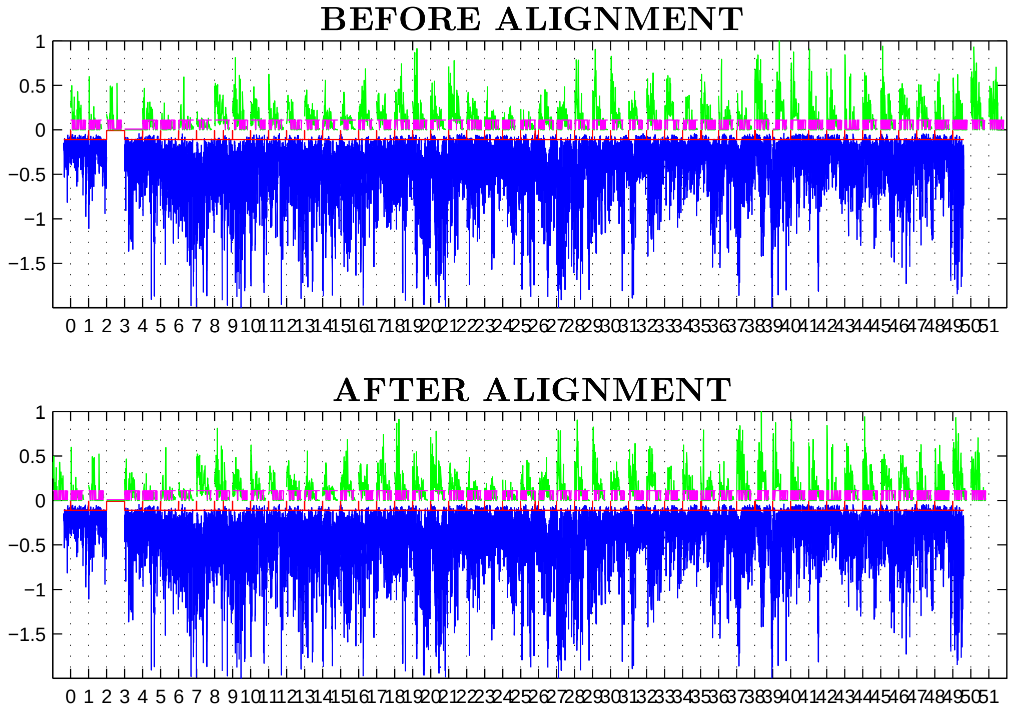


Figure 5.7: Result of the alignment procedure implemented by *KEY_alignment*.

5.3.2 checkAlignment

The main outputs of this function are the matrices tk_m, Ds_m, Ns_m, where -once again- each row represents a specific key of the acquisition under consideration, while each column corresponds to a packet of the *i–th* key. Therefore, the element (i, j) of tk_m is filled with the specific temporal instant related to the *j–th* packet of the *i–th* key, while the element (i, j) of Ds_m, Ns_m includes respectively the value of probe and the number of sifted bits matched to the *j–th* packet of the *i–th* key.

The sub-function **getAlignment**, called by *checkAlignment*, performs a further refined alignment, no no longer of each key but now on each packet, in order to get the correspondent probe value for each of them. This is crucial to evaluate the consistency between different values of probe and the equivalent number of acquired sifted bits. To implement this procedure, we exploit the probe temporal axis t_s as a reference: for every element of t_s, we check if that instant is matched with a time istant of the sifted packets. If so, we take the t_s–index of that instant and record it: this will be used to get the correspondent elements of D_s. If the transmission (see as a concatenation of packets) and the probe signal have different temporal length, then only the common temporal interval wil be considered for the analysis.

The sub-function **getErrorRate** processes the probe values and the number of sifted bits for each packet; the aim is to establish a *rate of valid acquisition* according to the following criterion: if the current transmission has required the establishment of a *Probe Physical Threshold*, then we already know from the above that we need to check if the acquisition of a packet has been catched when the probe value was over that threshold (remind that we can impose a suitable instrumental tolerance if needed). If this is not the case, then we can have two different anomalous cases:

- the packet acquisition in a specific instant has snapped even if the probe value was under the *Probe Physical Threshold*: this case is called a **false positive**;
- the probe values of the sample under consideration was over the *Probe Physical Threshold*, but we do not acquired any key signal: this case is labeled as a **missed acquisition**.

In turn, we have a so called **valid acquisition** when, for a specific time instant, the value of probe was over the *Probe Physical Threshold*, and we acquired the key properly. The count of this records for each packet of the global transmission allows to determine a *valid acquisition rate* R_{VA} according to the following formula:

$$R_{VA} = \frac{n^o\ of\ valid\ acquisitions}{total\ n^o\ of\ acquisitions} \tag{5.3}$$

The value of R_{VA} is very close to unit for most of the transmissions, as a further demonstration that there was a good correspondence between probe and key packet during the acquisition mechanism.

5.4 Data processing and Optimal Threshold evaluation

This section represents the core of the overall analysis, as the functions described beyond provide the computation of the quantities such as the *Optimal Threshold* that allow us to derive the results that will be presented in *Chapter 6*.

5.4.1 KeyRatePrediction

The main purpose of the processing realized by this function, is to evaluate if it is possible to obtain a tangible benefit from the thresholding: recalling what has been introduced in the previous chapter regarding the imposition of an optimal threshold, we desire to see if that **Optimal Threshold** -that is, a suitable probe value for which it is possible to obtain a largest number of secret key bits, or to realize secret key even in conditions that generally would not allow that- actually exists. Therefore, we need to determine the probe value for which we have the maximum number of *secret bits* and, if this optimal probe value is different from the Probe Physical Threshold imposed during the acquisition, we save this value in a suitable vector, one for each transmission. In particularly critical conditions of high background, we will see that a significant increase in the number of secret bits is effectively possible, by imposing the suitable value of optimal threshold. Additionally, we will emphasize that in some specific cases the improvement is not only in the number of secret bits, but rather it lies in the fact that without the optimal threshold it wouldn't be possible to share a secret key, while the introduction of this new expedient allow us to make the secret key exchange possible.

The main task of *KeyRatePrediction* is to process the results on the *sifted bits*, *secret bits* and *QBER* trends as a function of the values of probe threshold, by calling the sub-function **expLkey**, which evaluates the expected key length depending on the scintillation threshold (and also yields the corresponding sifted key length and QBER). The outputs are:

- *keyBitsThr*, a vector containing the bound on the maximum number of **final secret key bits** for each threshold th_i, computed by using the asymptotic bound:

$$K_{sec} = K_{sift}(1 - 2H(QBER)) \tag{5.4}$$

- *th*, the probe thresholds vector. Unless otherwise specified, for the realization of this vector all the values of the overall transmission are used, once they have been duly reorganised in ascending order.
- *siftedBitsThr*, a vector containing the overall number of sifted bits corresponding to packets for which probe signal is greater than threshold, for each threshold th_i
- *QberThr*, a vector containing the overall *QBER* corresponding to packets for which probe signal is greater than threshold, for each threshold th_i

The optimal threshold is fixed in correspondence to the value of probe that permits to obtain the maximum value of *keyBitsThr*, that is the **maximum number of secret bits**. *KeyRatePrediction* supplies the following outputs, needed for the graphic elaborations and consequent evaluations of *Chapter 6*:

MatrixOfValues: the rows of this matrix are representative of the keys in the current acquisition (just one row if we want to evaluate the overall transmission in a global way), while each column represents:

- **Column n.1** is the number of *sifted bits* when no threshold has been established (in that case the minimum allowed value is the Probe Physical Threshold, set before the acquisition)
- **Column n.2** represents the *expected secret key length* when no threshold has been established;
- **Column n.3** is the *QBER* value when no threshold has been established;
- **Column n.4** corresponds to the probe signal *optimal threshold value*, evaluated in correspondence to the maximum number of secret bits;
- **Column n.5** represents the fraction of packets for which the probe signal is over the optimal threshold value;
- **Column n.6** stands for the number of *sifted bits* related to the established optimal threshold value;
- **Column n.7** is the *expected secrey key length* related to the established optimal threshold value;

- **Column n.8** corresponds to the $QBER$ value when the optimal threshold value has been established.

A further output of the function under consideration is $PKTalign_{cell}$; it stands for a cell array which most impressive elements are the following:

- **The overall matrix th**: it is composed by a number of rows equal to the number of keys for each acquisition (it is also possible to select the analysis of the overall transmission, in this case this matrix will have a single row). Each row contains the probe values considered as thresholds for the investigation.

- **The overall matrix Ls**, which structure is equivalent to that of th, and in which rows we record the value of *siftedBitsThr* computed in *expLkey*, that is, the vector containing the overall number of sifted bits corresponding to packets for which probe signal is greater than threshold, for each threshold th_j (where j is now the column ID).

- **The overall matrix Lk**, with a structure equal to the previous, in which rows we set the values of *keyBitsThr*, that is, the bound on the maximum number of final secret key bits for each threshold th_j.

- **The overall matrix Qbth**, again with the same number of rows and columns as the matrices that have been already defined, in which rows we record the values of *QberThr*, the vector containing the overall QBER corresponding to packets for which the probe signal is greater than threshold, for each threshold th_j.

Figure 5.8 shows the sifted and secret key length, the $QBER$ and the average number of sifted bits per packet, computed over the packets for which the correspondent probe value was greater than th_i; all are plotted versus the probe threshold, within a single transmission in which the level of background noise is high. Each dot mark represents a single value of $QBER$ or *sifted bits* related to a single packet, corresponding to one probe measurement. As expected, by enlarging the threshold value we suffer for an overall sifted key length decreasing, but in turn we can benefit of a decrease in the $QBER$ and an increase on the average number of sifted bits per packet, again as a function of threshold. Furthermore, we can clearly see that the number of secret key length grows and reaches the maximum value for a suitable optimal threshold. This results are consistent with our target, that is, to find a threshold for which the increase of the SNR overcomes the sifted bit decreasing, in such a way that the overall secret key length benefits from the imposition of that treshold.

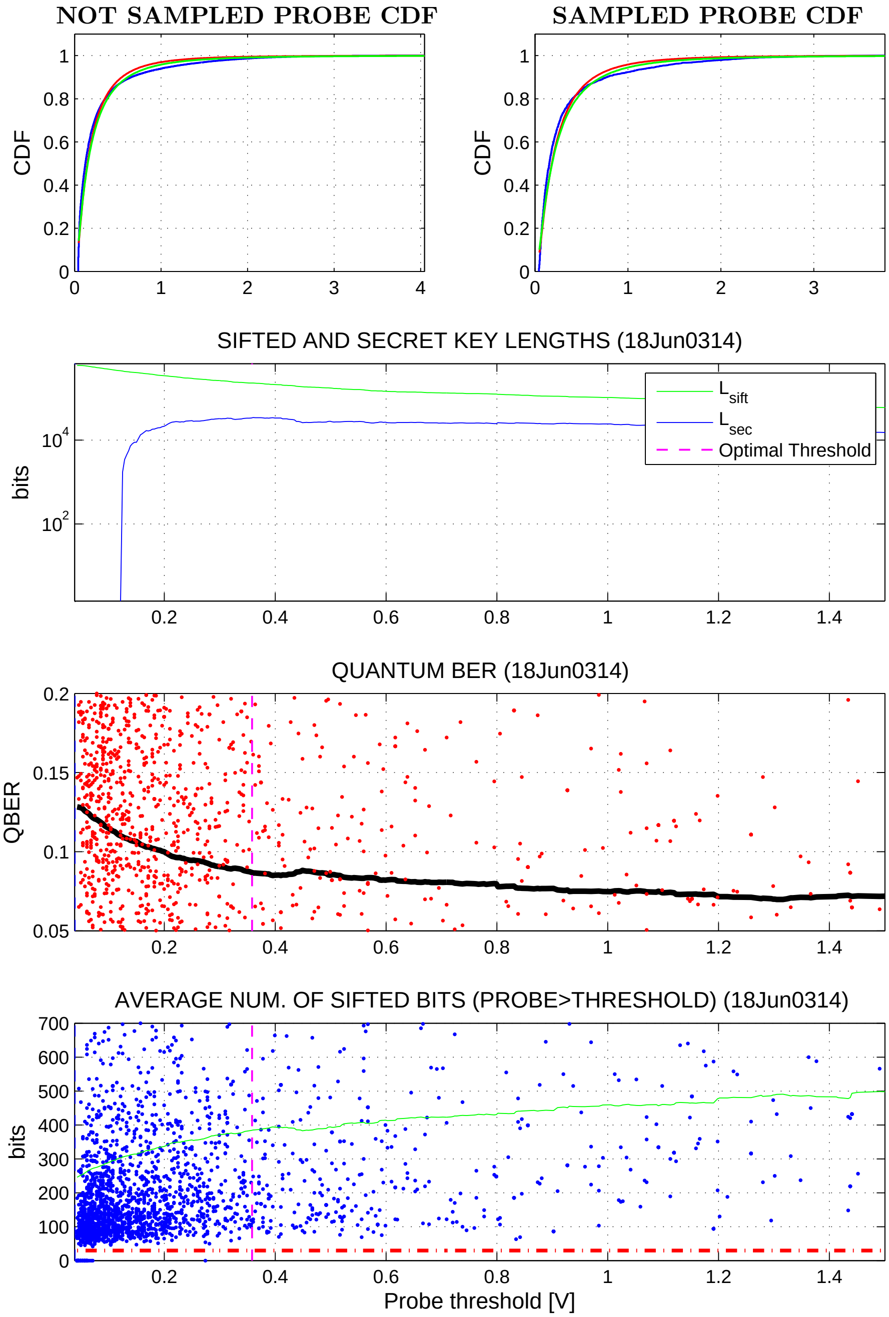


Figure 5.8: Sifted and Secret key length, QBER and average number of sifted bits as a function of threshold.

All these results will be discussed thoroughly and the important conclusions will be presented in the following. For the sake of completeness, we finally need to introduce the function **ProbeKeyAnalysis**, that manages the overall processing of data and the analysis of results. This function has the task to call the previous functions and sub-functions, to administer the variables and to realise the exemplifying plots for a graphic evaluation of the main results. This can be done for each acquisition, while the global analysis of the set of the overall transmissions is committed to an apposite script, calling *ProbeKeyAnalysis* for each data capture and recording the total results in order to compare them and inspect the trends for the various cases. In conclusion, in this chapter we have presented the basic building blocks implemented in order to realize the processing of data.

6 | Analysis of results

This chapter is devoted to the disclosure of the remarkable results that have been obtained from the data analysis described in the previous chapter. The sections will respectively focus on the **probe distribution** -*Section 6.1*- where we will see that as expected it follows a typical lognormal statistics, and on **the consistency between Probe and Key signal** -*Section 6.2*- in which an interesting parallelism between probe signal and key signal is presented. *Section 6.3, Section 6.4* and *Section 6.5* are devoted to the **sifted key length**, **QBER** and **secret key length** respectively, and we will see that an important validation of our *threshold-proposal* for the improvement of the quality of transmission and secret key exchange can be drawn from this analysis.

6.1 The probe distribution

In the previous chapter we have presented the fundamental results related to the probe distribution. In this section we will give a further insight and we will clarify the implications of our probe elaboration. In the introductive chapter it was pointed out that, so far, the main obstacle related to free-space propagation was the presence of atmospherical turbulence, and that as a consequence the statistics of the received signal amplitude is transformed from Poissonian to Lognormal. If this is the case, then the peculiar transmissivity peaks related to a lognormal distribution can be exploited in order to obtain secure key even in bad conditions. The analysis of the probe data for the overall acquisitions clearly shows that the lognormal behaviour is confirmed (see *Figure 6.1*).

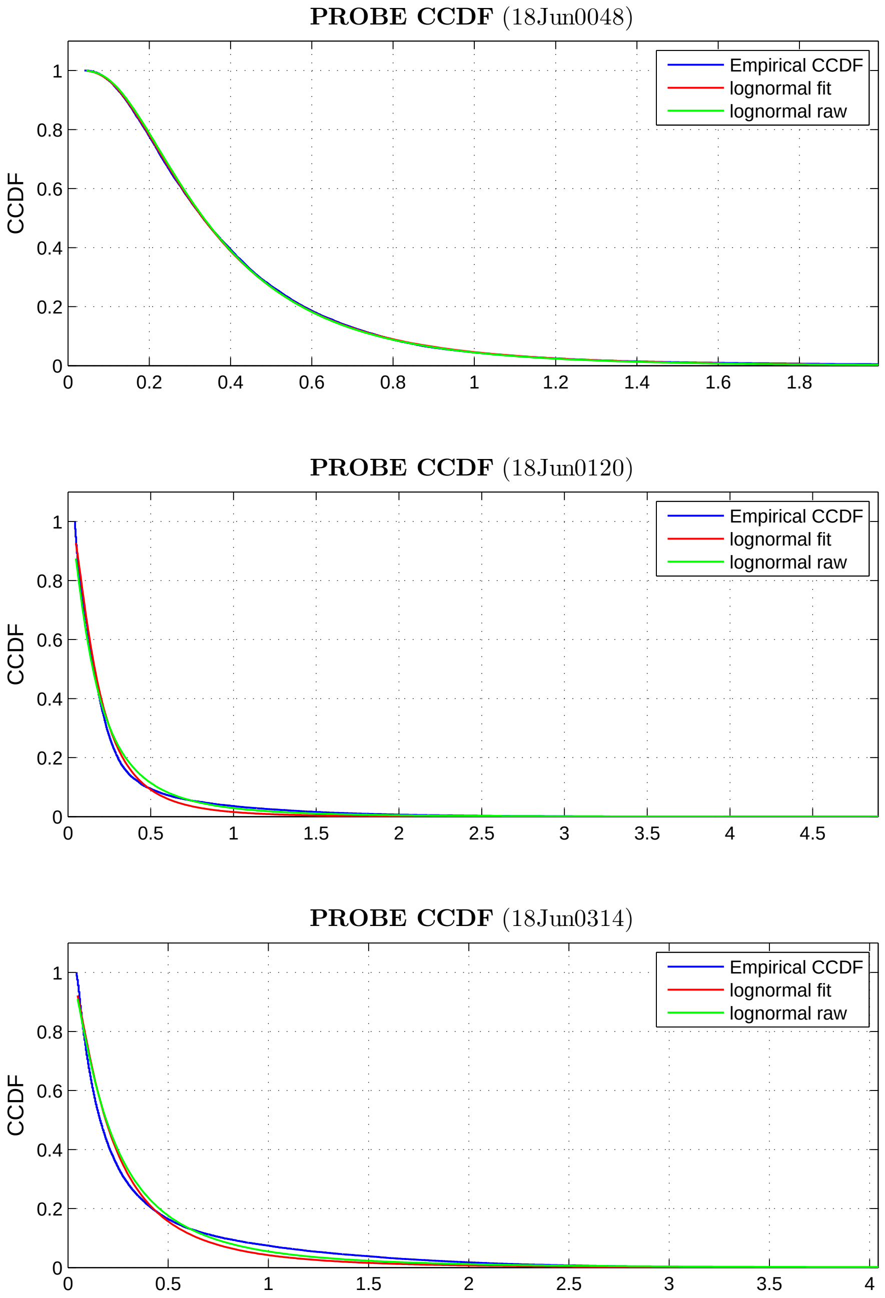


Figure 6.1: CCDF of the probe signal, sampled at 1.6 ms, for three different transmission sessions. The lognormal CCDF is also shown for comparison.

This result is important because it allows us to process the trasmission data with the aim to test the feasibility of our *turbulence exploitation*. Therefore, due to the lognormal statistics of the channel we expect a few intensity peaks. In conclusion, we have validated the correctness of the lognormal-distribution hypothesis for our channel: this means that we are dealing with a situation in which our turbulence exploiting approach is viable, as the intensity peaks caused by turbulence can in principle allow a quality enhancement of secure transmissions.

6.2 The consistency between Probe and Key signal

This section is devoted to the comparison between probe and key signal samples, with the aim to validate the correspondence between high probe values, and a high count of bits in the related packet of information. In *Figure 6.2* we can see qualitatively how this consistency is actually confirmed by the experimental results: lower values of the probe signal correspond to fewer counts for the acquisition, and viceversa.

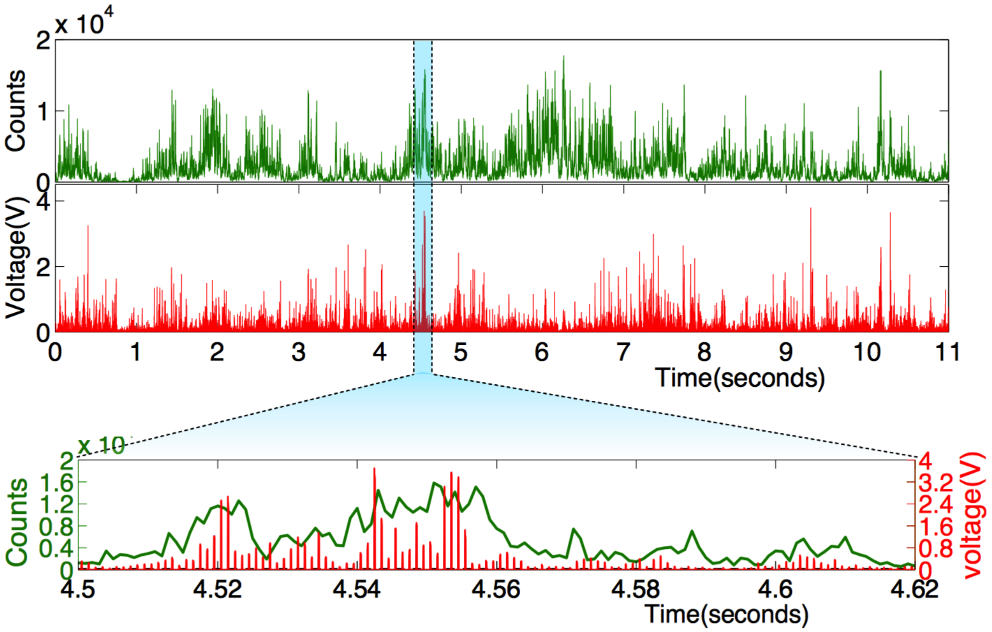


Figure 6.2: Consistency between probe signal and acquisition counts.

We recall that our aim was to exploit the information obtained thanks to the observation of the probe signal in order to catch instants of high transmettivity and acquire the correspondent key signal. At the basis of this formulation there was the hypothesis that the probe and key signal experience the same channel status, so that instants of high probe values match instants of high counts in the key signal. This correspondence allows us to validate the intuition expressed at the very beginning of our dissertation: we can probe the channel by means of a classical signal, and exploit that information to acquire the photon signal if and only if the probe value is above a given threshold; this in fact guarantees that the collected signal will have a correspondent higher number of counts, and an enhanced *SNR*.

6.3 Sifted key length with probe threshold

The first thing we need to observe about the sifted key length is that its behaviour as a function of threshold is in itself decreasing. This is because we are only considering the acquisitions of packets for which the probe signal was above a given threshold, and this selection causes the overall number of sifted key bits to lower down, as we will have a decrease of the counts in a given time. Neverthless, we would like to see whether there is a consequent global benefit on imposing a threshold. We can recognize this improvement by observing the behaviour of the **average** number of sifted bits **per packet** as a function of threshold (*Figure 6.3*): this value grows with the threshold (at least up to the mean received probe amplitude). This result reveals that as a matter of fact the threshold imposition implies a *SNR* enhancement and a consequent increase of the average number of sifted bits per packet, concurrent with an enrichment of the overall performances of the system; this is because thresholding involves a decreasing of the overall counts, but as they are accurately selected they appear to be qualitatively excelling. The following figure shows an example of average number of sifted bits per packet's growth for a transmission that we will take as a reference in the following: it corresponds to the acquisition carried out on June 18th, at 03:14 a.m. in conditions of high background and strong noise status, that is, perfectly matching with our presuppositions and requirements.

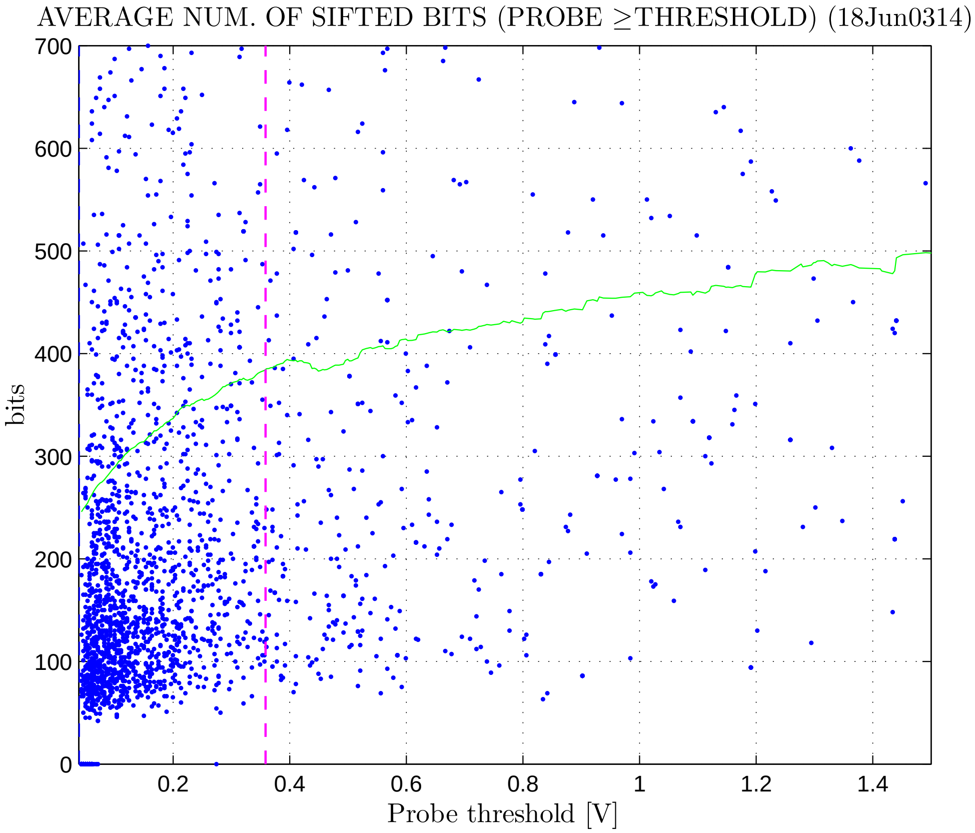


Figure 6.3: Average number of sifted bits per packet, correspondent to values of probe greater than threshold (x axis).

As a further element of appraisal, we evalutated the normalized trend of the average number of sifted bits per packet for a given set of acquisitions, as a function of the normalized threshold; the factor of normalization is the mean value of the overall probe signal. Additionally, we propose a polynomial fitting of second degree for these trends, with the aim to evaluate the parameter's progression with further accuracy; both plots are depicted in *Figure 6.4*.

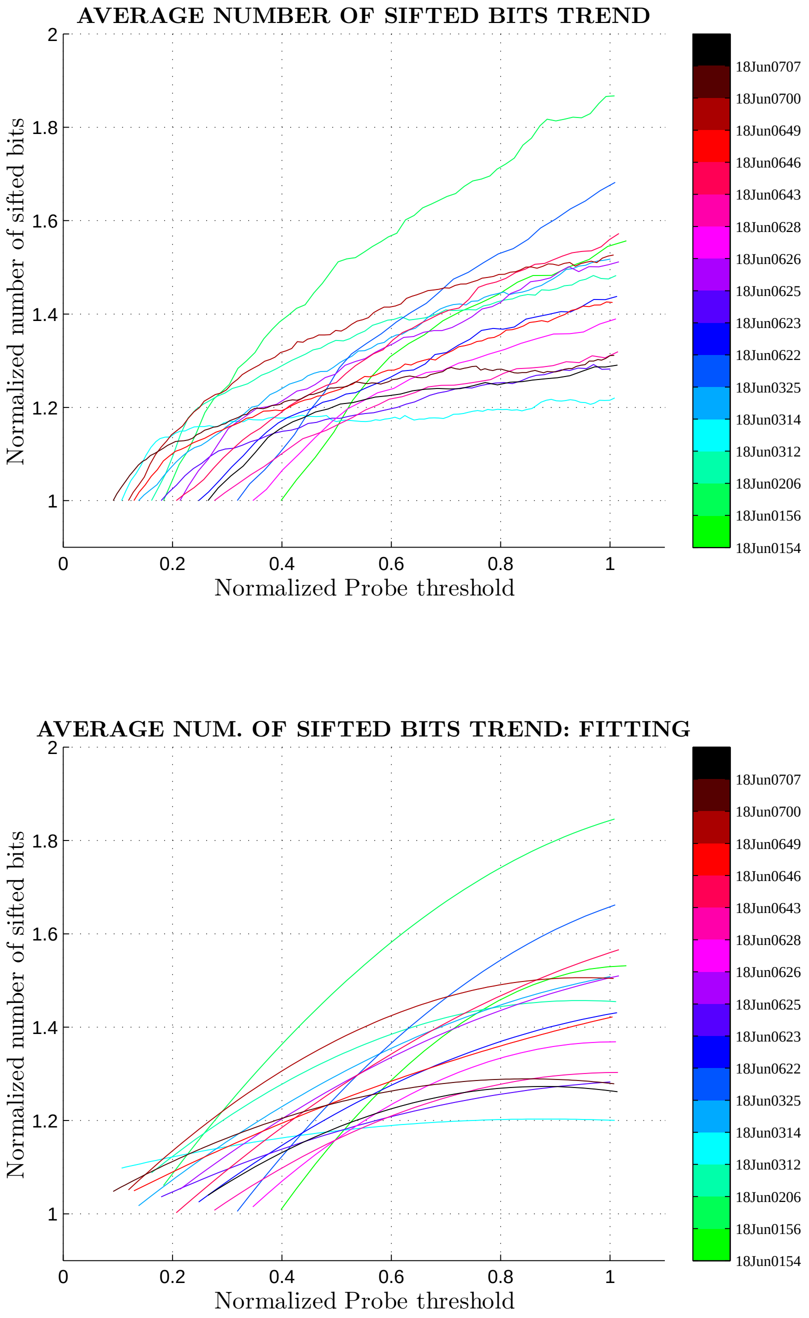


Figure 6.4: Average number of sifted bits per packet trend, and 2^{nd}-degree fitting

The main conclusion that can be deduced is thus that a suitable threshold establishment allows an improvement and a growth of the average sifted bits per packet, as a further validation of the fact that the threshold imposition corresponds to a *SNR* enhancement and thus the quality of the acquisition is in inself improved. For comparative purpose we propose a photo sequence that characterizes the background and brightness conditions for three of the acquisitions previously considered (*Figure 6.6-6.8*).

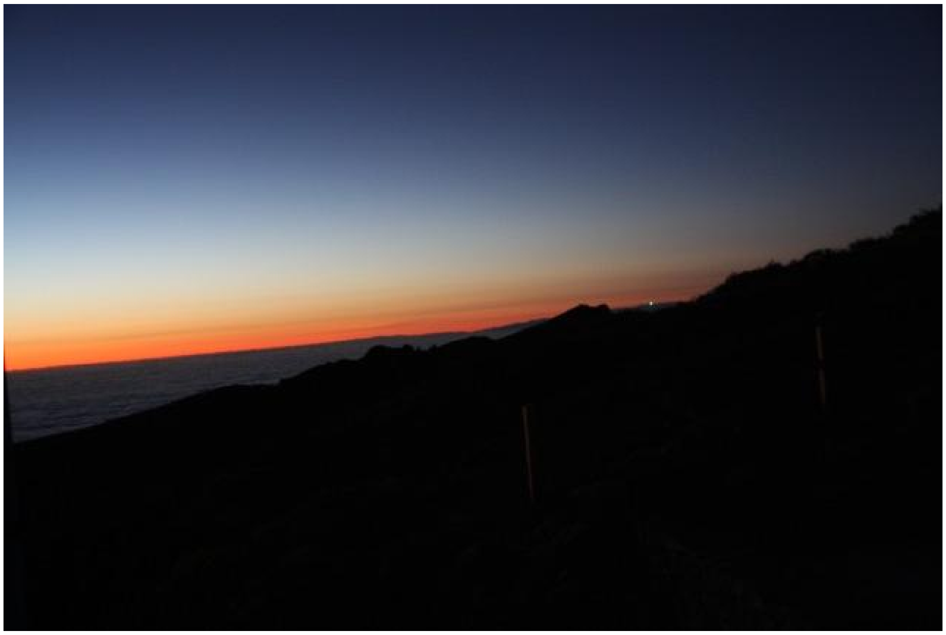


Figure 6.5: Background and brightness conditions at 06:27 a.m.

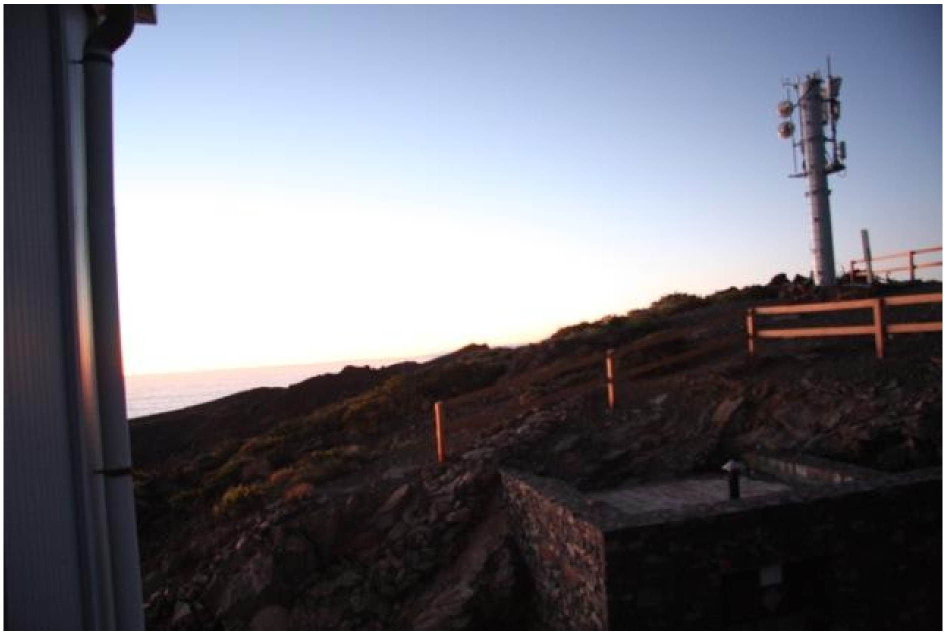


Figure 6.6: Background and brightness conditions at 06:41 a.m.

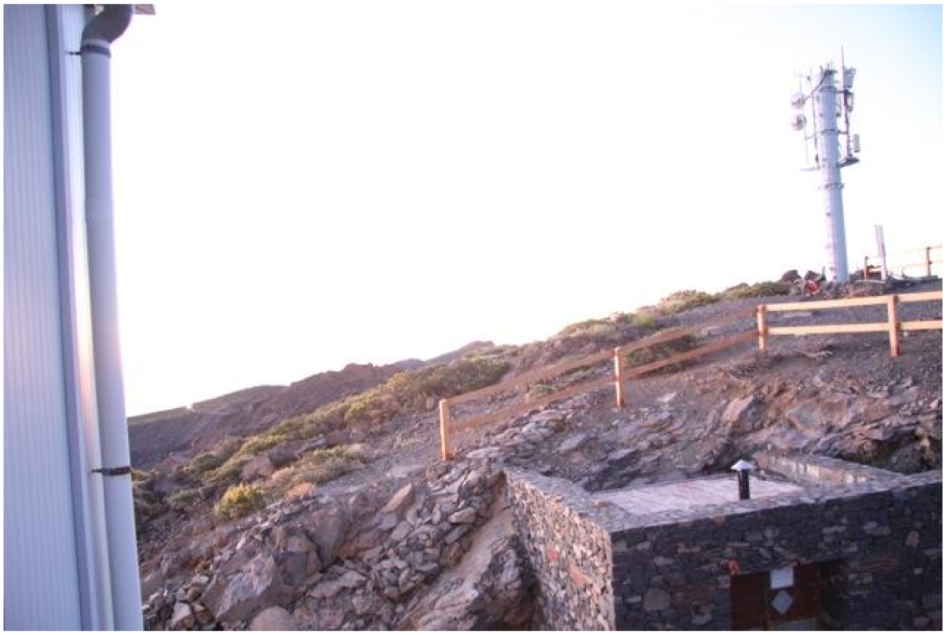


Figure 6.7: Background and brightness conditions at 06:50 a.m.

6.4 QBER with probe threshold

In this section we will examine the *QBER* behaviour as a function of threshold. In order to verify what was claimed in the previous section, we would like to see that we have a concrete benefit on the *SNR* and on the overall transmission quality, thanks to the thresholding approach, by checking for a *QBER* decrease. *Figure 6.8* definitely shows that, in condition of particularly high background, this improvement is effectively possible: as the threshold grows -provided that its value is reasonable-, we can see a clear decreasing in the *QBER* parameter.

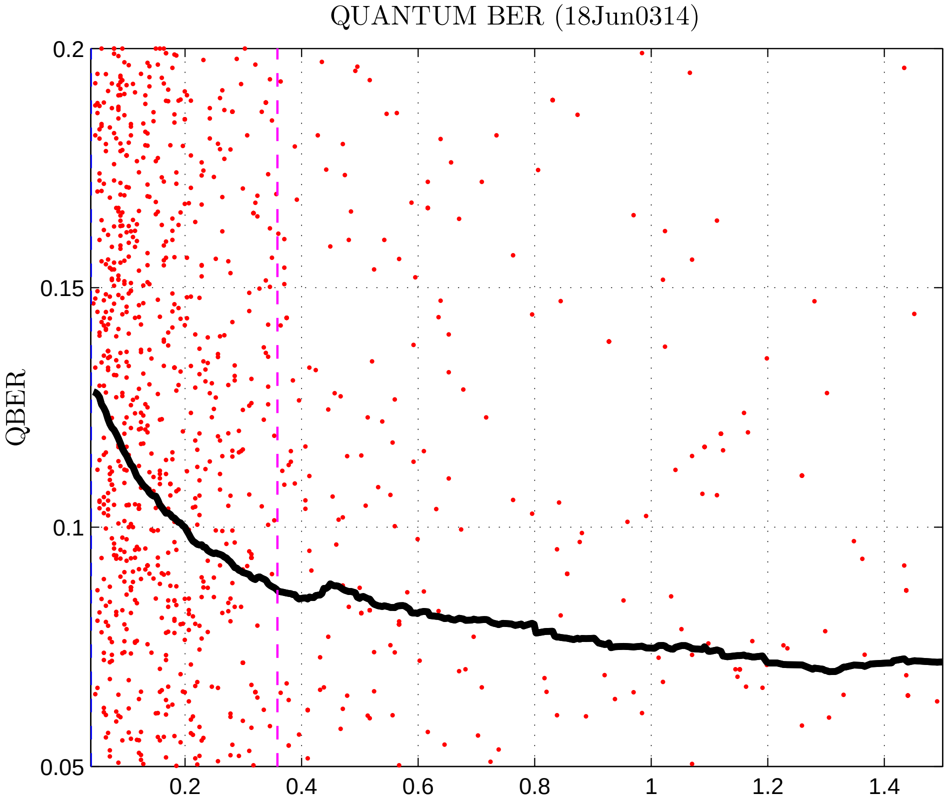


Figure 6.8: Quantum BER as a function of threshold.

6.5 Secret key length with probe threshold

We have already pointed out in section 6.3 that the sifted key length as a function of threshold will necessarily experiences a decrease, since we intentionally take into consideration only the packets correspondent to probe values greater than the threshold established. Neverthless, we have in turn underlined that we have a consequential growth of the average number of sifted bits per packet, following a *SNR* enhancement and an improvement of performances and quality of the system. *Figure 6.9* shows that the sifted key length effectively decrease with threshold, but in turn we can see an astonish result: in conditions of high background and high noise status, the asymptotic bound on the secret key length achivable from the system can experience a maximum peak in correspondence to a suitable value of threshold named **optimal threshold**. Furthermore, in this peculiar case we can see that the number of resulting secret bits was zero when no threshold was imposed, while the threshold set up involves the amazing possibility of obtaining a considerable amount of secret bits from the secret key exchange, that was previously unfeasible and inconceivable. This is a stunning result, because it validate what was announced in *Section 3.2.3, equation (3.10)*: by selecting only the transmissions with low QBER -thanks to thresholding-, we guarantee that the QBER decreasing overcomes the sifted bits loss, in order to obtain an overall enhancement of the secret key rate. This result suggests that, with suitable expedients and thanks to the threshold imposition, it is possible to obtain a secret key exchange even in critical situations of high background that would generally inhibit its realization.

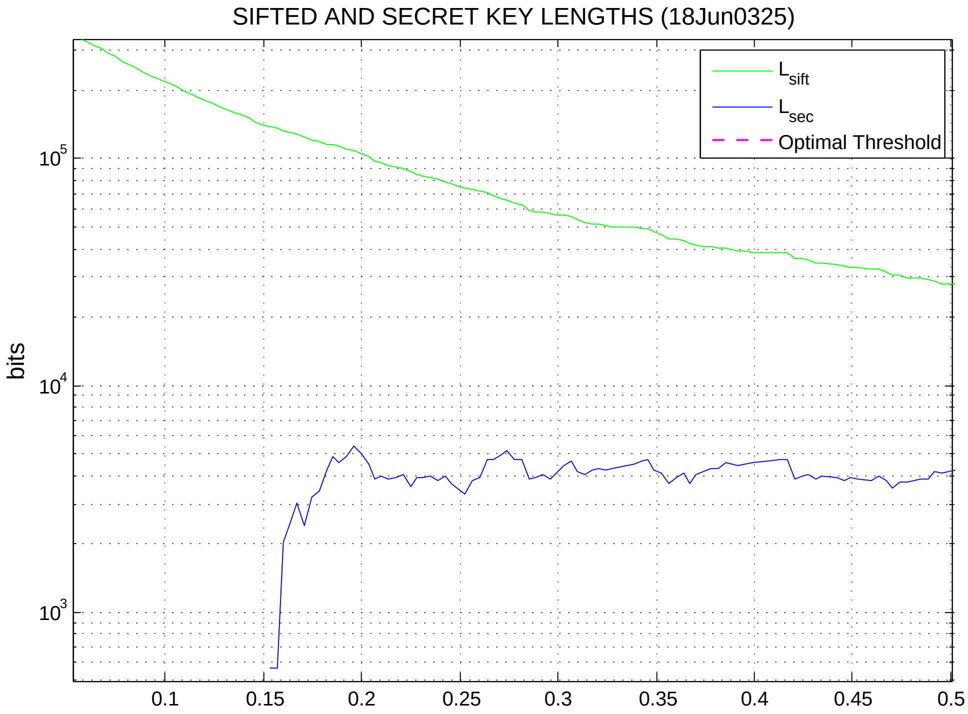


Figure 6.9: Secret and sifted key lengths for a specific acquisition (at 03:25 a.m.).

The considerations deduced in the previous sections can be gathered in a single figure in order to evaluate the effect and the importance of the achieved results: we can clearly see that, in conditions of high background and luminosity, the imposition of an optimal threshold involves a *QBER* decreasing and a correspondent increasing of the secret key length, allowing to realize in principle a satisfying secret key exchange even in conditions considered -until now- impossible. With *Figure 6.10* we have a further validation that the *SNR* decreasing is remarkable enough to compensate the decrease in the number of counts detected, due to the choice of acquiring only the packets correspondent to probe values greater than the imposed threshold. This result persuase us to support the idea of the *exploiting turbulence approach*, and to propose it as a promising countermeasure for the realization of an efficient secret key exchange system, even in critical channel situations.

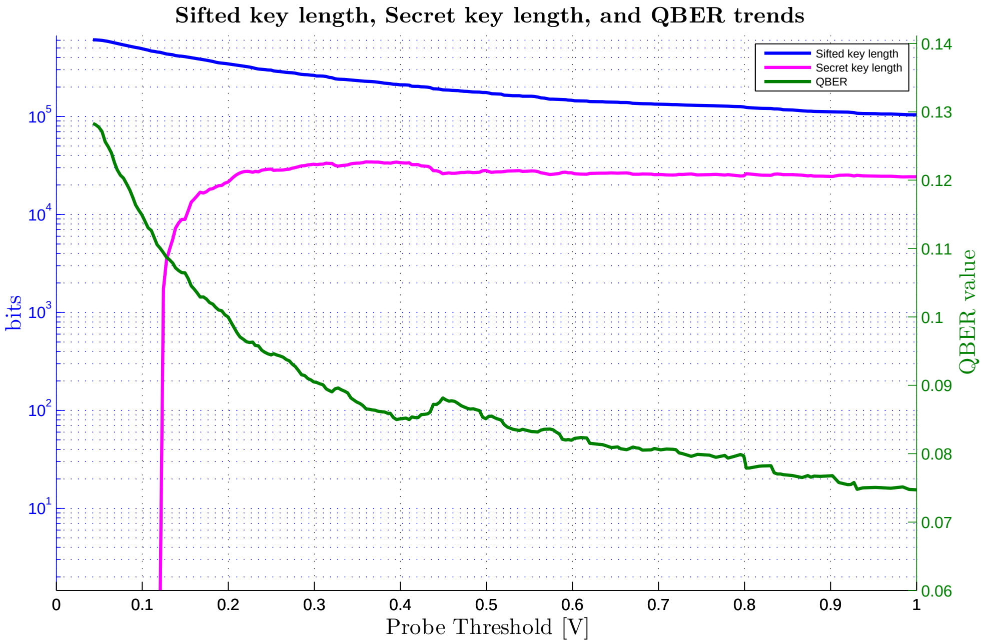


Figure 6.10: Sifted key length, Secret key length, and QBER trends for acquisition at 03:14 a.m.

6.6 The benefits of a threshold establishment

An additional and impressive result can be deduced by comparing the *QBER* values observed for acquisitions with high background levels. We have already pointed out that the threshold can be *hardware* or *software*: in the former we imposed a Probe Physical Threshold in advance, that is a probe value whereunder we do not acquire any signal -physically speaking-, while in the latter we have subsequently evaluated an Optimal Threshold based on the analysis of acquired data: the samples beneath that value have been simply ignored. We will see that the *software* approach is really valuable when dealing with transmissions that are not strongly hardware-limited, while it faces some limitations when used with acquisitions showing a preimposed high hardware threshold; this is due to the fact that if the hardware probe picking is significantly selective, we will deal with a restricted amount of key data for the statistics, as the number of packets over the hardware threshold is itself limited.

The elaborations of data that have been done with the functions presented in *Chapter 5* allow us to fully analyse all transmissions; the most relevant for our purpose are the acquisitions for which the background was sufficiently high: this is because we can emphasize the benefits of imposing a threshold when the SNR can be enhanced, thus when the QBER presents initial high values that we are trying to improve. Therefore, in the following we will deal with transmissions that have experienced substantial background noise conditions, and for which the preimposed hardware threshold was not excessively high. For these acquisitions, *Figure 6.11* shows a comparison between the starting *QBER* values (that is, the values without software thresholding, but with a possibly hardware preimposed threshold), represented by yellow dots, and the optimal *QBER* values, corresponding to the **Optimal probe threshold** value and evaluated by finding the maximum on the secret key length (blue dots). The first acquisitions are representative of a situation in which the *QBER* value at the very beginning is already low, so that we cannot experience any threshold-based improvement; nevertheless, for the subsequent recoveries we can foresee two interesting results.

1. For our first observation we will consider acquisitions with the same background conditions (characterized by a background identification that have been recorded during the recovery of data) such as:

   - 18Jun0325
   - 18Jun0328
   - 18Jun0329

- 18Jun0331

The first transmission shows a very low hardware thresholding (40 mV), while the subsequent are much more significant (500 mV and 1 V): by comparing the yellow dots, that is, the starting *QBER* values obtained by only imposing an *hardware* threshold, we can clearly see that the *QBER* benefits from that, as the *SNR* grows if we impose a selective acquirement of packets. Selection means that we acquire a lower number of packets (when the probe is above the hardware threshold), but the quality of the samples is enhanced; thus, the *QBER* value slows down. Nevertheless, we remark that an exaggerated hardware-threshold value causes the statistics for the current acquisition to be not significant (as we do not acquire enough samples to consider them for a reliable statistics).

2. Another impressive conclusion can be deduced by observing the impact of *software thresholding*: we shall compare yellow dot with blue dot for each aquisition, where the former specifies the *QBER* value without software thresholding (eventually with an hardware one), while the latter define the *QBER* value after a suitable optimal software thresholding, where the optimal threshold is computed in correspondence to the maximum value of the secret key length. We can clearly see that the **software threshold approach**, in the same way as the hardware case, allows to obtain a substantial improvement in the performances of the system. A particular focus is needed to enlighten that, in cases of particularly high background and noise conditions, for which the initial *QBER* values were above the maximum amount allowing a secret key exchange ($QBER = 11\%$), the threshold imposition permits to decrease the *QBER* value of transmission under the 11%, thus allowing to share a secret key even in conditions for which it was otherwise impossible.

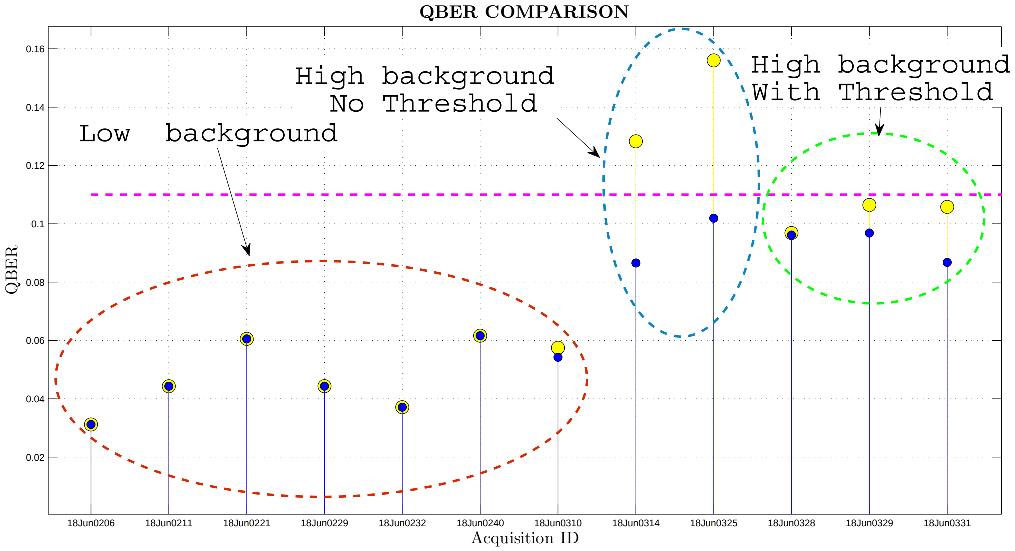


Figure 6.11: QBER comparison for acquisitions with high background conditions.

6.7 Conclusions

This work was realized with the aim to evaluate the possibility of implementing concretely the **exploiting turbulence** approach of Capraro et al. [29]: we know that atmospherical turbulence is one of the main problems in long-distance optical transmissions, and that it may limit the feasibility of secure key exchange between distant terminals, such as satellites. The amazing result is that we effectively achieve various benefits, not only in the **average amount of sifted bits per packet**, which grows as a validation of the fact that through packets selection we improve the quality of our acquisitions, but also in the **QBER value**, which decreases as a further confirmation of a *SNR* enhancement. Furthermore, for some trasmissions for which the *QBER* value was so high to inhibit key exchange, we shown that -thanks to a suitable threshold imposition- it is potentially possible to complete successfully a **secret key exchange** even in loath conditions, in which otherwise it would not be possible to accomplish a secure communication. Additionally, we have seen that in some cases we have a secure key length growth. As a consequence, the quality of the transmission is enhanced, allowing us to think of exploiting this approach for the benefit of long-distance transmissions and satellite communications. These results -and the data processing allowing them- concern the upper bound to expected key rates: in the follow up work, the analysis will be extended to the finite key regime, taking into account the need to estimate the channel and its threshold dependent QBER in a real implementation. At the present stage, the *turbulence exploiting approach* is a promising solution that could experience a considerable development and that embodies the potential to influence the future approach to long-distance communications techniques.

Bibliography

[1] R. M. Gray, “Entropy and Information Theory”, *Springer Verlag*, 1990.

[2] D. B. Drajic, “60 Years of Shannon Information Theory”, *IEEE Trans. on Commun.*, September 2007.

[3] M. Wilde, “Classical Shannon Theory”, 2012.

[4] G. Van assche, “Quantum Cryptography and Secret-Key Distillation”, *Cambridge University Press*, 2006.

[5] C. E. Shannon, “Communication Theory of Secrecy Systems”, *Bell System Technical Journal*, 28:4, pp 656-715, October 1949.

[6] N. Gisin, “Quantum Cryptography on noisy channels: quantum versus classical key-agreement protocols”, *Phys. Rev. Lett. 83*, February 1999.

[7] A. N. Shirazi, “Quantum Superposition, Mass and General Relativity”, April 2011.

[8] C. H. Bennett, “Quantum Information Theory”, *IEEE Trans. on Commun.*, vol. 44, n. 6, October 1998.

[9] G. Brassard, “Quantum Information Theory”, 1998.

[10] M. Wilde, “Quantum Information and Entropy”, 2012.

[11] G. Cariolaro, “Comunicazioni Quantistiche”, April 2009.

[12] W. K. Wootters, “The no-cloning Theorem”, *Physics Today*, February 2009.

[13] P. Thorsteinson, “Asymmetric Cryptography”, July 2009.

[14] R. Rivest, “Unconditionally Secure Authentication”, Lecture 3 : September 1997.

[15] D. R. Stinson, “Universal Hashing and Authentication Codes”, *Designs, Codes and Cryptography archive*, Volume 4 Issue 4, October 1994.

[16] S. B. Wilson, “Key agreement protocols and their security analysis”, *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, Pages 30-45, September 1997.

[17] M. Maurer, “Secret Key Agreement by Public Discussion for common information”, *IEEE Trans. on Commun.*, vol.39 n.3, May 1993.

[18] S. Wolf, “Unconditionally security in Cryptography”, *Lecture Notes in Computer Science*, vol. 1561, pp 217-250, 1999.

[19] Bennett, Brassard, “Generalized privacy amplification” , *IEEE Trans. on Commun.*, vol.41, November 1995.

[20] Bennett, Brassard, “Experimental Quantum Cryptography”, *Journal of Cryptology*, vol. 5, pp 3-28, 1992.

[21] Bennett, Brassard, “Quantum Cryptography, Public Key Distribution and coin tossing”, *IEEE Trans. on Commun.*, 1984.

[22] “swissquantum”, *IDquantique* Site.

[23] T. Calver, “An empirical analysis of the cascade secret key reconciliation protocol for quantum key distribution”, September 2011.

[24] Bennett, Brassard, “Privacy amplification by public discussion”,*SIAM Journal on Computing - Special issue on cryptography archive*, vol. 17, April 1988.

[25] Bennett, “Quantum Cryptography using any two non-orthogonal states”, *Phys. Rev. Lett.*, vol. 68, n.21, May 1992.

[26] M. Javed, “A survey of the prominent Quantum Key Distribution protocols”, *Proceedings of the 7th International Conference on Frontiers of Information Technology*, n. 39, 2009.

[27] M. Dusek, “Unambiguous discrimination of linearly independent states as an eavesdropping strategy”, 2002.

[28] I. Ordavo, “Free-space Quantum Cryptography”, June 2006.

[29] I. Capraro, A. Tomaello, A. Dall’Arche, F. Gerlin, R. Ursin, G. Vallone, P. Villoresi, “Impact of Turbulence in long range quantum and classical communications”, *Phys. Rev. Lett. 109, 200502*, July 2012.

[30] W. Milonni, H. Carter, “Effects of propagation through atmospheric turbulence on photon statistics”, *J. Opt. B: Quantum Semiclass. Opt. 6 S742*, 2004.

[31] M. Canale, D. Bacco, S. Calimani, F. Renna, N. Laurenti, G. Vallone, P. Villoresi, “A prototype of a free-space QKD scheme based on the B92 protocol”, *ISABEL ’11 Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, n. 186, 2011.

Ringraziamenti

Desidero innanzitutto ringraziare il Professor Nicola Laurenti, ed ugualmente il Dott. Matteo Canale, per la disponibilità, la gentilezza, l'impegno (e la pazienza) dimostrati nell'accompagnarmi in questi mesi di lavoro.

Questo traguardo, questa gioia, voglio dedicarli ai capisaldi della mia vita, a chi mi ha cresciuto e in tutto questo tempo mi ha sostenuto, a chi mi è stato vicino e con pazienza ha sopportato i miei momenti difficili, a chi mi ha dato la possibilità di costruire il mio futuro passo dopo passo. A chi è arrivato tanti anni fa e continua a rimanere nella mia vita, a chi è maturato, ha gioito, ha pianto insieme a me.. a chi ha trovato il modo giusto per spronarmi, a chi ha imparato a conoscermi, a chi mi ha capito ed è stato forte al posto mio quando ne avevo più bisogno, a chi mi è stato ad ascoltare, a chi mi vuole bene con semplicità.. a chi continua ad esserci.

Perchè la soddisfazione personale è poca cosa se non è condivisa con chi è capace di dare un senso in più alle gioie che viviamo.