Imperial College London

QFFF DISSERTATION

IMPERIAL COLLEGE LONDON

DEPARTMENT OF PHYSICS

Quantum Cryptography

Author:
Yoann PIÉTRI

Supervisor:
Jonathan HALLIWELL

September 25, 2020

Submitted in partial fulfillment of the requirements for the degree of Master of Science of
Imperial College London.

Abstract

In this paper, we review how the laws of Quantum Mechanics allow creating unconditionally secure protocols in cryptography, i.e. protocols where the security is ensured by physical laws and bounds and not on some computationally hard problems. We first review the use of Quantum Mechanics to perform unconditionally secure secret key distribution, and we then extend the ideas to other cryptographic tasks including public-key cryptography, digital signatures, and fingerprinting.

Contents

Chapter 1

Introduction

1.1 The beginnings of cryptography

Cryptography is the science, at the crossroads of mathematics, physics, and computer science, that tends to design protocols to prevent malicious third-party from reading private messages. Even if the development of computers during the 20th century made the research in cryptography explode, the use of cryptographic methods was common before. It is believed that Julius CAESAR used an encryption method, today known as *Caesar Cipher* or *Alphabet Shift Cipher*[1]. The principle is very simple: imagine you want to encode the 26 letters of the Roman alphabet (A to Z), you then assign each letter a number (A is 0, B is 1, ..., Z is 25). You then choose a secret key which is a non-zero integer. The encrypted message is composed of letters, the code of each letter being given by the modulo 26 addition between the code of the original letter and the secret key. Basically, the alphabet is shifted by a constant:
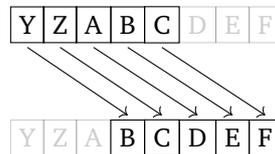
Figure 1.1: Caesar Cipher with $K = 3$

If we note n the position of the char in the string, U_n the unencrypted char, E_n the encrypted char, and K the shift:

$$E_n = U_n + K \bmod[26] \tag{1.1}$$

For instance, the secret message

IMPERIAL COLLEGE LONDON

with the secret key 13, would become

VZCREVNY PBYYRTR YBAQBA

since

n	Char	U_n	E_n	Char
1	I	8	21	V
2	M	12	25	Z
3	P	15	2	C
⋮	⋮	⋮	⋮	⋮

This code is very simple to break, as you can easily find patterns. For instance, you can compare the frequencies of each letter of the alphabet. In the English language, the most common letter is E with a frequency of 12.7%, followed by the letter T with a frequency of 9.06% [2]. You then compute the frequencies of the encrypted message. The most common letter in the encrypted text will probably be the encrypted char corresponding to E, or T. This method can be applied to all encryption schemes were a letter is always transformed to the same letter.

For this scheme, it also just possible to test the 25 possible secret keys and stop when you have a meaningful message.

Fortunately, cryptology has evolved since them. Even before the invention of the computer, the Germans used a very secure scheme (at that time) to encrypt message during the Second World War. It was secure in two ways:

- the setup has high combinatorial complexity. Germans have to choose 3 rotors in 5 possibilities, choose each initial positions for the rotors (26 possibilities each), and then choose 10 pairs in the plugboard, resulting in the combinatorial complexity

$$5 \times 4 \times 3 \times 26^3 \times \frac{26!}{6! \times 10! \times 2^{10}} = 158,962,555,217,826,360,000 \tag{1.2}$$

This is not possible to test all possibilities for a human, nor at that time for a machine.

- Unlike the *Caesar Cipher*, a letter would not become the same letter every time, as the rotors were moving at each letter. This is important because pattern methods (to compare frequencies of the language with the ones in the encrypted text) cannot be used.

The Enigma machine was broken by a team mainly lead by Alan Turing using a machine called Bombe and some flaws of the Enigma machine. In the end, the Enigma configuration could be found in 20 minutes.

1.2 Modern Cryptography

With the arrival of computers and communication networks came also a greater need for cryptography.

In our modern world, cryptography is ubiquitous. When you load a web page in a browser with a small padlock next to the URL, you are using symmetric and asymmetric encryption without knowing it. When you open some messaging app, like WhatsApp, you are using, again without knowing it[1] symmetric and asymmetric encryption, providing end-to-end encryption [3].

Symmetric and asymmetric encryption are different in their essence. A symmetric encryption scheme uses a secret key to encrypt and to decrypt the message. This means that the two (or more) parties that are communicating need to have the same, and secret, key. Symmetric schemes have several issues:

- The key exchange must be truly secret to ensure the privacy of the encrypted message;

- If you use the key to encrypt more than one time, information can be extracted from the encrypted message;

- The protocol doesn't scale well (i.e. if you want to have encrypted discussion between several users, you have to share the secret key between all the users, which mean, more chance of information leakage).

They have also some advantages: there are easier to understand and to implement and they need less computational power than asymmetric encryption.

In contrast, an asymmetric encryption scheme or public-key encryption scheme is a scheme where two keys are used:

---

[1]I mean without the user making anything for the cryptographic exchange to happen. It is totally transparent for the user.

- The *public key* is used only to encrypt messages. It can be distributed without risk to anyone who wants to send an encrypted message to the owner of the *private key*. The public key can be thought of as an open padlock. Once closed only the private key can unlock it.

- The *secret key* is used only to decrypt messages (and to generate the public key). It should be kept secret. The secret key can be seen as a key that opens every padlock closed by the public key.

One of the most used asymmetric encryption scheme today is the RSA (named after the names of its inventors: RIVEST, SHAMIR, and ADLEMAN who designed in the '70s [4]). It is based on some mathematical observations. Let's recall some arithmetical properties:

- A prime number is a whole number with exactly 2 distinct divisors. Every integer greater or equal than 2, that is not a prime number is called a composite number.

- Every whole number can be uniquely written as a product of prime numbers:

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k} \tag{1.3}$$

- A product of two prime numbers is called a semi-prime number. It has 3 or 4 distinct divisors (3 if it is a square of a prime number).

The basic idea of the RSA scheme follows: it easy to multiply 2 prime numbers to obtain a semi-prime number but it's difficult to factorize a semi-prime number into 2 prime numbers.

$$\begin{aligned}
3 \times 7 &=? & &\text{Easy!} \\
11 \times 13 &=? & &\text{A little less easy but still ok} \\
187 &=?\times? & &\text{The alert reader may get it} \\
748747 &=?\times? & &\text{This gets nasty}
\end{aligned} \tag{1.4}$$

In practice, for a 2048 bits long RSA key, a malicious party would have to factorize something like:

10016444466812877516651347592092877606999325867156134902126474870576401310371509197937849497
32061828879847934816861684862864326449214280155473757303841537670351486905855745788953294686653
05667852687855685298115910404311303180404287100354588108313006482467735715047743256036128648480
80027194762485965856140145864228400743999303156570382089086775865731055296724143521221327468628
21950171266360637073763193766827057457206146627252158883606266393926431447227342695628623860494
83076188549980295606990827731968687429507788792780286440882172770001367957911700000685949637652
38831914470509293382332669418868301436781248853885000370663778352581253239270257156871660127150
01725765933851378635689651151763527144099274447723857372797474452663650725422387256011846500895
56049862683135640206298862612679119720709968586034215160997260304220155673434151135668320749865
84807932093124539029156912634836160456728007753201898072897827815590459999295298908223557195231
58977763724441639028178539046224952247530731887239092769161189850803594847326119864462181341673
60716012369946975020768242661592323585459972285070236101616423672439653172724479999925676798119
71560093919447685551083829047142039685301977153590924844326332056772159786693521935447299870583
1 [2]

A human being cannot factorise this semi-prime number, nor can a machine at the time this paper is written. In 1991, the RSA laboratories published 50 semi-prime numbers (called the RSA numbers) from 100 to 2048 decimal digits. For some numbers, a reward was offered for the factorisation of the number. Some easy numbers were factorised quite quickly (the RSA-100 was factorised less than 2 weeks after the publication of the challenge) and some of them are still being factorised today (the example of the RSA-250 that was factorised in February 2020 is relevant) even if the challenge ended in 2007. The biggest RSA number to have ever been factorized is the RSA 768, that was factorized in 2009 using the General Number Field Sieve (GNFS).

The GNFS is the best-known algorithm for factorising integer (at least for large integers). The GNFS has an algorithmic complexity of

---

[2]This is the semi-prime number of my personal RSA key, which is in the public part of the key.

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln(n))^{\frac{1}{3}}(\ln(\ln(n)))^{\frac{2}{3}}\right) \tag{1.5}$$

where n is the integer to factorise. As we won't discuss much more of how the GNFS works, the interested reader may refer to [5] for a discussion on the algorithm and the complexity. The GNFS is one the best algorithm to factorise integer, but, nevertheless, it would take more than a billion years to factorise a 2048 digit key. In this sense, the RSA is a secure protocol.

I would like to emphasise here that the security of the RSA scheme is based on a computationally hard mathematical problem and the belief that we won't find another more efficient and much quicker algorithm to solve the problem. But we are believing in the security of the RSA because it has survived 50 years of attacks.

Symmetric and asymmetric schemes often work together in what we call hybrid cryptosystems. This idea is further developed in chapter 4.

1.3 Quantum computer or the end of cryptography?

In the early '80s, the idea of a quantum computer began to grow and with it, the hope to compute and simulate things that were not possible or will ever be possible on a classical computer [6, 7]. In particular, FEYNMAN proposed to simulate Physics with a quantum computer, that he didn't assimilate to the quantum analog of a Turing machine but rather, to a universal quantum simulator. BENIOFF, in his paper, explored the possibility of a quantum Turing machine, or universal quantum computer. Without speaking more about the idea of a quantum Turing machine, we will see that we are today far from having such a computer.

One of the main advantages of quantum information is the superposition principle. The qubit, the analog of the classical bit, can take the value $|0\rangle$ (analog of the bit 0), the value $|1\rangle$ (analog of the bit 1), or a superposition of the two, meaning that with one qubit, we can store much more information than with one bit. We saw some vulgarisation programs that compared the classical bit to a switch. Then a qubit was compared to a switch that could be open, closed, and both at the same time. But this doesn't feel like a good comparison, and they often don't speak about the measurement.

The information encoded in the superposition of state, if not used correctly, will be destroyed when a measurement on the qubit is made. Here is a graphical representation of the difference between a bit and a qubit:
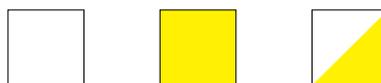
Figure 1.2: Graphical representation of qubits

The first one is $|0\rangle$ and the second one is $|1\rangle$. The third qubit is an equitable mixture of $|0\rangle$ and $|1\rangle$. We could envisage all mixtures of $|0\rangle$ and $|1\rangle$. Also, we said that measurement is an operation where we loose information. Here the measurement operation could be modeled as the question: "what is the color of the square ?". For both the first and second qubits, the answer is obvious ("white" for the first one and "yellow" for the second one). For the last one, regardless of the answer, information will be lost, and for this case, the answer would be "white" half of the time and "yellow" the other half.

There was also the hope of a quantum speedup, i.e. to compute or simulate things faster on a quantum computer than on a classical computer one even if the computation or simulation was possible in the first place.

One of the simplest examples is GROVER's algorithm [8]. The basic idea of GROVER's algorithm is to find an element in a list. Supposing a list of length N, using a classical algorithm, the best case is to find the element in the first position, the worst case, in the Nth position, and on average,

you need $\frac{N}{2}$ operations, which means a complexity in $O(N)$. Using a technique called amplitude amplification, GROVER's algorithm has a complexity of $O(\sqrt{N})$.

Another promising algorithm is SHOR's algorithm. It was introduced in 1997 by Peter SHOR [9]. The purpose of SHOR's algorithm is to factorize integers and has a complexity of

$$(\ln(n))^2(\ln(\ln(n)))(\ln(\ln(\ln(n)))) \tag{1.6}$$

This algorithm is much faster than the most efficient classical algorithm: the GNFS, as the figure below shows:

(a) Comparison of GNFS and Shor's algorithm for different frequencies

(b) Comparison of GNFS and Shor's algorithm for different frequencies and architectures
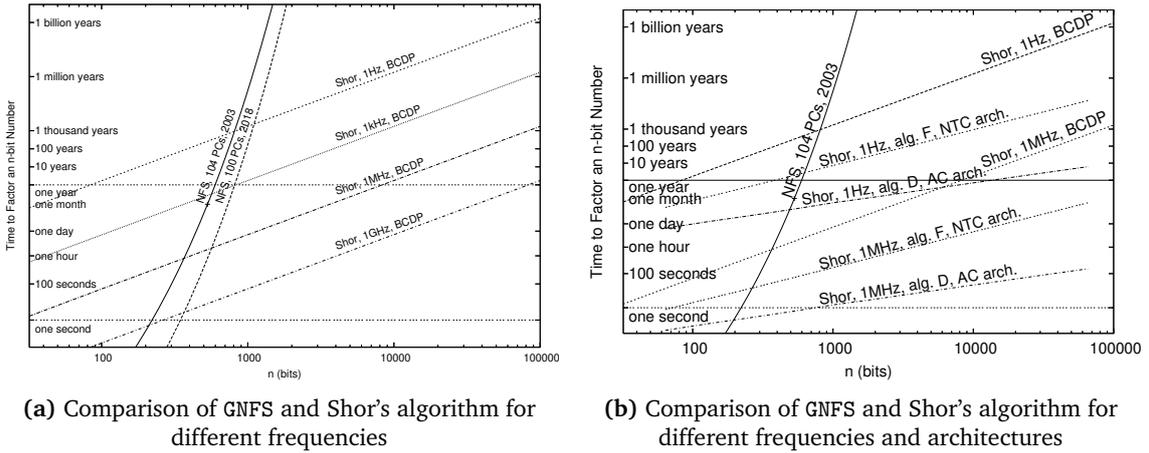
Figure 1.3: Comparison of GNFS and Shor's algorithm

(it was proved by VAN METER *et al.* in 2008 that the execution depends on the architecture of the quantum computer and the frequency [10]. Figures are from this article).

We won't explain in detail how SHOR's algorithm works, but it's based on a classical algorithm with a quantum subroutine, that finds a period of a function, using Quantum Fourier Transform (see [9] for more detail).

Today, we mostly use the RSA cryptosystem with a key of length 2048. The GNFS would need more than a billion years to factorise the key, but using SHOR's algorithm at 1MHz, it would take between tens of seconds and less than a month depending on the architecture, which is a huge speedup.

Then, we might break the RSA in the foreseeable future. There are however several issues. Although Google claimed to have reached Quantum Supremacy [11], i.e. they claimed to have computed something that is not computable on classical computers (they estimated at 10 000 years the amount of time needed on the most powerful classical computers to compute what they did in 200 seconds), they have done a task that is quite resilient to errors. SHOR's algorithm is less error-resilient and the error rate that we have in quantum computers today is too high for factorising large integers. Today, the largest integer factorised by SHOR's algorithm is 21 [12].

Nevertheless, we might see the RSA broken in the near future. What must be understood is that any encrypted message saved by a malicious party could be decrypted as soon as RSA is broken. This means that any private message sent today can become a public message later. To prevent those issues, some governments and companies are working in an area called post-Quantum Cryptography.

1.4 Post-Quantum Cryptography

In 2016, the NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST) of the US Department of Commerce issued a report on post-Quantum Cryptography [13]. It first described what is the core of modern cryptography: public-key encryption, digital signatures, and key exchange. Some

other components are required and we will talk of them later. Then the report continues by describing how a potential quantum computer would be capable of breaking or at least weakening some of those algorithms. The RSA, as we saw, would become insecure and the AES (Advanced Encryption Standard) which is a widely used symmetric scheme to encrypt data would require larger keys (GROVER's algorithm can help breaking symmetric encryption).

The issue is the fact that those algorithms use some kind of mathematical problem that is hard to solve on a classical computer (factorising whole numbers or the discrete log problem) and that we could experience a quantum speedup to solve those problems.

Then there are two approaches to solve this problem:

- find mathematical problems that are impossible to solve in a reasonable amount of time even with a quantum computer;

- use physical constraints, using for example quantum mechanics to design unconditionally secure protocols.

The first point is more on the classical side of Quantum Cryptography and that is usually what we mean when we talk about post-Quantum Cryptography. Some problems are already good candidates for such algorithms and some of them are already implemented. Indeed, even if SHOR's algorithm can break cryptosystems based on large integer factorisation or discrete logarithms, it can't be applied to some other cryptosystems like [14]:

- Hashed-based cryptography (for example MERKLE's hash tree)

- Code-based cryptography (MCELIECE's hidden-Goppa-Code)

- Lattice-based cryptography (NTRU)

- Multivariate-quadratic-equations cryptography

- Symmetric cryptography (recent symmetric cryptography schemes are considered pretty secure against quantum attacks)

GROVER's algorithm can provide help to break some of those schemes, but as the algorithm has a smaller speedup, it is sufficient to use a larger key [14]. But we may one day find a quantum algorithm that can help breaking any of these cryptosystems (even if these schemes are believed to be quantum-safe and selected by the NIST). Therefore, we consider solutions from point two, where some schemes are proven to be resistant, in principle, to an attacker with unlimited quantum power.

We will look forward to the second point for the rest of this paper, i.e. we will see how it is possible to implement key exchange (see chapter 3), public cryptography (public key, digital signatures, fingerprinting) (see chapter 4) using quantum information theory. In chapter 5, we investigate other components useful in cryptography that can be improved using quantum information theory.

The role of the NIST is to standardise protocols. In this purpose, they launched in 2016 a request for public candidatures: researchers and computer scientists would submit a potential quantum-safe algorithm to the NIST. On July 22, 2020, the third round of this request ended, and four algorithms are finalists for public-key encryption and key-establishment and three are finalists for digital signatures (there are also some alternative candidates). Those schemes use methods cited above (we find the NTRU and MecEliece schemes for instance). They make pick one or some algorithms for the standardisation process that will come during the next few years. In the report, NIST estimated that a quantum computer capable of break the RSA2048 could be available in 2030 and they advised the companies to make the change during the second half of the 2020's decade. The standards are then expected for around 2022.

Chapter 2

Preliminaries

This chapter aims to derive key results in Quantum Cryptography that will be used in the following chapters. It defines some notations and recalls some basic rules of Quantum Mechanics.

2.1 Notations

We are setting the work of Quantum Cryptography and Quantum Computing in a two-dimensional Hilbert space $\mathcal{H}$. The vectors in $\mathcal{H}$ are quantum states and named qubits (for quantum bits) and denoted with the bracket notations: $|\phi\rangle \in \mathcal{H}$.

We denote by $\boldsymbol{\sigma_x}, \boldsymbol{\sigma_y}$ and $\boldsymbol{\sigma_z}$ the Pauli operators (the operators will be in bold). The eigenstates of $\boldsymbol{\sigma_z}$ are denoted $|0\rangle$ and $|1\rangle$

$$\boldsymbol{\sigma_z} |0\rangle = |0\rangle \quad \boldsymbol{\sigma_z} |1\rangle = - |1\rangle \tag{2.1}$$

$|0\rangle$ and $|1\rangle$ are orthogonal states ($\langle 0|1\rangle = 0$) and form an orthonormal basis of $\mathcal{H}$. The basis is denoted $\mathcal{B}_z = \{|0\rangle, |1\rangle\}$. In this basis, the Pauli operators read

$$\boldsymbol{\sigma_x} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \boldsymbol{\sigma_y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \boldsymbol{\sigma_z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.2}$$

We denote by $|\pm\rangle$ the states

$$|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}} \tag{2.3}$$

They are the eigenstates of the $\boldsymbol{\sigma_x}$ operator:

$$\boldsymbol{\sigma_x} |+\rangle = |+\rangle \quad \boldsymbol{\sigma_x} |-\rangle = - |-\rangle \tag{2.4}$$

They are orthogonal states and form a basis, that we denote by $\mathcal{B}_x = \{|+\rangle, |-\rangle\}$.

If we consider two Hilbert spaces $\mathcal{H}_a$ and $\mathcal{H}_b$, their tensor product is denoted $\mathcal{H}_a \otimes \mathcal{H}_b$ and is composed of the states

$$|\psi\rangle_a \otimes |\phi\rangle_b = |\psi\rangle_a |\phi\rangle_b = |\psi\rangle |\phi\rangle = |\psi\rangle \otimes |\phi\rangle = |\psi\phi\rangle \in \mathcal{H}_a \otimes \mathcal{H}_b \tag{2.5}$$

for $|\psi\rangle_a \in \mathcal{H}_a$, $|\phi\rangle_b \in \mathcal{H}_b$.

We end this section by giving some mathematical notations:

- $\oplus_2$ is the modulo-2 addition;

- $\lfloor \cdot \rfloor$ is the floor function;

- $\mathbb{P}(A)$ denotes the probability of an event A.

2.2 Quantum gates

In classical computers, we have, at the heart of computers, gates, that are implemented with transistors. Those gates take one or more bits as entry and output one or more bits. For example, the NOT gate is a gate that inverts the bit (a 0 becomes a 1 and a 1 becomes a 0)
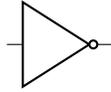
Figure 2.1: Representation of the NOT gate

Input	0	1
Output	1	0

Table 2.1: Logic table of the NOT gate

Quantum operators are unitary operators applied on one or more qubits (a unitary operator on $\mathcal{H}$ is a on one-qubit gate, on $\mathcal{H} \otimes \mathcal{H}$ is a two-qubit gate, etc...).

For example, one could consider the quantum analog of the NOT gate, i.e. a gate U such that, for all $|\phi\rangle$,

$$U |\phi\rangle = \psi \quad \text{and} \quad \langle\psi|\phi\rangle = 0 \tag{2.6}$$

This is not possible since such an operator would impose

$$\begin{aligned} U |0\rangle &= |1\rangle \\ U |1\rangle &= |0\rangle \end{aligned} \tag{2.7}$$

and then, we would have

$$U |+\rangle = U \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{|1\rangle + |0\rangle}{\sqrt{2}} = |+\rangle \tag{2.8}$$

Nevertheless, what we call the quantum NOT gate is the σ_x gate since

$$\sigma_x |0\rangle = |1\rangle \quad \text{and} \quad \sigma_x |1\rangle = |0\rangle \tag{2.9}$$

The equivalent of the logic table of a logic gate would be the matrix representation in a canonical basis ($\{|0\rangle, |1\rangle\}$ for $\mathcal{H}$, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ for $\mathcal{H} \otimes \mathcal{H}$, etc...). For instance, for the σ_z gate, we would have

Figure 2.2: Representation of the σ_x gate

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.10}$$

Matrix of the σ_x gate in the $\mathcal{B}_z$ basis

A quantum gate related to the σ_x gate, but this time is a two-qubit gate, is the controlled-NOT or **CNOT** gate. This is a gate where σ_x is applied on the second qubit if the first qubit is $|1\rangle$.

$$\begin{aligned} \textbf{CNOT} |0\rangle |0\rangle &= |0\rangle |0\rangle \\ \textbf{CNOT} |0\rangle |1\rangle &= |0\rangle |1\rangle \\ \textbf{CNOT} |1\rangle |0\rangle &= |1\rangle |1\rangle \\ \textbf{CNOT} |1\rangle |1\rangle &= |1\rangle |0\rangle \end{aligned} \tag{2.11}$$

The first qubit is called the control qubit and the second one is called the target qubit.

If we have a superposition of $|0\rangle$ and $|1\rangle$ for the control qubit then

$$\mathbf{CNOT}(\alpha\,|0\rangle + \beta\,|1\rangle)\,|0\rangle = \alpha\mathbf{CNOT}\,|0\rangle\,|0\rangle + \beta\mathbf{CNOT}\,|1\rangle\,|0\rangle = \alpha\,|0\rangle\,|0\rangle + \beta\,|1\rangle\,|1\rangle \tag{2.12}$$

Control qubit

Target qubit
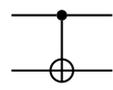
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \left(\begin{array}{c|c} \mathbb{1}_2 & \mathbf{0_2} \\ \hline \mathbf{0_2} & \sigma_x \end{array} \right) \tag{2.13}$$

Figure 2.3: Representation of the **CNOT** gate

Matrix of the **CNOT** gate in the canonical basis

An important gate is the Hadamard gate H, which is the transfer matrix between the $\mathcal{B}_z$ and $\mathcal{B}_x$ bases:

$$\begin{aligned} H\,|0\rangle &= |+\rangle \\ H\,|1\rangle &= |-\rangle \\ H\,|+\rangle &= |0\rangle \\ H\,|-\rangle &= |1\rangle \end{aligned} \tag{2.14}$$

In quantum circuits, we represent it by
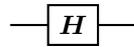
$$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.15}$$

Figure 2.4: Representation of the Hadamard gate

Matrix of the Hadamard gate in the $\mathcal{B}_z$ basis

It has its importance when we will speak about measurement in quantum circuits.

Finally, the **SWAP** gate is a gate that takes two qubits as input and outputs two qubits, such that, for all $|\phi\rangle$ and $|\psi\rangle$,

$$\mathbf{SWAP}(|\phi\rangle\,|\psi\rangle) = |\psi\rangle\,|\phi\rangle \tag{2.16}$$

i.e. the two qubits are swapped.

It can be represented by

$$\mathbf{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.17}$$

in the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

In quantum circuits, we represent it by
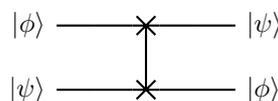
Figure 2.5: Representation of the SWAP gate

2.3 Measurement

Let $\mathcal{B} = \{|b_i\rangle\}_i$ be an orthonormal basis of some Hilbert space $\mathcal{H}$. Then a general state $|\phi\rangle \in \mathcal{H}$ can be written

$$|\phi\rangle = \sum_i \alpha_i |b_i\rangle \tag{2.18}$$

with

$$\sum_i |\alpha_i|^2 = 1 \tag{2.19}$$

Then, a measurement of $|\phi\rangle$ in the $\mathcal{B}$ basis yields result $|b_i\rangle$ with probability $|\alpha_i|^2$. After measurement, the state collapse to $|b_i\rangle$ (measurement alters the system).

In quantum circuits, the measurement in the $\mathcal{B}_z$ basis is represented by
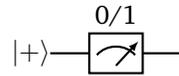
Using the properties of the Hadamard gate, a measurement in the $\mathcal{B}_x$ basis is performed as follows:
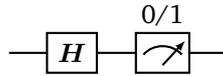
2.4 No-cloning theorem

A central result in quantum computing and Quantum Cryptography is the no-cloning theorem [15]:

> **Theorem (No-cloning theorem):** Let $\mathcal{H}_a$ and $\mathcal{H}_b$ be two Hilbert spaces. There is no unitary operator $\boldsymbol{U}$ acting on $\mathcal{H}_a \times \mathcal{H}_b$ such that, for every $|\Psi\rangle_a$,
>
> $$\boldsymbol{U}(|\Psi\rangle_a |0\rangle_b) = |\Psi\rangle_a |\Psi\rangle_b \tag{2.20}$$
>
> It basically means that we cannot clone an arbitrary unknown quantum state.

Proof. The proof is straightforward and is based on *Reductio ad absurdum*. Let's suppose we have such a unitary operator. We then have

$$\begin{aligned} \boldsymbol{U} |0\rangle |0\rangle &= |0\rangle |0\rangle \\ \boldsymbol{U} |1\rangle |0\rangle &= |1\rangle |1\rangle \end{aligned} \tag{2.21}$$

Then let's consider an arbitrary state $\alpha |0\rangle + \beta |1\rangle$ with $\alpha, \beta \in \mathbb{C}$. Then,

$$\boldsymbol{U}(\alpha |0\rangle + \beta |1\rangle) |0\rangle = \alpha \boldsymbol{U} |0\rangle |0\rangle + \beta \boldsymbol{U} |1\rangle |0\rangle = \alpha |0\rangle |0\rangle + \beta |1\rangle |1\rangle \tag{2.22}$$

but we also have, from the definition of U that

$$\boldsymbol{U}(\alpha |0\rangle + \beta |1\rangle) |0\rangle = (\alpha |0\rangle + \beta |1\rangle)(\alpha |0\rangle + \beta |1\rangle) = \alpha^2 |0\rangle |0\rangle + \alpha\beta(|0\rangle |1\rangle + |1\rangle |0\rangle) + \beta^2 |1\rangle |1\rangle \tag{2.23}$$

The expressions of 2.22 and 2.23 are different unless $\alpha = 1$ or $\beta = 1$ and then, such a $\boldsymbol{U}$ cannot exist. $\qquad\square$

2.5 Conjugate coding

Conjugate coding is one of the most important notions of Quantum Cryptography. The term is attributed to WIESNER, who proposed the idea in an unpublished (at first) and unnoticed article written in 1970, but was published in 1983 after Quantum Cryptography became a more plausible idea [16]. It is also called *quantum coding* and *quantum multiplexing*. The basic idea is that we can encode our classical bits 0 or 1 in different bases, and that a measurement in one of the basis will completely randomise the result in the other.

Consider for example the bases $\mathcal{B}_z$ and $\mathcal{B}_x$ defined by

$$
\begin{aligned}
\mathcal{B}_z &= \{|0\rangle, |1\rangle\} \\
\mathcal{B}_x &= \{|+\rangle, |-\rangle\} \\
|\pm\rangle &= \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}
\end{aligned}
\tag{2.24}
$$

We encore our classical bits using the following table

	$\mathcal{B}_z$	$\mathcal{B}_x$
0	$|0\rangle$	$|+\rangle$
1	$|1\rangle$	$|-\rangle$

Table 2.2: Equivalence bits and qubits

Now imagine you want to encode the bit 1. If you do so in the $\mathcal{B}_z$ basis, you will have the state $|1\rangle$. Now if you measure this state in the $\mathcal{B}_z$ basis, you will always obtain $|1\rangle$ and recover the good bit as

$$
\begin{aligned}
|\langle 0|1\rangle|^2 &= 0 \\
|\langle 1|1\rangle|^2 &= 1
\end{aligned}
\tag{2.25}
$$

But, if you make a measurement in the $\mathcal{B}_x$ basis, you will end up with $|+\rangle$ (and recover the bit 0, so the bad one) half of the time and $|-\rangle$ (and recover the good bit) the other half since

$$
\begin{aligned}
|\langle +|1\rangle|^2 &= \left|\frac{\langle 0| + \langle 1|}{\sqrt{2}}|1\rangle\right|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2} \\
|\langle -|1\rangle|^2 &= \left|\frac{\langle 0| - \langle 1|}{\sqrt{2}}|1\rangle\right|^2 = \left|-\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}
\end{aligned}
\tag{2.26}
$$

We say that $\mathcal{B}_z$ and $\mathcal{B}_x$ are *conjugate* to each other.

Note: in Quantum Cryptography, we are often working with light and polarization of light to make our qubits. In this case, the bases are called rectilinear (R) and diagonal (D) and are defined by

$$
\begin{aligned}
R &= \{|\leftrightarrow\rangle, |\updownarrow\rangle\} \\
D &= \{|\nwarrow\!\!\searrow\rangle, |\nearrow\rangle\}
\end{aligned}
\tag{2.27}
$$

associating

$$
\begin{aligned}
|\leftrightarrow\rangle &= |0\rangle \\
|\updownarrow\rangle &= |1\rangle \\
|\nwarrow\!\!\searrow\rangle &= |+\rangle \\
|\nearrow\rangle &= |-\rangle
\end{aligned}
\tag{2.28}
$$

2.6 Entanglement, EPR paradox and BELL inequalities

In 1935, EINSTEIN, PODOLSKY, and ROSEN proposed a thought experiment to prove that the theory of Quantum Mechanics was incomplete. They proposed the notion of hidden variables to solve the issue of particles that would be correlated after being separated [17].

Shortly after SCHRÖDINGER published a paper on entanglement [18], and it is described as "the characteristic trait of quantum mechanics". Entanglement describes a situation when two or more particles cannot be described on their own, i.e. the quantum state of the whole system cannot be factored into quantum states for each particle. The historical example is the singlet spin state: consider two spin-$\frac{1}{2}$ particles, that are paired in such a way that the overall spin is 0. Measuring the spin of one particle immediately gives the spin of the other as it is perfectly opposite.

In 1964, BELL showed that considering local realism, some quantity will respect the so-called BELL inequalities, and if the BELL inequalities were to be violated, then the hypothesis of 1935 of local realism would be rejected. We will see an example of application, and derivation, of BELL (or CHSH see [19]) inequality. The BELL test has been performed several times (first time in 1972 [20]) and results rejected hidden local variables.

Let's recall some definitions: a pure state, or separable state of $\mathcal{H}_a \otimes \mathcal{H}_b$ is a state that is not entangled and can be represented in the form

$$|\phi\rangle_a \otimes |\phi\rangle_b \tag{2.29}$$

A state that cannot be factored in such a way is called a mixed state, or entangled state. For instance the singlet spin state can be represented by

$$\frac{1}{\sqrt{2}} \left(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle \right) \tag{2.30}$$

which cannot be factored.

More formally, for a state $|\phi\rangle$, we can define the density operator (or density matrix)

$$\boldsymbol{\rho} = |\phi\rangle \langle\phi| \tag{2.31}$$

This is an Hermitian positive semi-definite operator, with unit trace.

A state is pure if and only if

$$\boldsymbol{\rho}^2 = \boldsymbol{\rho} \tag{2.32}$$

The four BELL states are defined by

$$
\begin{aligned}
\left|\Phi^+\right\rangle_{AB} &= \frac{1}{\sqrt{2}} \left(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B \right) \\
\left|\Psi^+\right\rangle_{AB} &= \frac{1}{\sqrt{2}} \left(|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B \right) \\
\left|\Psi^-\right\rangle_{AB} &= \frac{1}{\sqrt{2}} \left(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B \right) \\
\left|\Phi^-\right\rangle_{AB} &= \frac{1}{\sqrt{2}} \left(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B \right)
\end{aligned}
\tag{2.33}
$$

They are maximally entangled states.

In the following, an EPR source is a source that emits pair of entangled particles.

2.7 Quantum teleportation

Imagine that Alice has a qubit $|\phi\rangle$. She wants to teleport her qubit to Bob. This is possible using quantum teleportation, which was first proposed in 1993 [21]. This can be done as follows:

1. An entangled pair is created and shared between Alice and Bob;

2. Alice performs a BELL measurement of the entangled pair qubit and $|\phi\rangle$. This measurement has 4 outcomes possible, meaning that we can encode the result with 2 classical bits. After this measurement, Bob's entangled qubit is in the same state as $|\phi\rangle$ or in one of three other states, closely related to $|\phi\rangle$;

3. Alice sends the result of measurement encoded in two classical bits over the classical channel;

4. Depending on the result, Bob performs one of four actions to his qubit and end up with $|\phi\rangle$.

Note that after the BELL measurement, Alice's qubit is not in the state $|\phi\rangle$, and hence, this does not violate the no-cloning theorem.

Formally, let's suppose that Alice has the following target qubit

$$|\phi\rangle_T = \alpha\,|0\rangle_T + \beta\,|1\rangle_T \tag{2.34}$$

with α, β unknown (they will remain unknown by the end of the teleportation). The T stands for Target.

For the EPR pair, they are using one of the four maximally entangled BELL states. The BELL state that is used is decided beforehand and for the following, we suppose they share the BELL state

$$\left|\Phi^+\right\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \tag{2.35}$$

where the A subscript refers to Alice and the B one to Bob.

Then the total system state is

$$|\phi\rangle_T \otimes \left|\Phi^+\right\rangle_{AB} = \frac{1}{\sqrt{2}}(\alpha\,|0\rangle_T + \beta\,|1\rangle_T) \otimes (|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \tag{2.36}$$

Using the following relations, which are easily verifiable:

$$
\begin{aligned}
|0\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}\left(\left|\Phi^+\right\rangle + \left|\Phi^-\right\rangle\right) \\
|0\rangle \otimes |1\rangle &= \frac{1}{\sqrt{2}}\left(\left|\Psi^+\right\rangle + \left|\Psi^-\right\rangle\right) \\
|1\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}\left(\left|\Psi^+\right\rangle - \left|\Psi^-\right\rangle\right) \\
|1\rangle \otimes |1\rangle &= \frac{1}{\sqrt{2}}\left(\left|\Phi^+\right\rangle - \left|\Phi^-\right\rangle\right)
\end{aligned}
\tag{2.37}
$$

and some algebra:

$$
\begin{aligned}
|\phi\rangle_T \otimes \left|\Phi^+\right\rangle_{AB} &= \frac{1}{\sqrt{2}}(\alpha\,|0\rangle_T + \beta\,|1\rangle_T) \otimes (|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \\
&= \frac{\alpha}{2}\,|0\rangle_T\,|0\rangle_A\,|0\rangle_B + \frac{\beta}{2}\,|1\rangle_T\,|0\rangle_A\,|0\rangle_B + \frac{\alpha}{2}\,|0\rangle_T\,|1\rangle_A\,|1\rangle_B + \frac{\beta}{2}\,|1\rangle_T\,|1\rangle_A\,|1\rangle_B \\
&= \frac{\alpha}{2}\left(\left|\Phi^+\right\rangle_{TA} + \left|\Phi^-\right\rangle_{TA}\right)|0\rangle_B + \frac{\beta}{2}\left(\left|\Psi^+\right\rangle_{TA} - \left|\Psi^-\right\rangle_{TA}\right)|0\rangle_B \\
&\quad + \frac{\alpha}{2}\left(\left|\Psi^+\right\rangle_{TA} + \left|\Psi^-\right\rangle_{TA}\right)|1\rangle_B + \frac{\beta}{2}\left(\left|\Phi^+\right\rangle_{TA} - \left|\Phi^-\right\rangle_{TA}\right)|1\rangle_B
\end{aligned}
$$

We finally end up with

$$
\begin{aligned}
|\phi\rangle_T \otimes \left|\Phi^+\right\rangle_{AB} = \frac{1}{2}\Big(&\left|\Phi^+\right\rangle_{TA}(\alpha\,|0\rangle_B + \beta\,|1\rangle_B) \\
&+ \left|\Phi^-\right\rangle_{TA}(\alpha\,|0\rangle_B - \beta\,|1\rangle_B) \\
&+ \left|\Psi^+\right\rangle_{TA}(\alpha\,|1\rangle_B + \beta\,|0\rangle_B) \\
&+ \left|\Psi^+\right\rangle_{TA}(\alpha\,|1\rangle_B - \beta\,|0\rangle_B)\Big)
\end{aligned}
\tag{2.38}
$$

When Alice makes the BELL measurement, she will have one of the four results: $|\Phi^+\rangle_{TA}$, $|\Phi^-\rangle_{TA}$, $|\Psi^+\rangle_{TA}$, $|\Psi^-\rangle_{TA}$, that we can encode on two classical bits by:

Result	Encoding
$|\Phi^+\rangle$	00
$|\Phi^-\rangle$	01
$|\Psi^+\rangle$	10
$|\Psi^-\rangle$	11

Table 2.3: Encoding the BELL measurement result

Now, also depending on the measurement result, Bob will end up with slightly different states:

- If the result is $|\Phi^+\rangle_{TA}$, Bob's state is $\alpha |0\rangle + \beta |1\rangle = |\phi\rangle$. In this case, Bob does nothing;

- If the result is $|\Phi^-\rangle_{TA}$, Bob's state is $\alpha |0\rangle - \beta |1\rangle$ and he can apply the Z-gate (the $\boldsymbol{\sigma_z}$ Pauli matrix) since

$$\boldsymbol{\sigma_z} |0\rangle = |0\rangle$$
$$\boldsymbol{\sigma_z} |1\rangle = - |1\rangle \tag{2.39}$$
$$\boldsymbol{\sigma_z}(\alpha |0\rangle - \beta |1\rangle) = \alpha |0\rangle + \beta |1\rangle = |\phi\rangle$$

- If the result is $|\Psi^+\rangle_{TA}$, Bob's state is $\alpha |1\rangle + \beta |0\rangle$ and he can apply the X-gate (the $\boldsymbol{\sigma_x}$ Pauli matrix) since

$$\boldsymbol{\sigma_x} |0\rangle = |1\rangle$$
$$\boldsymbol{\sigma_x} |1\rangle = |0\rangle \tag{2.40}$$
$$\boldsymbol{\sigma_x}(\alpha |1\rangle + \beta |0\rangle) = \alpha |0\rangle + \beta |1\rangle = |\phi\rangle$$

- If the result is $|\Psi^-\rangle_{TA}$, Bob's state is $\alpha |1\rangle - \beta |0\rangle$ and he can apply the X-gate and Z-gate since

$$\boldsymbol{\sigma_z}\boldsymbol{\sigma_x}(\alpha |1\rangle - \beta |0\rangle) = \boldsymbol{\sigma_z}(\alpha |0\rangle - \beta |1\rangle) = \alpha |0\rangle + \beta |1\rangle = |\phi\rangle \tag{2.41}$$

Note that $\boldsymbol{\sigma_z}\boldsymbol{\sigma_x} = i\boldsymbol{\sigma_y}$, which is almost the Y-gate.

In every case, Bob ends up with the correct qubit and Alice's state has collapsed after the measurement so she doesn't have the initial qubit anymore.

We can sum up the quantum teleportation with the following table:

Result of measurement	Classical bits	Bob's state	Gate to apply	Bob's final state
$|\Phi^+\rangle$	00	$\alpha |0\rangle + \beta |1\rangle$	$\mathbb{1}$	$|\phi\rangle$
$|\Phi^-\rangle$	01	$\alpha |0\rangle - \beta |1\rangle$	$\boldsymbol{\sigma_z}$	$|\phi\rangle$
$|\Psi^+\rangle$	10	$\alpha |1\rangle + \beta |0\rangle$	$\boldsymbol{\sigma_x}$	$|\phi\rangle$
$|\Psi^-\rangle$	11	$\alpha |1\rangle - \beta |0\rangle$	$\boldsymbol{\sigma_z}\boldsymbol{\sigma_x}$	$|\phi\rangle$

Table 2.4: Quantum teleportation

2.8 Quantum Shannon Theory

2.8.1 VON NEUMANN entropy

In his book from 1932 [22], John VON NEUMANN introduced the notion of entropy for a quantum state by analogy with the thermodynamical GIBBS entropy.

For a quantum state $\boldsymbol{\rho}$, the VON NEUMANN entropy $S(\boldsymbol{\rho})$ is defined by

$$S(\boldsymbol{\rho}) = - \operatorname{Tr}(\boldsymbol{\rho}\ln(\boldsymbol{\rho})) \tag{2.42}$$

It has the following properties:

- $S(\boldsymbol{\rho}) \geq 0$;

- $S(\boldsymbol{\rho}) = 0$ if and only if $\boldsymbol{\rho}$ is a pure state;

- $S(\boldsymbol{\rho}) \leq n$ where n is the dimension of the Hilbert space. When the equality holds, we say that the state is maximally entangled.

The VON NEUMANN entropy is then a measure of how entangled a state is. A maximally entangled state (for instance a BELL state) is a state that maximises the VON NEUMANN entropy. If the Hilbert space if finite-dimensional, then the maximal value for S is reached when $\boldsymbol{\rho} = \frac{\mathbb{1}}{\dim(\mathcal{H})}$.

If we have two states $\boldsymbol{\rho_X}$ and $\boldsymbol{\rho_Y}$, it is possible to define the conditional VON NEUMANN entropy:

$$S(\boldsymbol{\rho_X}|\boldsymbol{\rho_Y}) = S(\boldsymbol{\rho_{XY}}) - S(\boldsymbol{\rho_Y}) \tag{2.43}$$

and the relative VON NEUMANN entropy:

$$S(\boldsymbol{\rho_X}||\boldsymbol{\rho_Y}) = -\operatorname{Tr}(\boldsymbol{\rho_X}\ln(\boldsymbol{\rho_Y})) - S(\boldsymbol{\rho_X}) = \operatorname{Tr}(\boldsymbol{\rho_X}(\ln(\boldsymbol{\rho_X}) - \ln(\boldsymbol{\rho_Y}))) \tag{2.44}$$

2.8.2 Mutual information

In the classical world, if X and Y are two random variables, we can quantify the amount of information that can be deduced from one variable by observing the other with mutual information (also called SHANNON mutual information):

$$I(X:Y) = D_{KL}(P_{(X,Y)}||P_X \otimes P_Y) \tag{2.45}$$

where D_{KL} is the KULLBACK-LEIBLER divergence, $P_{(X,Y)}$ is the joint distribution of X and Y and P_X, P_Y are the marginal distributions.

It can be written
$$I(X:Y) = H(X,Y) - H(X|Y) - H(Y|X) \tag{2.46}$$
where $H(X,Y)$ is the joint entropy, and $H(X|Y)$ and $H(Y|X)$ are conditional entropies.

The mutual information is positive and symmetric

$$\begin{aligned} I(X:Y) &\geq 0 \\ I(X:Y) &= I(Y:X) \end{aligned} \tag{2.47}$$

In analogy, let $\mathcal{H}_{XY} = \mathcal{H}_X \otimes \mathcal{H}_Y$ be the product of two Hilbert spaces and $\boldsymbol{\rho_{XY}}$ a state in $\mathcal{H}_{XY}$. Applying partial traces we have:

$$\begin{aligned} \boldsymbol{\rho_X} &= \operatorname{Tr}_Y(\boldsymbol{\rho_{XY}}) \\ \boldsymbol{\rho_Y} &= \operatorname{Tr}_X(\boldsymbol{\rho_{XY}}) \end{aligned} \tag{2.48}$$

We define the quantum mutual information (or VON NEUMANN mutual information) by

$$I(X:Y) = S(\boldsymbol{\rho_X}) + S(\boldsymbol{\rho_Y}) - S(\boldsymbol{\rho_{XY}}) \tag{2.49}$$

in analogy with the classical case. We immediately see that the mutual information is symmetric.

Then

$$\begin{aligned} I(X:Y) &= S(\boldsymbol{\rho_X}) + X(\boldsymbol{\rho_Y}) - S(\boldsymbol{\rho_{XY}}) \\ &= -\operatorname{Tr}_X(\boldsymbol{\rho_X}\ln(\boldsymbol{\rho_X})) - \operatorname{Tr}_Y(\boldsymbol{\rho_Y}\ln(\boldsymbol{\rho_Y})) + \operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}\ln(\rho_{XY})) \\ &= -\operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}\ln(\boldsymbol{\rho_X}\otimes\mathbb{1}_Y)) - \operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}\ln(\mathbb{1}_X\otimes\boldsymbol{\rho_Y})) + \operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}\ln(\boldsymbol{\rho_{XY}})) \\ &= \operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}(\ln(\rho_{XY}) - \ln(\boldsymbol{\rho_X}\otimes\mathbb{1}_Y) - \ln(\mathbb{1}_X\otimes\boldsymbol{\rho_Y}))) \\ &= \operatorname{Tr}_{XY}(\boldsymbol{\rho_{XY}}(\ln(\rho_{XY}) - \ln(\boldsymbol{\rho_X}\otimes\boldsymbol{\rho_Y}))) \\ &= S(\boldsymbol{\rho_{XY}}||\boldsymbol{\rho_X}\otimes\boldsymbol{\rho_Y}) \end{aligned}$$

$$I(X:Y) = S(\boldsymbol{\rho_{XY}}||\boldsymbol{\rho_X}\otimes\boldsymbol{\rho_Y}) \tag{2.50}$$

The relative entropy is the quantum analog of the KULLBACK-LEIBLER divergence.

The relative entropy can be shown non-negative and hence, we have the two following properties for the quantum mutual information.

$$I(X:Y) \geq 0$$
$$I(X:Y) = I(Y:X) \tag{2.51}$$

2.8.3 HOLEVO's theorem

Theorem (HOLEVO's theorem [23]): Let $\rho_1, \rho_2, \cdots, \rho_n$ be mixed states and X a random variable with $P(X = i) = p_i$ for $1 \leq i \leq n$. Let ρ be

$$\rho = \sum_i p_i \rho_i \tag{2.52}$$

and Y the outcome of measurement on ρ. Then,

$$I(X:Y) \leq \chi$$
$$\chi = S(\rho) - \sum_i p_i S(\rho_i) \tag{2.53}$$

χ is called HOLEVO information.

This theorem, also called HOLEVO's bound, states that there is an upper bound of the information that can be known on a quantum state by performing a measurement.

Chapter 3

Quantum Key Distribution

The goal of Quantum Key Distribution (QKD) is to share a string of random bits between two parties, using quantum technologies and ensure that only the two parties have access to a meaningful part of the key.

We will first discuss in more detail the utility of having a secret random string of bits shared between two parties. Then we will discuss `BB84` which is historically the first scheme of Quantum Key Distribution, and also the first time Quantum Cryptography gained interest in the scientific community and became something possible. Next, we will talk about entangled Quantum Key Distribution with the protocols `Ekert91` and `BBM92`, and we will also discuss quantum repeaters and relays by the same occasion. Finally, we will relax one assumption of Quantum Key Distribution and take an interest in Device Independent Quantum Key Distribution (DIQKD).

As in most examples of cryptography, we will denote by Alice (A) and Bob (B) the two parties that want to exchange a key, and Eve (E) a malicious party that wants to gain information on the key.

3.1 What to do with a random key ?

But surely, Alice and Bob want to share some things more interesting than a string of random numbers, like a secret photo, a very important password, or what they have eaten for lunch.

But they can do it. Imagine they already share the secret key

$$01101001$$

want to share the secret string

$$\text{Banana Pie}$$

Using the UTF8 encoding, the B will become in binary

$$01000010$$

The encrypted version of the message is the modulo 2 addition between the plain message and the key:

$$
\begin{aligned}
E &= U \oplus_2 K \\
U &= E \oplus_2 K
\end{aligned}
\tag{3.1}
$$

the second equation comes from the fact that $\oplus_2 \cdot \oplus_2 = \mathbb{1}$. Then, Alice's encrypted message will begin with:

	0	1	0	0	0	0	1	0	U
$\oplus_2$	0	1	1	0	1	0	0	1	K
=	0	0	1	0	1	0	1	1	E

Table 3.1: Encryption of B of Banana Pie

and Bob can easily decrypt the message with:

	0	0	1	0	1	0	1	1	E
$\oplus_2$	0	1	1	0	1	0	0	1	K
=	0	1	0	0	0	0	1	0	U

Table 3.2: Decryption of B of Banana Pie

We have a symmetric scheme, i.e. we have to securely share a secret key and quantum mechanics can help us.

This secret key could also be used with a standard classical encryption scheme like AES.

To assure maximal privacy, each bit of the private key should be used to encrypt one bit of a message, meaning that when Alice and Bob have run out of bits from the key, they should exchange a new key: this is the one-time pad (OTP).

3.2 The oldest protocol: BB84

3.2.1 Assumptions

For the time being, we make the following assumptions:

 i) Alice and Bob try to cooperate;

 ii) Alice and Bob share a perfect (i.e. without loss) quantum channel that can be subject to eavesdropping;

 iii) Alice and Bob share a classical communication channel, that can be subject to *passive* eavesdropping. This channel transfers the information without losses or errors (using classical error correction for instance);

 iv) Alice and Bob each have a truly trusted random generator;

 v) Alice and Bob are in secure physical locations (i.e. no information can go out except for the two above channels);

 vi) Alice and Bob have trusted classical devices;

 vii) Alice and Bob have trusted measurement devices.

We will see how can those assumptions can be relaxed. Under those assumptions, it is possible to design a theoretically secure key distribution protocol.

3.2.2 The protocol

The BB84 protocol is one of the first examples of plausible Quantum Cryptography. The aim is to share a secret key without any pre-shared secret (or a very small one). It was designed by BRASSARD and BENNETT in 1984 [24].

Alice

Bob

Quantum channel

00101100

$|0\rangle\,|0\rangle\,|-\rangle\,|+\rangle\,|-\rangle\,|1\rangle\,|0\rangle\,|0\rangle$

01101110

$\mathcal{B}_z\mathcal{B}_z\mathcal{B}_x\mathcal{B}_x\mathcal{B}_x\mathcal{B}_z\mathcal{B}_z\mathcal{B}_z$ $\mathcal{B}_z\mathcal{B}_x\mathcal{B}_x\mathcal{B}_z\mathcal{B}_x\mathcal{B}_z\mathcal{B}_x\mathcal{B}_z$

Classical channel
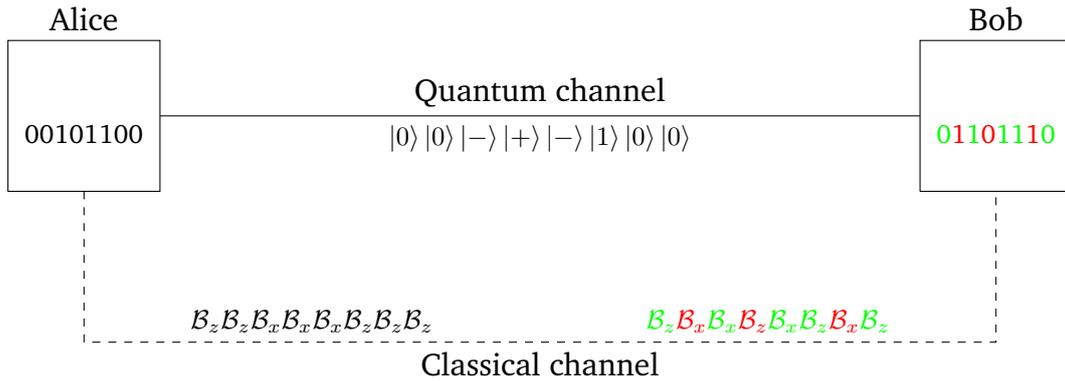
Figure 3.1: Schematic diagram of BB84

The exchange proceeds as follows:

1. Alice chooses a random bit using the random generator;

2. Alice chooses a random basis $\mathcal{B}_z$ or $\mathcal{B}_x$ using the random generator;

3. Alice encodes the chosen bit in the chosen basis and sends the qubit to Bob using the quantum channel;

4. Bob chooses a random basis $\mathcal{B}_z$ or $\mathcal{B}_x$ using the random generator;

5. Bob measures the received qubit in the chosen basis and decodes the bit. If Bob has the same basis as Alice, they share the same bit, and, if not, they share the same bit with a probability of $\frac{1}{2}$ and a different bit with the same probability (section 2.5).

6. Alice and Bob repeat the steps 1 to 5 until a reasonable amount (we will see what we mean by that) of bits have been exchanged.

7. Then they share the bases used for encoding and decoding using the classical channel.

8. They discard every bit where the bases are not the same and keep the other ones, without revealing the value of the bits.

9. To verify that no one eavesdropped, they publicly share a reasonable amount of bits and verify they agree. If they agree on all the bits, they discard the used bits and keep the key. If they disagree, they throw away the key and start again.

At the end of the operation, Alice and Bob share an identical and secure private key. Let's see an example before seeing what could go wrong if Eve tries to gain information on the exchanged bits.

Let's see an example:

n	A's bit	A's basis	A's qubit	B's basis	B's qubit	B's bit	Key
1	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	0	0
2	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_x$	$|+\rangle\,/\,|-\rangle$	0/1	
3	1	$\mathcal{B}_x$	$|-\rangle$	$\mathcal{B}_x$	$|-\rangle$	1	1
4	0	$\mathcal{B}_x$	$|+\rangle$	$\mathcal{B}_z$	$|0\rangle\,/\,|1\rangle$	0/1	
5	1	$\mathcal{B}_x$	$|-\rangle$	$\mathcal{B}_x$	$|-\rangle$	1	1
6	1	$\mathcal{B}_z$	$|1\rangle$	$\mathcal{B}_z$	$|1\rangle$	1	1
7	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_x$	$|+\rangle\,/\,|-\rangle$	0/1	
8	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	0	0

Table 3.3: Example of realization of the BB84 protocol

When the bases are the same (1, 3, 5, 6, 8), Alice and Bob share the same bit.

When the bases are not the same, we have something like $|0\rangle / |1\rangle$ which means that the measurement will give either outcome with a probability of $\frac{1}{2}$, but Bob doesn't know, *a priori* that he didn't have the same basis as Alice. For Bob the sequence might be:

$$00111110$$

During the phase where Alice and Bob share their bases, they will discard the bits 2, 4, and 7 and at the end, they will share the secret key:

$$01110$$

Before discussing the performances of this algorithm, let's see how it protects against passive eavesdropping.

Let's suppose we have an eavesdropper Eve. She measures in a random basis (or always the same basis, in either case it will lead to problems) and re-encodes the bit in the same basis she measured. She measures and sends with no notable delay.

Here is an example of realisation (this time instead of letting $|0\rangle / |1\rangle$, we randomly choose measurement outcomes)

n	A's bit	A's basis	A's qubit	E's basis	E's qubit	B's basis	B's qubit	B's bit	Key
1	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	0	0
2	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_x$	$|-\rangle$	1	
3	1	$\mathcal{B}_x$	$|-\rangle$	$\mathcal{B}_z$	$|1\rangle$	$\mathcal{B}_x$	$|+\rangle$	0	0
4	0	$\mathcal{B}_x$	$|+\rangle$	$\mathcal{B}_x$	$|+\rangle$	$\mathcal{B}_z$	$|1\rangle$	1	
5	1	$\mathcal{B}_x$	$|-\rangle$	$\mathcal{B}_x$	$|-\rangle$	$\mathcal{B}_x$	$|-\rangle$	1	1
6	1	$\mathcal{B}_z$	$|1\rangle$	$\mathcal{B}_z$	$|1\rangle$	$\mathcal{B}_z$	$|1\rangle$	1	1
7	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_x$	$|-\rangle$	1	
8	0	$\mathcal{B}_z$	$|0\rangle$	$\mathcal{B}_x$	$|+\rangle$	$\mathcal{B}_z$	$|0\rangle$	0	0

Table 3.4: Example of realization of the BB84 protocol with eavesdropper

We only look at bits 1, 3, 5, 6, and 8 because the others are discarded. On the bits 1, 5, 6, Alice and Eve share the same basis so Eve cannot be detected. With the 8th bit, Eve doesn't have the same basis as Alice but chance made Alice and Bob share the same bit at the end. However, with the 3rd bit, Eve has not the same basis as Alice, and this time Alice and Bob don't share the same bit value. By sharing this bit for verification at the end, they will see that Eve was present.

But we can ask ourselves: how many bits do Alice and Bob have to share in order to ensure to find Eve if she exists.

Let's denote $\mathcal{B}^A, \mathcal{B}^E, \mathcal{B}^B$ the bases of, respectively, Alice, Eve, and Bob at an iteration of the protocol. There are four, equally probable, possibilities:

$$
\begin{aligned}
\mathcal{B}^A = \mathcal{B}^E = \mathcal{B}^B \quad & p = \frac{1}{4} \\
\mathcal{B}^A = \mathcal{B}^E \neq \mathcal{B}^B \quad & p = \frac{1}{4} \\
\mathcal{B}^A \neq \mathcal{B}^E = \mathcal{B}^B \quad & p = \frac{1}{4} \\
\mathcal{B}^A = \mathcal{B}^B \neq \mathcal{B}^E \quad & p = \frac{1}{4}
\end{aligned}
\tag{3.2}
$$

The cases where $\mathcal{B}^A \neq \mathcal{B}^B$ are not interesting since the bits will be discarded. Also, the case where $\mathcal{B}^A = \mathcal{B}^E = \mathcal{B}^B$ provides no information.

Let's look at the case $\mathcal{B}^A = \mathcal{B}^B \neq \mathcal{B}^E$. Imagine, without loss of generality, that $\mathcal{B}^A = \mathcal{B}^B = \mathcal{B}_z$ and $\mathcal{B}^E = \mathcal{B}_x$ and that Alice ant to encode a 0.
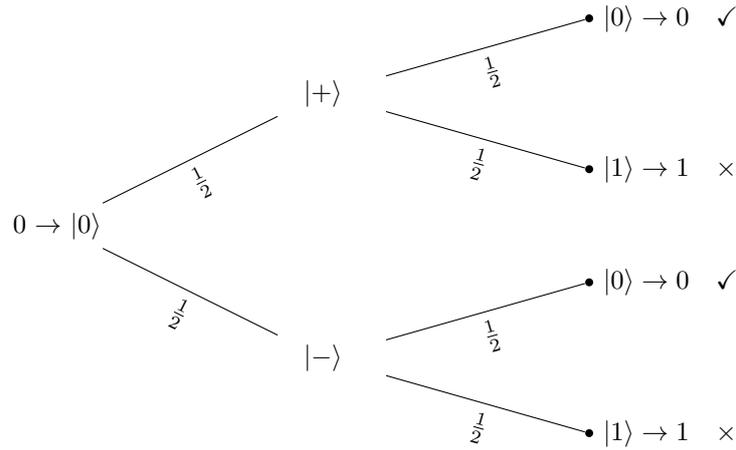
Figure 3.2: Probability tree

Then if Eve is present, and her basis is different from the one of Alice and Bob, there is probability of $\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = \frac{1}{2}$ to find her.

After discarding the bits, we have only two possibilities for the bases:

$$
\begin{aligned}
\mathcal{B}^A = \mathcal{B}^E = \mathcal{B}^B \quad p = \frac{1}{2} \\
\mathcal{B}^A = \mathcal{B}^B \neq \mathcal{B}^E \quad p = \frac{1}{2}
\end{aligned}
\tag{3.3}
$$

And then the probability to not find her by revealing one bit is

$$
P(\mathcal{B}^A = \mathcal{B}^E = \mathcal{B}^B) + P(\mathcal{B}^A = \mathcal{B}^B \neq \mathcal{B}^E) \times P(\text{Alice and Bob have same bit}) = \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4} \tag{3.4}
$$

As the events are supposed independents, the probability to not find Eve by giving away k bits is

$$
\overline{p_k} = \left(\frac{3}{4}\right)^k \tag{3.5}
$$

And then the probability to find Eve, assuming she exists, by giving away k bits is

$$
p_k = 1 - \left(\frac{3}{4}\right)^k \tag{3.6}
$$

The more bits are given away, the more chance we have to find Eve.
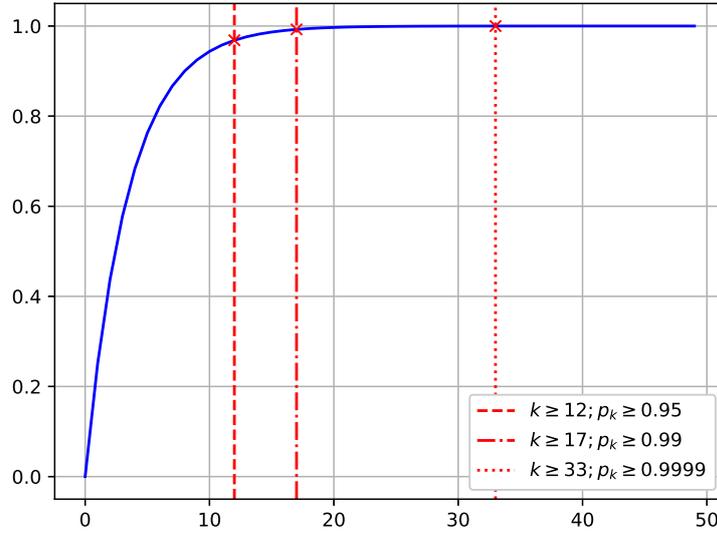
Figure 3.3: Probability to find Eve assuming she exists

Let's see the efficiency of the protocol. The length of the key L is given by

$$L = N - N_{incorrect} - k \tag{3.7}$$

where N is the number of sent qubits, $N_{incorrect}$ the number of mismatching bases, and k the number of qubits given away to see if Eve is present. On average, $N_{incorrect} = \frac{N}{2}$. If we give a fraction of what is left atfer the announcement of the bases, let say $k = \frac{L}{3}$, then

$$
\begin{aligned}
L &= N - \frac{N}{2} - \frac{L}{3} \\
L &= \frac{3}{8}N
\end{aligned}
\tag{3.8}
$$

We can define the efficiency $\eta = \frac{L}{N}$ and then

$$\eta = \frac{3}{8} = 37.5\% \tag{3.9}$$

3.2.3 Relaxing assumptions

Now, we review the assumptions that we made for QKD and we see how we can relax them.

Assumption (i): Alice and Bob try to cooperate

Even if this assumption seems to be useless, there are contexts where the parties cannot trust each other and want to cheat. This can be studied as quantum coin flipping (or tossing) and is studied in section 5.2.

Assumption (ii): Alice and Bob share a perfect (i.e. without loss) quantum channel, that can be subject to eavesdropping;

In practice, quantum channels are noisy. We see this idea in subsection 3.2.4.

Also, the problem of active eavesdropping must be addressed.

What we mean by *passive* eavesdropping is to listen to communications on the channel without trying to alter or delete the data or to impersonate one of the legitimate users of the channel.

If we don't want active eavesdropping, we need an authenticated quantum channel. This can be done using a small pre-shared secret (see [25]).

Assumption (iii): Alice and Bob share a classical communication channel, that can be subject to *passive* eavesdropping. This channel transfers the information without losses or errors (using classical error correction for instance);

Alice and Bob need to share a classical channel to send the information about the bases used and about the bits that are given away to see if Eve is present. This channel can be subject to passive eavesdropping because the information transmitted using this channel gives no information about the key, except the maximal length.

In practice, we could have *active* eavesdropping on the classical channel. Hence, there is a need to check the authenticity and the integrity of the messages (i.e. to verify that the message was indeed sent by the good person and that it was not altered in any way during the communication). As said in [26], this matter is often skipped in the QKD literature as it is classical considerations that are already well treated.

There are different ways to do that:

- Using a small, already shared, private key. Then QKD is viewed as a private key enlarger since you can create a bigger private key using a small one;

- Using public-key cryptosystems.

Assumption (iii): Alice and Bob each have a truly trusted random generator

The random generator is used to choose the bits and the bases. If the random generator is biased, information on the key or to attack could be used

However, if the generator is not trusted (for example, it was modified by a third malicious party to play a predefined sequence that looks like random), then, it would be easy for an attacker to get access to the key.

The idea of a quantum random generator for cryptography is discussed in section 5.1.

Assumption (iv): Alice and Bob are in secure physical locations (i.e. no information can go out except for the two above channels)

A malicious third party could have a camera inside the lab, or a module, on the measurement apparatus for example, that gives result of the measurements. We suppose that no information can leak out the physical place to prevent this.

Assumption (v): Alice and Bob have trusted classical devices

Again, an eavesdropper could have tricked the classical device before the exchange to substitute the secret key by a pre-decided key. This assumption cannot be relaxed.

Assumption (vi): Alice and Bob have trusted measurement devices

For this protocol to work, Alice and Bob assume that the measurement apparatus gives what they are expected to give. But they could have been modified by a malicious third-party. They cannot transmit the result of the measurements, due to assumption iv but they could be tricked in some way. To relax this assumption, we need to talk about Device Independent Quantum Key Distribution (DIQKD), see section 3.4.

3.2.4 The issue with the distance: error correction and privacy amplification

In practice, neither the quantum channel nor the source of qubits is perfect.

If we take the example of photons, we need a single-photon source which is experimentally really hard to produce. For example, using attenuated LASER pulses, we could have no photon emitted, or 1 photon emitted or 2 or more photons emitted. In the last case, the extra photons could be used to make a so-called photon number splitting (PNS) attack [27]. The basic idea is that the eavesdropper Eve keeps one photon over the n photons of the pulse and sends the $(n-1)$ other photons to Bob. Then when the bases are announced over the classical channel, Eve measures the photons when Bob and Alice agreed and she gets the sifted key.

Another issue is the fact that neither the quantum channel nor the detectors are perfect. First, there are losses in the optical fiber and following [27, 28], we can model the losses by

$$F = 10^{\frac{-\alpha d + c}{10}} \tag{3.10}$$

- F is called the transmission efficiency of the fiber;

- α are the losses of the fiber in $\mathrm{dB\,km^{-1}}$;

- d is the distance in km;

- c are eventual distance-independent losses on the fiber;

Now the detector is not perfect neither: there are two cases

- the photon is transmitted on the fiber, but the detector doesn't detect it. We denote η the probability for a detector to detect the photon;

- no photon is transmitted, but one of the detectors detects something. It is a dark count and is due to thermal noise. We denote D the probability for a detector to do a dark count.

It is also possible that the two detectors of Bob detect something simultaneously (two dark counts or one dark count and a real photon). We suppose that those events are discarded from the key (instead of noting the basis he used, Bob notes "fail" for the basis, and when Alice and Bob share the bases, Alice will discard the failed measurements).

Then we can calculate the probability that a bit goes into the sifted key. First, Bob must choose the same basis as Alice (otherwise, the bit is immediately discarded). It has probability $\frac{1}{2}$. Then there are two cases:

- the photon passes and is detected on the detector. Also, no dark count is registered on the second detector. The probability is
$$\frac{1}{2}F\eta(1-D) \tag{3.11}$$

- no photon passes or a photon passes but not detected and a dark count is registered and one detector (but not on the other). The probability is

$$\frac{1}{2}(1-F\eta)2D(1-D) \tag{3.12}$$

So the total probability that a bit contributes to the sifted key is

$$p_{total} = \frac{1}{2}\left(F\eta + (1-F\eta)2D\right)(1-D) \tag{3.13}$$

Next, we want to know which bits are correctly received. It is possible to define the visibility V: the signal passes perfectly in the fiber with probability V and becomes noise with probability $1-V$.

Then, to be a correct bit, the photon must pass the fiber without being absorbed (F) and without becoming noise (V), must be detected (η) at the detector and there must be no dark count on the other detector ($1 - D$). Also, Bob needs to choose the same basis as Alice ($\frac{1}{2}$). Hence the probability to accept a correct bit photon in the sifted key is

$$p_{signal} = \frac{1}{2} F \eta V (1 - D) \tag{3.14}$$

Then the signal to noise ratio, or visibility (Bob's visibility of Alice's bit) is

$$V_B = \frac{p_{signal}}{p_{total}} = \frac{F \eta V}{F \eta + (1 - F \eta) 2D} \tag{3.15}$$

This visibility decreases as the distance d increases. This sounds logical as more and more photons are absorbed. Then,

- At low distances, dark counts D and the optical visibility of the fiber V are responsible for losses;

- At high distances, the losses come from the absorption of photons in the fiber.

Note that if we had only losses with the distance, we could do quantum key distribution at arbitrarily large distances anyway, as the losses would only make the bit rate decrease. But we have noise (visibility, noise by Eve, and dark counts) and hence, at one moment, the signal will become weaker than the noise, and the communication will become impossible.

For simulation, we take the following values (see [28, 29]):

- $\alpha = 0.25 \mathrm{dB\,km}^{-1}$

- $D = 10^{-4}$

- $V = 0.99$

- $\eta = 0.3$

- $c = 0$
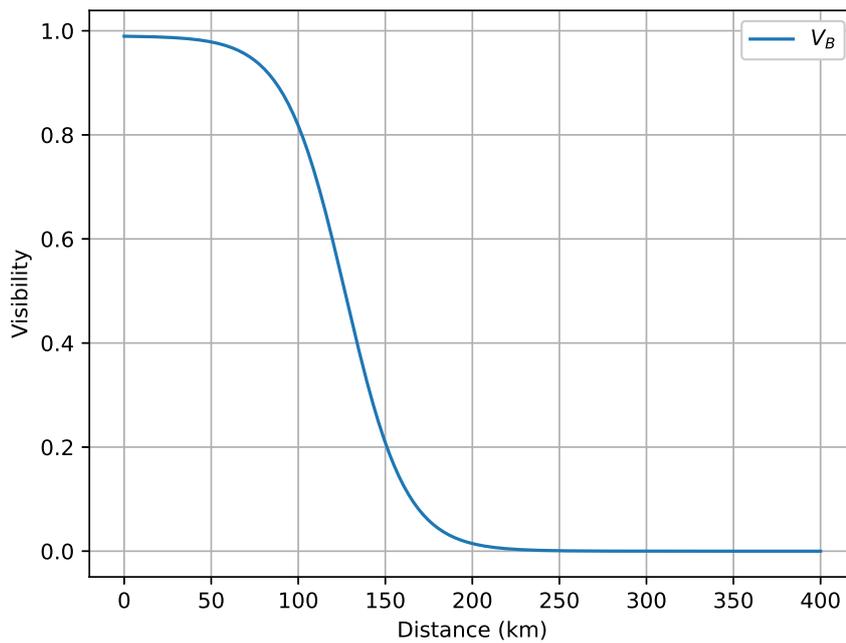
Figure 3.4: Bob's visibility of Alice's bit

How these unavoidable errors are handled by Alice and Bob and from which distance the QKD becomes impossible are the questions we will try to answer next.

Before starting the protocol, Alice and Bob agree on a qubit rate. Alice sends the qubit at this rate and Bob measures at the same rate. If none of the detectors or the two of them tick, then Bob marks the qubit as a failure and this will be announced during announcement of the bases in order for Alice to delete the bits for her key.

We can then update the length of the key in equation 3.7, as $N_{incorrect}$ will on average be equal to

$$N_{incorrect} = (1 - p_{total})N = \left(1 - \frac{1}{2} \left(F\eta + (1 - F\eta)2D \right) (1 - D) \right) N \qquad (3.16)$$

$$L = \frac{1}{2} \left(F\eta + (1 - F\eta)2D \right) (1 - D)N - k \qquad (3.17)$$

Now, Alice and Bob share a key of the same length but with potential errors due to dark count and noise. This is a problem in two ways: first, when they discard bits to verify that no eavesdropper listened during the exchange and secondly, the need to share the exact same private key.

For the first part, Alice and Bob agree on a threshold. If, when they discard bits, they find more errors than the predefined threshold, they assume that an eavesdropper was listening and they discard the key, if not, they keep the key.

They then proceed to error correction and privacy amplification.

The goal of the error correction phase is to delete the errors (either natural or caused by an eavesdropper) in order for Bob and Alice to share the same private key. This is called *reconciliation*. This can be done using the classical channel, which means that an eavesdropper might gain information by listening to this step. This will be the role of privacy amplification to reduce the knowledge of potential eavesdropper.

A protocol for reconciliation was proposed in 1991, partly by the same authors that proposed the BB84 protocol several years before (BENNETT and BRASSARD), in an article called *Experimental Quantum Cryptography* and which discuss of the matter of reconciliation and privacy amplification [30].

The idea is to do parity checks on blocks of the sifted key.

Alice and Bob may beforehand decide and a random permutation of the key. Then, using the length of the sifted key and an estimation of the error rate, they split the sifted key into blocks in such a way that they expect to have zero or one error in each block. If the blocks are too large, they may not detect correctly the errors and if the blocks are too small, they will leak too much information.

Before explaining in more detail the protocol, let's just recap what is a parity check. The parity of a bit string is 0 if there is an even number of 1 in the string and 1 if there is an odd number of 1 in the string. The effect of a one-bit flip on the string is the flip of the parity.

If Alice and Bob share a string with some errors, they will have the same parity of there is an even number of errors. On the other hand, they will have a different parity if there is an odd number of errors.

Then, if the blocks are split correctly so that there is zero or one error in the block, they can detect an error by comparing the parities (if the parities are the same, there is no error and if the parities are different there is one error). If the block has two errors, they will have the same parities and will think that there is no error.

Before describing precisely the protocol, we must consider the fact that giving away the parity of blocks gives away some information. In order to reduce the leakage of information, Alice and Bob agree to delete the last bit of the block. At the bits are random, it will delete a zero with probability $\frac{1}{2}$ and a one with probability $\frac{1}{2}$. So it will leave the parity unchanged with probability $\frac{1}{2}$ and flip it with probability $\frac{1}{2}$, so the parity is random again.

So Alice and Bob split the sifted key into blocks and check the parity. If the parity is different, they assume to have one error and search it by a bisective search.

Definition (BINARY): The bisective search works like this (see [30, 31]), also called the `BINARY` primitive:

1. Bob splits the string in 2^a, compute the parity of the first half, and ask the parity to Alice over the classical channel. The parity computed by Bob is referred to as the *current parity*. The parity sent by Alice is referred to as the *correct parity*;

2. Bob compares the correct parity sent by Alice with the current parity. If the parities are different, the first half is selected and if the parities are the same, the second half is selected.

3. Bob announces the selected half of the string over the public channel, and they start over at step 1, the new string being the half of the string selected.

4. Eventually, they will find the error (in the [30] protocol, the error will be discarded since they discard the last bit after each parity check).

This bisective search finds the error in less than $\lceil \log(l) \rceil$ where l is the length of the initial string.

The `BINARY` primitive can only be run on a block with odd error parity.

---

^{a}If the block has an odd length, one of the sub-block will be one bit shorter than the other
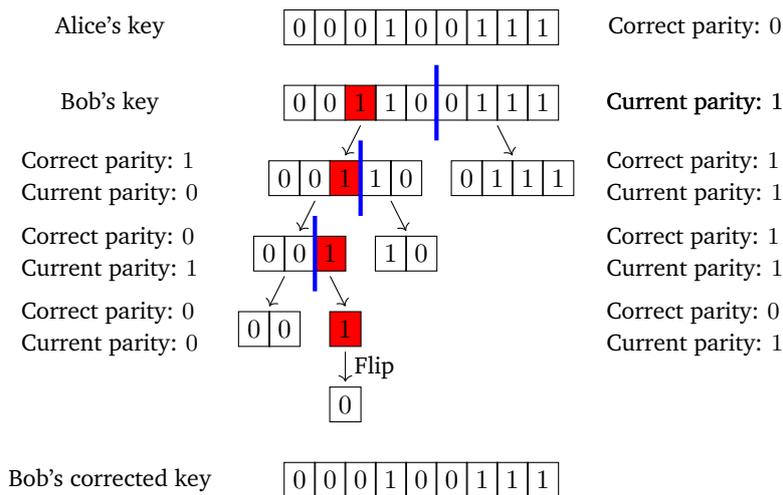

Figure 3.5: Run of `BINARY`

If there is more than one error, but an odd number of errors since `BINARY` can only ran on blocks with an odd number of errors, the `BINARY` primitive will correct only one error:
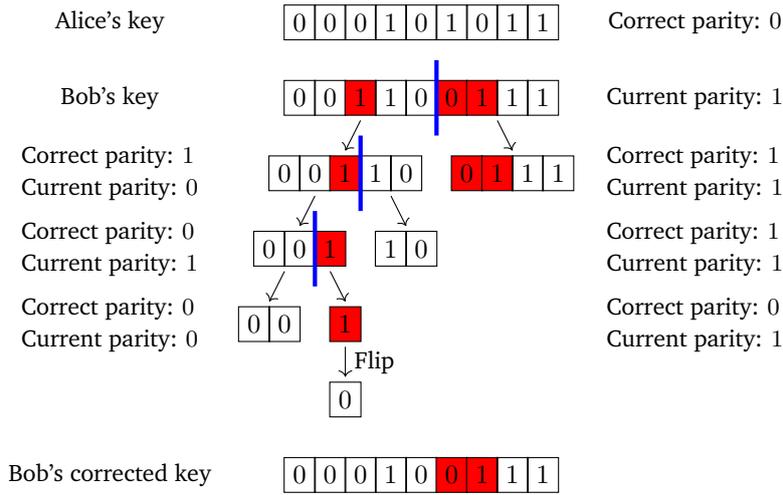
Figure 3.6: Run of `BINARY` with more than one error

Once they have done this process, they permute the sifted key again and they start again, with bigger blocks. Repeating this process will reduce the number of errors. This process becomes less and less efficient to find the last errors. In the end, Alice and Bob may change the strategy a little bit and compare parities of random subsets (announce publicly) and not ordered blocks.

At some point, they will successfully pass the parity check a predefined number of times, 20 times for example, in a row. They then assume to share an identical key.

This process of reconciliation needs a lot of bits, as a bit is discarded whenever a parity check is made.

We can update the formula 3.7:

$$L = N - N_{incorrect} - k - N_{parity} \tag{3.18}$$

when N_{parity} is the number of parity checks. N_{parity} will be related to the number of errors N_{errors} but we will not find the relation between the two (we will do that for the improved version of `BINARY`).

Using equation 3.14, we can estimate the number of errors:

$$N_{errors} = \left(1 - \frac{1}{2}F\eta V(1-D)\right)N \tag{3.19}$$

N_{errors} being the number of bits where Alice and Bob do not agree after announcing publicly the bases.

This protocol is not optimal and leaks too much information (resulting in the suppression of bits). In 1994, BRASSARD and SALVAIL proposed an optimal algorithm and a bit less optimal algorithm, but much more usable in practice [31].

For that purpose, they defined what is a reconciliation theorem, what is an optimal, efficient, ideal, and almost ideal reconciliation theorem.

Also, hey assimilated the quantum private channel to a binary symmetric channel. A Binary Symmetric Channel with crossover probability p (BSC$_p$) is a channel taking a binary value as input and giving a binary value as output. The bit is transmitted correctly with probability $1 - p$ and is flipped with probability p. It is represented schematically in figure 3.7.
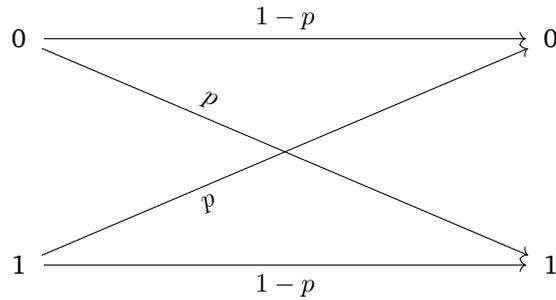
Figure 3.7: Binary symmetric channel

In the following, we denote the error rate, also called Quantum Bit Error Rate (QBER) by Q instead of p.

The proposed protocol is called `CASCADE`. It has similarities with the protocol presented in [30] but allows us to keep the bits instead of discarding them.

As before, the `CASCADE` protocol takes as input the noisy key and an estimated error rate and gives as output the reconciliated key and some information that was leaked on the public channel.

> **Definition (CASCADE protocol):** The `CASCADE` protocol works like this (each step will be discussed in further detail below).
>
> The protocol works in passes. At the beginning of each pass, except the first one, Bob shuffles the key. This shuffle can be announced over the public channel[a]. Then, Bob splits the key into equal blocks [b]. The number of blocks $k(i, Q)$ depends on the error rate Q and the pass i. These blocks are referred to as *top-blocks*.
>
> Then Bob computes the parity for each top-block and asks to Alice the correct parities. He compares the correct parities with the current parities and applies the `BINARY` primitive to each block with different parities.
>
> After applying the `BINARY` primitive, all top-blocks will have the correct parity. Bob then use the cascade effect to correct bits from the previous iterations.
>
> _____
> [a] It is also possible to only announce the final shuffle to Alice
> [b] The last block can be a bit shorter if the length is not a multiple of the number of blocks

Now let's understand two things: why the key is shuffled at the beginnings of each pass and why the protocol is called CASCADE.

The shuffling is useful to separate errors that are too close apart. Imagine the worst-case scenario (see figure 3.8) where two errors are just next to each other. If at the first pass, the two errors are in the same block (which is quite likely), without shuffling they would stay in the same block as the size of the block increases between passes. The shuffling allows those errors to be separated and to correct more errors.
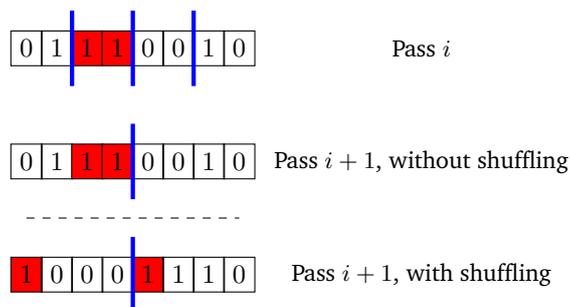

Figure 3.8: Shuffling process in the `CASCADE` protocol

The protocol is called `CASCADE` because of the cascade effect. As previously stated, a difference

between CASCADE and the protocol described in [30] is that bits are not discarded when parities are computed. Keeping the bits allows to correct bits from previous iterations.

Imagine that, at pass i a bit (in red in figure 3.9) is corrected. Then, Bob can apply the correction to previous passes, and one block in the previous pass will flip parity from even to odd (remember that after the end of a pass, each top-block has even parity). Then Bob can apply the BINARY primitive on these blocks to correct more errors, and the cascade will apply again. The cascade can be very deep, depending on the number of passes.
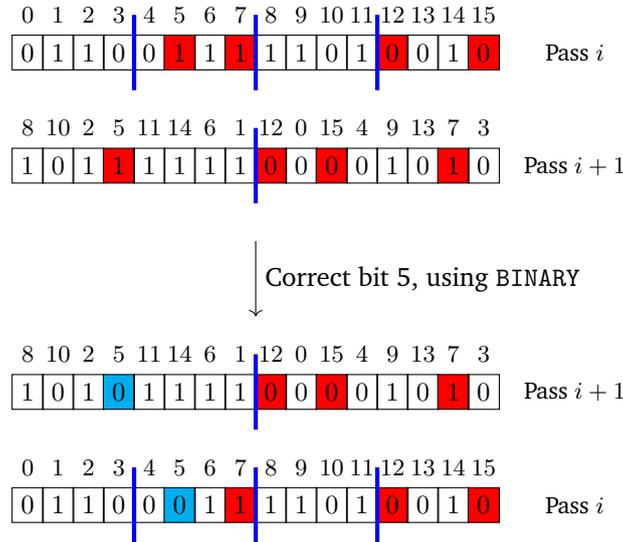

Figure 3.9: Cascade effect in the CASCADE protocol

At iteration $i + 1$, the bit 5 (in red) is corrected using BINARY on the first top-block. After correction, the bit is marked in blue. Looking at the iteration i, the second top-block has now an odd error parity and BINARY can be applied to correct the seventh bit. The cascade is then applied to all previous iterations additionally using the seventh bit.

Keeping the bits and exchanging the parities over the public channel means that some information is leaked to the eavesdropper. The protocol is a practical protocol that leaks more information than the theoretical bound but close enough to it for $Q \leq 0.15$ to be used in real applications.

Aside: does Alice send all the parities without Bob asking or does she only reply to questions from Bob. In the original protocol [31], Alice sends all the correct parities of the top-blocks (assuming that the shuffle was sent by Bob). But in [26], it was presented in a server-client infrastructure: Alice is just responding to Bob's "Ask parity" messages. In this scheme, Alice has the only mission to reply to the messages and to compute parities. By having such a light work, she could handle a large number of requests (remembering that she is playing the role of a server).

In [31], the CASCADE benchmark used 4 passes with the following number of blocks:

$$
\begin{aligned}
k(1, Q) &= \left\lfloor \frac{0.73}{Q} \right\rfloor \\
k(2, Q) &= 2 \times k(1, Q) \\
k(3, Q) &= 2 \times k(2, Q) \\
k(4, Q) &= 2 \times k(3, Q)
\end{aligned}
\tag{3.20}
$$

The size of the blocks doubles at each pass. The only thing we have to understand is the $\left\lfloor \frac{0.73}{Q} \right\rfloor$. Q being the error rate, if we consider blocks with size $\left\lfloor \frac{1}{Q} \right\rfloor$, we have on average blocks with one error. But we will also have blocks with 2 errors. By taking $\left\lfloor \frac{0.73}{Q} \right\rfloor$, we will have more blocks with zero errors, but less with two errors. Blocks with zero errors are fewer trouble than blocks with two errors (they are not corrected). We also have to bear in mind that the smaller the blocks the

greater we leak information.

Hence, for regular values of Q, we have the following block sizes

	$k(1,Q)$	$k(2,Q)$	$k(3,Q)$	$k(4,Q)$
$Q = 0.01$	73	146	292	584
$Q = 0.05$	14	28	56	112
$Q = 0.10$	7	14	28	56
$Q = 0.15$	5	10	20	40

Table 3.5: Block sizes for the 4 passes for different values of Q

As expected, the block size decreases as the error rate increases.

Note that the number of passes can be different from 4. It is determined by Alice and Bob beforehand, taking into consideration the parameter Q.

Note that the two error correction protocols don't ensure that the final keys will be the same. However, Alice and Bob can verify that they have the same string by producing a fingerprint of the key (see section 4.1 for a discussion on fingerprints). The hash (or fingerprint) will be identical if they share the same key and radically different even with small errors. Moreover, as the hash is exchanged over the classical channel, an eavesdropper can access it, but he can't compute the original strings in polynomial time. While they don't share an identical string, they repeat the error correction process.

Alice and Bob share an identical key but an eavesdropper has gained information on the key by two different ways:

- by eavesdropping without being detected during the quantum transmission;

- using the public information exchanged on the classical channel.

To eliminate the information gained by a potential eavesdropper, they proceed to privacy amplification. Again, this is done classically using the public channel. The first paper on privacy amplification was published in 1988 by BENNETT, BRASSARD, and ROBERT [32]. In 1995, BENNETT, BRASSARD, CREPEAU, and MAURER published a more general paper on the subject [33], called *Generalized Privacy Amplification*.

Note that privacy amplification can be applied to other situations than just Quantum Cryptography.

The goal of privacy amplification is to create a secret key from a larger common key on which an eavesdropper may have gained information, that is subject to some constraints.

To explain privacy amplification, we use the work of BENNETT, BRASSARD, ROBERT, CRÉPAU and MAURER [32, 33].

We suppose that the eavesdropper has access to t bits of information on the key, with $t < n$, n being the length of the shared key W between Alice and Bob (if $t = n$, no privacy amplification can be done because Eve knows the key).

Alice and Bob will use a compression function g, that will be publicly announced. The final key $K = g(W)$ will be r-bit long with $r < n$. r depends on g and the quantity of information that Eve has at its disposal. Eve will know only negligible information on K.

In [32, 33] emphasised the importance of universal$_2$ classes of functions, for the purpose of privacy amplification.

> **Definition (Universal$_2$ class of functions):** Following [34], let H be a class of functions from a set A to a set B. This class is said to be universal$_2$ if, for all x, y in A
>
> $$\sum_{f \in H} \delta_f(x,y) \leq \frac{|H|}{|B|} \tag{3.21}$$

where

$$\delta_f(x,y) = \begin{cases} 1 & \text{if} \quad x \neq y \quad \text{and} \quad f(x) = f(y) \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

and $|\cdot|$ denotes the cardinal. In other terms, it means that for all $x \neq y$ in A the probability to have $f(x) = f(y)$ should be less than $\frac{1}{|B|}$ when f is chosen at random in H.

Universal$_2$ classes of functions exist. For example, the class of all functions from A to B is universal$_2$. A more interesting result is the fact that the class of all linear functions from $\{0,1\}^n$ to $\{0,1\}^r$ is universal$_2$ [34]. Those functions are described by $n \times r$ matrices, and then there are 2^{rn} functions, so rn bits are necessary to describe a function (it is possible to use smaller universal$_2$ classes of functions and only n bits are needed to describe a function).

Alice and Bob, or at least one of the two, choose at random a function in the linear functions from $\{0,1\}^n \to \{0,1\}^r$ where $r = n - t - s$ ($0 < s < n - t$ is a safety parameter). If Alice chooses the function, she announces it to Bob using the classical channel (she also announces s), and Alice and Bob apply the function to the key. Eve will know the function as the classical channel is public but, as she doesn't know all the bits of the key, she will lose information in applying the function. In fact, it was proved in [33] that the number of bits known by Eve after privacy amplification is less than

$$\frac{2^{-s}}{\ln(2)} \quad \text{bits} \tag{3.23}$$

To choose r, Alice and Bob must have an estimation of t, the number of bits known by Eve. But remember, Alice and Bob already sacrificed a number of bits to estimate the error rate. They can make the assumption that all errors were introduced by an eavesdropper (even if some errors were effectively introduced by noise and detector errors). By doing so, they will probably over-estimate the number of bits known by Eve and have a shorter key than the optimal case, but at least, Eve would know fewer bits.

3.2.5 Summary and length of the final key

Now we are going to do a quick summary of all the steps in the BB84 protocol, also we are going to keep track of the number of bits in the key. There are three main steps:

1. Quantum communication;

2. Reconciliation;

3. Privacy amplification.

During the first step, Alice sends N_i qubits to Bob. Bob may not measure all qubits, he will sometimes have no detector that ticks or the two of them. In this case, the bit is just discarded and Bob marks the basis as failed.

So, when Alice and Bob share the bases, they will discard, on average

$$N_s = N_i - \left(\frac{1}{2} + \frac{p_{fail}}{2}\right) N_i \tag{3.24}$$

Following the previous notations, we have

$$p_{fail} = \eta D + (1-\eta)D^2 + (1-\eta)(1-D)^2 \tag{3.25}$$

Next, Alice and Bob will sacrifice a certain number of bits to estimate the error rate

$$N_e = N_s - p_e N_s \tag{3.26}$$

where p_e is the proportion of bits used to estimate the error rate. The error rate is denoted t_{errors}.

Now Alice and Bob proceed to error correction. If they use the CASCADE protocol they don't discard any bits, but all the corrections, as they are done publicly will increase the information known by Alice. They end up with a corrected key with

$$N_c = N_e \tag{3.27}$$

Finally, Alice and Bob proceed to privacy amplification. The final key length will be equal to

$$N_f = N_c - t - s \tag{3.28}$$

where t is

$$t = t_{errors} + t_{PA} \tag{3.29}$$

t_{errors} can be approximated by

$$t_{errors} = \left(1 - \frac{1}{2}F\eta V(1 - D)\right) N_e \tag{3.30}$$

and t_{PA}, the information gained by Eve during the privacy amplification phase will depend on t_{errors} (or more precisely on the effective number of errors). Using the CASCADE protocol, we can roughly estimate the number of parity bits that are given to Eve by

$$t_{PA} = \sum_{i=1}^{4} k(i, t_{errors}) = 15 \times \left\lfloor \frac{0.73}{t_{errors}} \right\rfloor \tag{3.31}$$

if there are 4 passes. Now we can all put together and

$$
\begin{aligned}
N_f &= N_c - t_{errors} - \left\lfloor \frac{0.73}{t_{errors}} \right\rfloor - s \\
&= N_e - \left(1 - \frac{1}{2}F\eta V(1 - D)\right) N_e - \left\lfloor \frac{0.73}{\left(1 - \frac{1}{2}F\eta V(1 - D)\right) N_e} \right\rfloor - s \\
&= (1 - p_e)N_s - \left(1 - \frac{1}{2}F\eta V(1 - D)\right)(1 - p_e)N_s \\
&\quad - \left\lfloor \frac{0.73}{\left(1 - \frac{1}{2}F\eta V(1 - D)\right)(1 - p_e)N_s} \right\rfloor - s \\
&= (1 - p_e)\left(\frac{1}{2} - \frac{\eta D + (1 - \eta)D^2 + (1 - \eta)(1 - D)^2}{2}\right) N_i \\
&\quad - \left(1 - \frac{1}{2}F\eta V(1 - D)\right)(1 - p_e)\left(\frac{1}{2} - \frac{\eta D + (1 - \eta)D^2 + (1 - \eta)(1 - D)^2}{2}\right) N_i \\
&\quad - \left\lfloor \frac{0.73}{\left(1 - \frac{1}{2}F\eta V(1 - D)\right)(1 - p_e)\left(\frac{1}{2} - \frac{\eta D + (1 - \eta)D^2 + (1 - \eta)(1 - D)^2}{2}\right) N_i} \right\rfloor - s
\end{aligned}
\tag{3.32}
$$

It gives an approximate relation between N_f, N_i, and simulation parameters of the components. The only things Alice and Bob can choose are p_e, the number of passes in the CASCADE protocol, and s.

3.2.6 Eavesdropping strategy

Let's finish this discussion on the BB84 protocol by analyzing eavesdropping strategies. In the description of the protocol, we used elementary arguments to explain that the protocol is secure. But we only described a "intercept-and-resend" strategy, with an errorless channel. Here we are interested in an eavesdropper that can delay his measurement after the bases have been announced for instance, and what is the maximum information gain that an eavesdropper can have.

The information gain with an intercept-and-resend strategy was studied in [35]: Alice intercept with a measurement in the basis $\mathcal{B}_\theta$ which is the basis $\mathcal{B}_z$ rotated of an angle θ, and the information that Eve has on the raw key, per bit measured is

$$I(\theta) = 1 + p(\theta)\log_2(p(\theta)) + q(\theta)\log_2(q(\theta)) \tag{3.33}$$

where $p(\theta)$ is the probability of having the same bit as Alice sent

$$p(\theta) = \frac{1}{2} + \frac{\cos(\theta) + \sin(\theta)}{4} \tag{3.34}$$

and $q(\theta)$ is the probability of error

$$q(\theta) = \frac{1}{2} - \frac{\cos(\theta) + \sin(\theta)}{4} \tag{3.35}$$

$I(\theta)$ is maximum for $\theta = \frac{\pi}{4}$. Results for the information on the sifted key and the corrected key were also derived in this paper.

In [36], a more general situation was envisaged and results on Eve's information when POVM measurement where allowed were derived.

But here we are going to analyze the results of a paper by FUCHS *et al.* that derived a general bound on Eve's information and an optimal strategy for eavesdropping [37].

Here, we are considering a situation where Eve can delay her measurements until the bases have been announced. This is done as follows: when Eve receives a qubit from Alice, she makes a unitary interaction between the received qubit and a probe. The probe will then be measured after the bases have been announced. The considered measurement is general POVM measurement.

The derivation of the bound is quite long but the result is as follows. Eve's mutual information I is bounded by

$$I \leq \frac{1}{2}\phi\left(2\sqrt{Q(1-Q)}\right) \tag{3.36}$$

where $\phi : x \mapsto (1+x)\ln(1+x) + (1-x)\ln(1-x)$ and Q is the average error rate on both bases (from the natural noise). They also explained that the bound can be reached with an optimal strategy.

Finally, the key exchange can only be made if the mutual information between Alice and Bob is superior to the mutual information between Alice and Eve and between Bob and Eve, i.e. if

$$I_{AB} \leq \min(I_{AE}, I_{EB}) \tag{3.37}$$

then the exchange must not proceed. Using the expression that was derived it was proved that for

$$Q \geq \frac{1}{2} - \frac{1}{4}\sqrt{2} \simeq 14.6\% \tag{3.38}$$

then the exchange would not be secure.

Just before closing this section, let's name some known attacks that have been experimentally tested on some protocols of QKD [38] : we saw the "intercept-and-resend" attack, and we quickly

talked about photon number splitting attack. There is also the denial of service attack (you cut the fiber), where solutions include quantum networks and free space QKD. There are also phase randomisation and Trojan horse (Eve sends light in the quantum channel and analyses the back reflections) attacks that could be performed on the source. There are also attacks on the detectors, namely blinding (bling the detector using continuous wave illumination and bright pulses), time-shift (modulating detector efficiency over time with pulses), and dead-time (using dead time effect of single-photon detectors, i.e. the time needed after a detection to be able to detect another photon) attacks.

3.3 Entanglement-based QKD

In 1991, Artur EKERT proposed a new way to do Quantum Key Distribution using entanglement and BELL's theorem [39], a protocol known as Ekert91 or E91, which was closely followed by a modified version of BB84 that used quantum entanglement, the BBM92 protocol [40]. We start by describing the BBM92, which is conceptually closer to BB84.

3.3.1 The BBM92 protocol

The scheme requires again a quantum channel and a classical and authenticated channel where passive eavesdropping may be performed.

This time, instead of Alice sending qubits to Bob, an EPR pair is created and send to the parties (the EPR pair can be issued by Alice, Bob, or a third party). Alice and Bob then independently choose a basis, and measure in that basis, without telling the result to each other. After exchanging a predefined number of qubits, Alice and Bob share the bases they used over the classical channel, and each discards any measurement where they used different orientations.

As the two photons of an EPR pair are correlated, Alice and Bob should always have the same measurement outcome when measuring in the same basis (unless transmission errors, imperfect EPR, or eavesdropper), meaning that after this operation they end up with keys that should be equal, they are called the sifted keys.

Then Alice and Bob share some bits to try and estimate the error rate. If the error rate is above some threshold, they assume that the errors don't only come from natural errors, that an eavesdropper tried to gain some information and start a new exchange. Otherwise, they proceed to error correction and privacy amplification (as explained in subsection 3.2.4).
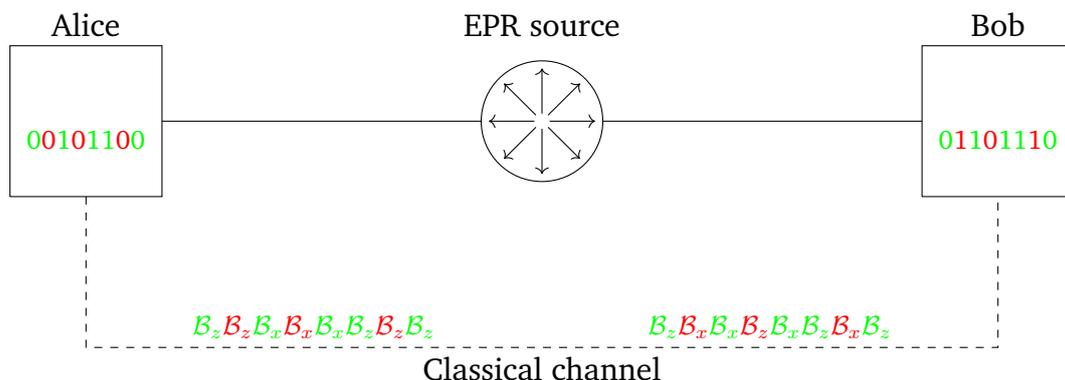

Figure 3.10: BBM92 principle schema

Now we can question the security of the protocol, and we have to examine the two main differences between BB84 and BBM92:

- In the BB84 scheme, the randomness of the key comes from a random generator (that can be classical or quantum, but is independent of the scheme) and assuming that Alice has good

intentions, this is good. Here the randomness is assured by the measurement of an EPR pair. It might be even a bit more secure than the original BB84 scheme;

- The EPR source can be located at Alice or Bob secure locations or at a third party. If the EPR source is located at, for instance, Alice's location, then the protocol is pretty much the same as BB84. In fact, Bob or any third party cannot distinguish if Alice is using an EPR source or not. In the second case, the source is placed between Alice and Bob. We will see in subsection 3.3.3 that it allows Quantum Key Distribution over longer distances but one be worried about a third malicious party modifying the EPR source.

In their paper, BENNETT *et al.* proved that a malicious third party Eve cannot modify the EPR source in order to gain information (the best she can, without increasing the error rate significantly, is a measurement uncorrelated from the EPR pair).

The scheme is really close to the BB84 one and at least as secure. It may have some advantages:

- If Alice and Bob decide to store the EPR pairs and to make the measurements just before the key is used, then the key cannot be leaked before its use as the information don't exist and is hidden, quantumly stored. In the other case, the key is stored classically and could be easily copied if anyone gained access to the physical locations;

- As we will see later, the maximal distance over which the exchange can be made is almost multiplied by 2 if the EPR source is placed in the middle.

3.3.2 The `Ekert91` protocol

The `Ekert91` protocol came before the `BBM92` protocol and is a bit more complicated as it uses BELL's inequality in order to detect a potential eavesdropper. It is considered less usable than the `BBM92` but can be used in a special case where the `BB884` and `BBM92` schemes cannot be used: Device Independent Quantum Key Distribution (see section 3.4).

Once again we have a source placed between Alice and Bob that emits pairs of entangled particles, in the state

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \tag{3.39}$$

Alice makes a measurement with a direction randomly chosen between $0, \frac{\pi}{8}, \frac{\pi}{4}$ (we denote the three possible directions a_1, a_2, a_3).

Bob makes a measurement with a direction randomly chosen between $-\frac{\pi}{8}, 0, \frac{\pi}{8}$ (we denote the three possible directions b_1, b_2, b_3).

Here note that we did not follow EKERT's original paper, were the pair would be in the state:

$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \tag{3.40}$$

and the angles are $0°, 45°, 90°$ for Alice and $45°, 90°, 135°$ for Bob. The idea is nevertheless the same, and the intermediary results only slightly different.

**(a)** Alice   **(b)** Bob
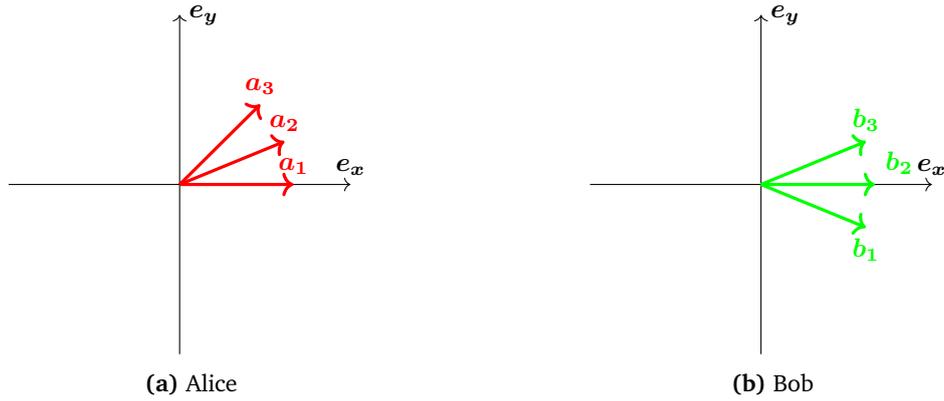
Figure 3.11: Directions for measurement for Alice and Bob

When Alice and Bob choose the same orientations (a_1, b_2 or a_2, b_3), they will end up with the same measurement outcome.

At the end of the quantum communication, Alice and Bob share the bases they used and they separate the results into two groups:

- The first group is composed of the results where Alice and Bob use different bases. This will be used to detect a potential eavesdropper.

- The second group is composed of the results where Alice and Bob used the same basis. This will be used to construct the secret key.

Then they publicly reveal the content of the first group. To understand this step, we have to define correlation coefficients. The correlation coefficient between a_i and b_j is

$$E(a_i, b_j) = \mathbb{P}_{00}(a_i, b_j) + \mathbb{P}_{11}(a_i, b_j) - \mathbb{P}_{01}(a_i, b_j) - \mathbb{P}_{10}(a_i, b_j) \tag{3.41}$$

where $P_{0/10/1}(a_i, b_j)$ is the probability for Alice to measure 0/1 with orientation a_i and for Bob to measure 0/1 with orientation b_j.

This can be written

$$E(a_i, b_j) = \mathbb{P}(a = b|ij) - \mathbb{P}(a \neq b|ij) \tag{3.42}$$

where a is the result of Alice's measurement and b the result of Bob's one.

Now let's compute a value for $E(a_i, b_j)$. We denote by θ_j^a and θ_j^b the angles corresponding to the orientations of measurement i.e.

$$\begin{aligned} a_i &= \cos(\theta_i^a)e_x + \sin(\theta_i^a)e_y \\ b_j &= \cos(\theta_j^b)e_x + \sin(\theta_j^b)e_y \end{aligned} \tag{3.43}$$

Then a measurement with orientation a_i corresponds to a measurement in the $\{|a_i\rangle, |a_i^\perp\rangle\}$ with

$$\begin{aligned} |a_i\rangle &= \cos(\theta_i^a)|0\rangle + \sin(\theta_i^a)|1\rangle \\ |a_i^\perp\rangle &= -\sin(\theta_i^a)|0\rangle + \cos(\theta_i^a)|1\rangle \end{aligned} \tag{3.44}$$

Hence, we have

$$\begin{aligned}
\mathbb{P}_{00}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= |\langle a_i b_j | \Phi^+ \rangle|^2 \\
&= \left| \frac{1}{\sqrt{2}} \left(\cos(\theta_i^a) \cos(\theta_j^b) + \sin(\theta_i^a) \sin(\theta_i^b) \right) \right|^2 \\
&= \frac{1}{2} \cos^2(\theta_i^a - \theta_j^b)
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}_{01}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= |\langle a_i b_j^\perp | \Phi^+ \rangle|^2 \\
&= \left| \frac{1}{\sqrt{2}} \left(-\cos(\theta_i^a) \sin(\theta_j^b) + \sin(\theta_i^a) \cos(\theta_i^b) \right) \right|^2 \\
&= \frac{1}{2} \sin^2(\theta_i^a - \theta_j^b)
\end{aligned}$$

We have similar expressions for $\mathbb{P}_{10}$ and $\mathbb{P}_{11}$ resulting in

$$\begin{aligned}
\mathbb{P}_{00}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= \frac{1}{2} \cos^2(\theta_i^a - \theta_j^b) \\
\mathbb{P}_{01}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= \frac{1}{2} \sin^2(\theta_i^a - \theta_j^b) \\
\mathbb{P}_{10}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= \frac{1}{2} \sin^2(\theta_i^a - \theta_j^b) \\
\mathbb{P}_{11}(\boldsymbol{a_i}, \boldsymbol{b_j}) &= \frac{1}{2} \cos^2(\theta_i^a - \theta_j^b)
\end{aligned} \tag{3.45}$$

Hence we have

$$E(\boldsymbol{a_i}, \boldsymbol{b_j}) = \cos^2(\theta_i^a - \theta_j^a) - \sin^2(\theta_i^a - \theta_j^b) = \cos\big(2(\theta_i^a - \theta_j^b)\big) \tag{3.46}$$

meaning that we have the following values:

i \ j	1	2	3
1	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$
2	0	$\frac{\sqrt{2}}{2}$	1
3	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$

Table 3.6: Value of the correlation coefficients

We then define a quantity when Alice and Bob choose different orientations:

$$S = E(a_1, b_1) + E(a_1, b_3) - E(a_3, b_1) + E(a_3, b_3) \tag{3.47}$$

Using table 3.6, we find

$$S = 2\sqrt{2} \tag{3.48}$$

Now let's see the effect of an eavesdropper. Let's suppose Eve tries to make a measurement on the two particles (she chooses a direction e_a for Alice's particle and e_b for Bob's particle), and after the measurement, she resends the same bit encoded in a qubit in the measurement basis.

The strategy of Eve is represented by a distribution probability:

$$\rho(\boldsymbol{e_a}, \boldsymbol{e_b}) \geq 0 \quad \iint \rho(\boldsymbol{e_a}, \boldsymbol{e_b}) \mathrm{d}^2 e_a \mathrm{d}^2 e_b = 1 \tag{3.49}$$

In this case

$$S(\boldsymbol{e_a}, \boldsymbol{e_b}) = E(\boldsymbol{a_1}, \boldsymbol{e_a})E(\boldsymbol{b_1}, \boldsymbol{e_b}) + E(\boldsymbol{a_1}, \boldsymbol{e_a})E(\boldsymbol{b_3}, \boldsymbol{e_b}) - E(\boldsymbol{a_3}, \boldsymbol{e_a})E(\boldsymbol{b_1}, \boldsymbol{e_b}) + E(\boldsymbol{a_3}, \boldsymbol{e_a})E(\boldsymbol{b_3}, \boldsymbol{e_b})$$
(3.50)

This expression can be simplified using a bit of algebra. We denote

$$\boldsymbol{e_a} = \cos(\theta_a^e)\boldsymbol{e_x} + \sin(\theta_a^e)\boldsymbol{e_y}$$
$$\boldsymbol{e_b} = \cos(\theta_b^e)\boldsymbol{e_x} + \sin(\theta_b^e)\boldsymbol{e_y}$$

and then

$$
\begin{aligned}
S(\boldsymbol{e_a}, \boldsymbol{e_b}) &= \cos(2(\theta_1^a - \theta_a^e))\cos\big(2(\theta_1^b - \theta_b^e)\big) \\
&+ \cos(2(\theta_1^a - \theta_a^e))\cos\big(2(\theta_3^b - \theta_b^e)\big) \\
&- \cos(2(\theta_3^a - \theta_a^e))\cos\big(2(\theta_1^b - \theta_b^e)\big) \\
&+ \cos(2(\theta_3^a - \theta_a^e))\cos\big(2(\theta_3^b - \theta_b^e)\big)
\end{aligned}
$$

This simplifies to

$$S(\boldsymbol{e_a}, \boldsymbol{e_b}) = \sqrt{2}\cos(2(\theta_a^e - \theta_b^e))$$
(3.51)

and the average value is

$$
\begin{aligned}
S &= \iint \rho(\boldsymbol{e_a}, \boldsymbol{e_b})S(\boldsymbol{e_a}, \boldsymbol{e_b})\mathrm{d}^2\boldsymbol{e_a}\mathrm{d}^2\boldsymbol{e_b} \\
&= \sqrt{2}\iint \rho(\boldsymbol{e_a}, \boldsymbol{e_b})\cos(2(\theta_a^e - \theta_b^e))\mathrm{d}^2\boldsymbol{e_a}\mathrm{d}^2\boldsymbol{e_b}
\end{aligned}
$$
(3.52)

We have

$$-1 \leq \cos(2(\theta_a^e - \theta_b^e)) \leq 1$$
(3.53)

and combining equations 3.49, 3.52, and 3.53, we end up with

$$-\sqrt{2} \leq S \leq \sqrt{2}$$
(3.54)

So, when Alice and Bob share the bits of the first group, they can compute the value of S. If they find $S = 2\sqrt{2}$, it means that the output they have was indeed obtained using entangled pairs. If they obtain $|S| \leq \sqrt{2}$, it means that an eavesdropper has perturbed the exchange (note that depending on the chosen angles, BELL's inequality may yield a different result. Often, one finds $|S| \leq 2$).

The advantage of the EKERT's protocol is the fact that, at a cost of one supplementary basis (i.e. two supplementary states), no information about the key is revealed to know if an eavesdropper was present (remember that in the BB84 and BBBM92 protocols, Alice and Bob estimate the error rate by publicly announcing bits from the key).

Now, the protocol is less efficient than the BB84 and BBM92 protocols. There are in total 9 combinations possible for the choice of the bases and only 2 of them contribute in the sifted. So the length of the sifted key, before any correction or privacy amplification is roughly

$$L = \frac{2}{9}N$$
(3.55)

but remember that this time, we don't give away any bit to verify the presence of an eavesdropper. A modified and more efficient version of EKERT's protocol is presented in section 3.4.

Having said all that, we must however remember that, even if Alice and Bob choose the same basis, their results are not perfectly correlated due to errors and will affect the value of S. Following the ideas in [41], we can define the Quantum Bit Error Rate (QBER) by

$$Q = \mathbb{P}(a \neq b | 12) + \mathbb{P}(a \neq b | 23) \tag{3.56}$$

Also, we use the following model. Instead of sharing the state $|\Phi^+\rangle$, Alice and Bob share the state

$$\rho = p \, |\Phi^+\rangle \langle \Phi^+| + (1-p)\frac{\mathbb{1}}{4} \tag{3.57}$$

(noisy state: the entangled pair is correctly shared with probability and become noise with probability 1-p).

Now, we can relate p and Q: they will have different outcomes when they should have the same, when: 1) the entangled pair becomes noisy $(1-p)$ and 2) Measurement outcomes, that are hazardous, are different $(\frac{1}{2})$, resulting in

$$Q = \frac{1}{2} - \frac{p}{2} \tag{3.58}$$

Note that, if $p = 1$, i.e., the perfect channel, we have $Q = 0$. If we have $p = 0$ (the noisiest channel), we have $Q = \frac{1}{2}$, which are totally uncorrelated results.

In the noisy case, the correlation coefficients are multiplied by p since results are uncorrelated when the signal becomes noise, resulting in

$$S = 2\sqrt{2}p = 2\sqrt{2}\left(\frac{1}{2} - \frac{p}{2}\right) \tag{3.59}$$

But, the security of this protocol is based on the violation of BELL's inequality. But if S is less than $\sqrt{2}$, i.e. when $p \leq \frac{1}{2}$, Eve's interference cannot be distinguished from noise. This corresponds to a quantum bit error rate of $Q = 25\%$.

3.3.3 Quantum relays and repeaters

As stated before in subsection 3.2.4, it is difficult to implement Quantum Key Distributions over long distances, in particular, due to the absorption of photons in the fiber. But we have physical realizations of QKD over hundreds of kilometers.

One way to easily increase the maximal distance is to use an entanglement-based QKD, with the EPR source placed between Alice and Bob. This would allow doubling the distance compared to QKD.

If we want to increase the distance even more, we need to use something more. Classically we use repeaters to increase communication distance. But in the quantum world, we cannot do the same because of the no-cloning theorem.

In 1998, BRIEGEL *et al.* proposed an idea of a quantum repeater [42]. This would allow to creating entangled pairs over an arbitrarily large distance.

But those quantum repeaters use entanglement purification and quantum memory, areas that need research. Nevertheless, there is a way to increase the maximal distance over which QKD can be achieved without those strong requirements. This is refereed to as quantum relays.

The idea of quantum relays is to extend what we did in putting the EPR between Alice and Bob: we have split the channel into two sections. It is possible to separate the channel in more than two sections. Before trying to generalize, let's try to compute Bob's visibility V_B in the case we use two segments.

The probability for the photon to pass half the fiber without being absorbed is then

$$10^{-\frac{\alpha \frac{d}{2}}{10}} = F^{\frac{1}{2}} \tag{3.60}$$

For a bit to contribute to the sifted key:

- Alice and Bob must choose the same basis: $\frac{1}{2}$;

- each photon must pass half the fiber without being absorbed and be detected, or being absorbed or being not detected and have a dark count on one of the detectors: $F^{\frac{1}{2}}\eta + (1 - F^{\frac{1}{2}}\eta)2D$;

- the other detector must have no dark count: $1 - D$.

resulting in the total probability:

$$p_{total,2} = \frac{1}{2} \left(F^{\frac{1}{2}}\eta + (1 - F^{\frac{1}{2}}\eta)2D \right)^2 (1 - D)^2 \tag{3.61}$$

To be not noisy,

- Alice and Bob must choose the same basis: $\frac{1}{2}$;

- each photon must pass half the fiber without being absorbed, without becoming noise, be detected and have no dark count on the other detector: $F^{\frac{1}{2}}\eta V(1 - D)$.

resulting in the total probability:

$$p_{signal,2} = \frac{1}{2} \left(F^{\frac{1}{2}}\eta V(1 - D) \right)^2 \tag{3.62}$$

Hence, Bob's visibility is

$$V_{B,2} = \frac{p_{signal,2}}{p_{total,2}} = \frac{(F^{\frac{1}{2}}\eta V)^2}{\left(F^{\frac{1}{2}}\eta + (1 - F^{\frac{1}{2}}\eta)2D \right)^2} \tag{3.63}$$

We compare V_B and $V_{B,2}$ in figure 3.12.
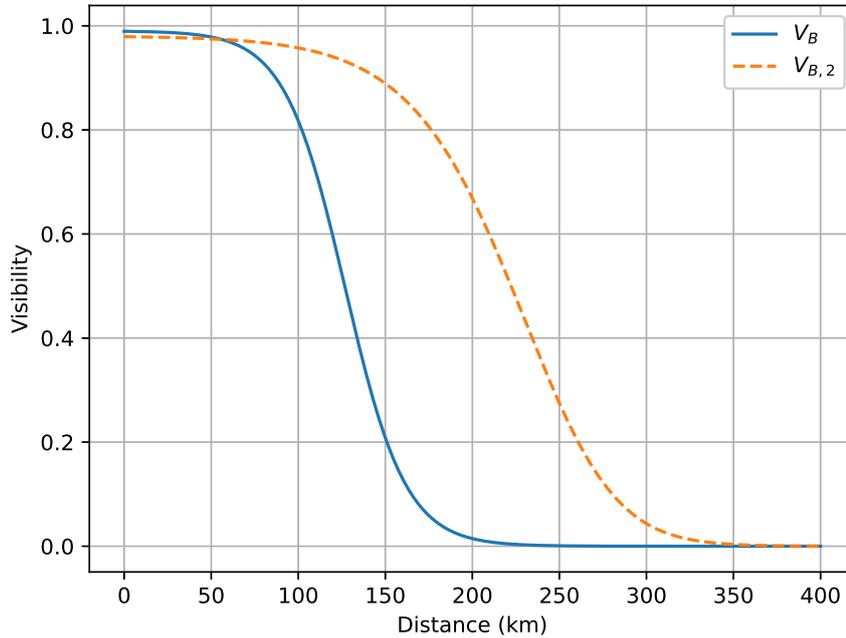
Figure 3.12: Comparison of V_B and $V_{B,2}$

Now we can consider splitting our communication into 3 parts, and then we will generalize to N parts. To do that, we use quantum teleportation (see subsection)

Let denote the intermediaries of the transmission by Charlies. The first intermediary is Charlie-1 and the second is Charlie-2.
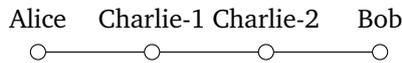

Figure 3.13: Key exchange with $N = 3$

Now let's suppose the following:

1. Charlie-1 emits a pair of entangled particles, one is sent to Alice and the other to Charlie-2;

2. Charlie-2 teleports his particle to Bob;

3. Alice and Bob have a pair of entangled particles. They can proceed as explained in the previous parts.
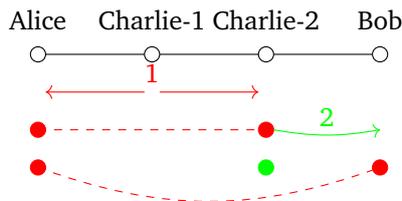
Figure 3.14: Key exchange with $N = 3$

Now, as before, we want to know Bob's visibility. First, the exchange between Alice and Charlie-2 must go well. Also, to do the quantum teleportation, Charlie-2 and Bob must share an entangled pair which is equivalent to one section, resulting in the following results.

We are not capable of qubit teleportation with 100% efficiency. Using linear optics, it is not however possible to do that and the teleportation fails with probability $\frac{1}{2}$ [43]. So we now have

$$p_{signal,3} = \frac{1}{2}\left(F^{\frac{1}{3}}\eta V(1-D)\right)^3 \frac{1}{2} \tag{3.64}$$

We denote

$$p_3 = \left(F^{\frac{1}{3}}\eta + \left(1 - F^{\frac{1}{3}}\right)2D\right)(1-D) \tag{3.65}$$

We would expect to have

$$p_{total,3} = \frac{1}{2}p_3^3 \tag{3.66}$$

like for the 1 and 2 sections cases but in fact, this is not the case. We indeed have the p_3 probability for the section from Charlie-1 to Alice. In the setup of the article, they suppose that the BELL measurement is made using a beam splitter, that combines the photons from the 2 fibers, and then, the BELL states are distinguished by the time of arrival of the photons. 2 BELL states will give photons arriving at different times (one of the BELL states gives different times and different sides while the other gives different times and same side) and the two last BELL states give the same time and the same side, and so the 2 photons arrive at the same detector half of the time. In this case, as only one detection is recorded, this bit don't go in the sifted key, unless a dark count was detected on another detector. Hence, we remove from p_3^2 the probability $\frac{1}{2}(F^{\frac{1}{3}}\eta)^2(1-D)^2$, which is the probability of having the two photons pass the fiber and having no dark count, divided by 2 to have half of the time, and then we add $\frac{1}{2}(F^{\frac{1}{3}}\eta)^2(1-D)^2 \times 2D$. And hence

$$p_{total,3} = \frac{1}{2}p_3\left(p_3 - \frac{1}{2}(1-2D)(F^{\frac{1}{3}}\eta)^2(1-D)^2\right) \tag{3.67}$$

Hence,

$$V_{B,3} = \frac{p_{signal,3}}{p_{total,3}} = \frac{\left(F^{\frac{1}{3}}\eta V(1-D)\right)^3}{2p_3\left(p_3^2 - \frac{1}{2}(1-2D)(F^{\frac{1}{3}}\eta)^2(1-D)^2\right)} \tag{3.68}$$

For 4 sections we can use 2 EPR sources and one entanglement swapping resulting in 1 BELL test.
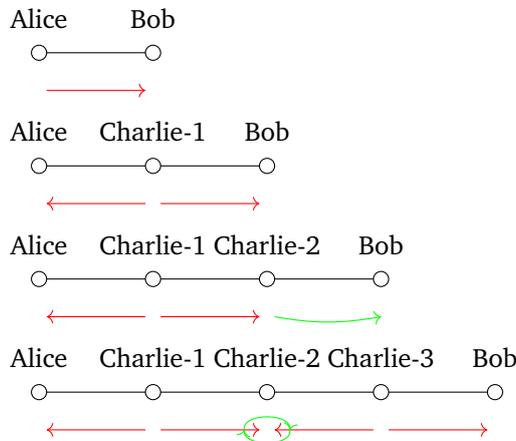
Figure 3.15: Schemes with 1,2,3 and 4 sections (entanglement swapping for 4 sections at Charlie-2)

We can generalize this to n sections. We will $\left\lfloor\frac{n-1}{2}\right\rfloor$ BELL tests and hence

$$p_n = \left(F^{\frac{1}{n}} \eta + \left(1 - F^{\frac{1}{n}} \right) 2D \right) (1 - D) \tag{3.69}$$

$$p_{signal,n} = \frac{1}{2} \left(\frac{1}{2} \right)^{\left\lfloor \frac{n-1}{2} \right\rfloor} \left(F^{\frac{1}{n}} \eta V (1 - D) \right)^n \tag{3.70}$$

$$p_{total,n} = \frac{1}{2} p_n^{2 - 2 \left\lfloor \frac{n-1}{2} \right\rfloor} \left(p_n^2 - \frac{1}{2} (1 - 2D)(F^{\frac{1}{n}} \eta)^2 (1 - D)^2 \right)^{\left\lfloor \frac{n-1}{2} \right\rfloor} \tag{3.71}$$

$$V_{B,n} = \frac{p_{signal,n}}{p_{total,n}} = \frac{\left(\frac{1}{2} \right)^{\left\lfloor \frac{n-1}{2} \right\rfloor} \left(F^{\frac{1}{n}} \eta V (1 - D) \right)^n}{p_n^{2 - 2 \left\lfloor \frac{n-1}{2} \right\rfloor} \left(p_n^2 - \frac{1}{2} (1 - 2D)(F^{\frac{1}{n}} \eta)^2 (1 - D)^2 \right)^{\left\lfloor \frac{n-1}{2} \right\rfloor}} \tag{3.72}$$

Bob's visibility for different values of n and for a distance between 0 and 1000km is plotted below:
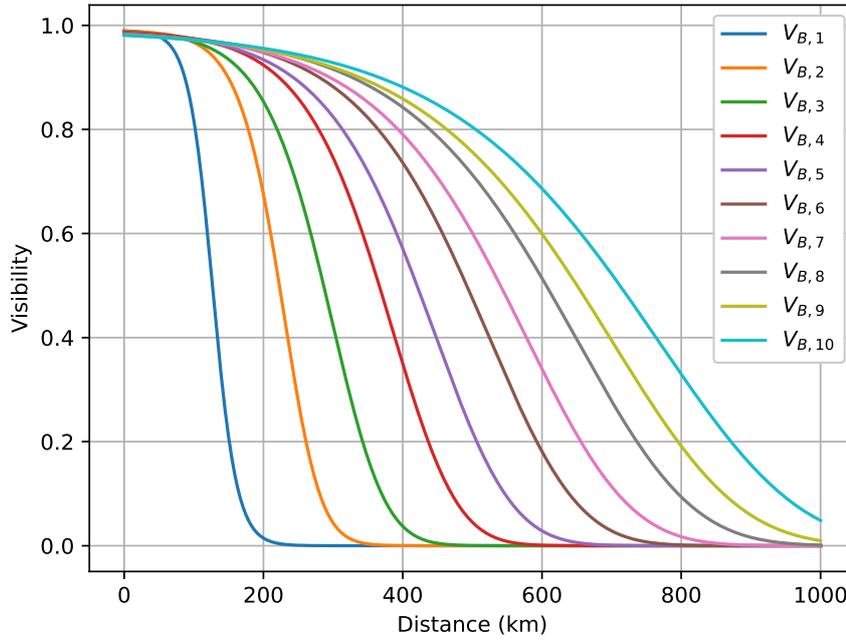

Figure 3.16: Comparison of $V_{B,n}$ for n varying from 1 to 10

Note that imperfect entanglement or imperfect quantum teleportation/entanglement swapping introduce noise and so we cannot split our channel into an arbitrarily large number of sections as the ratio signal to noise will also tend to zero. There is a way, using quantum memory and entanglement purification, to allow an arbitrarily large number of sections and arbitrarily long communication channels. This is referred to as quantum repeaters and was historically the first to be proposed, but are more complicated and less attainable in the foreseeable future.

3.4 Device Independent Quantum Key Distribution

In section 3.2, we made the assumption that the measurement apparatus owned by Alice and Bob could be trusted. The aim of Device Independent Quantum Key Distribution is to design an unconditionally secure protocol without making any assumptions on the measurement devices. They are seen as black boxes that produce a classical output from a quantum and classical input

(the classical input is the choice of the basis). The most extreme case, where the measurement apparatus has been created by the eavesdropper Eve is envisaged.

DIQKD is also used when referring to designing a protocol where the source cannot be trusted.

In [44], which is one of the first paper to talk about device independent quantum key distribution, and which proposed the first implementation of such a protocol, proposed a proof to see why classic QKD is not secure. The basic idea is that, if the measurement apparatus and the source cannot be trusted, they can be in a larger space than a two-qubit space. They could be in a four-qubit space, where the chosen state and chosen measurement operators reproduce the correlations (i.e. totally correlated when Alice and Bob choose the same basis, and uncorrelated otherwise) but the extra space allows Eve to get a tripartite state and to get information.

The basic idea of DIQKD protocols is to test the measurement devices with BELL's inequality. The `Ekert91` protocol makes use of BELL's inequality, and some DIQKD protocols are in fact modified versions of the `Ekert91` protocol. For instance, a protocol that was proposed in [45] and further analysed in [41] is a modified version of `Ekert91` and it goes as follows: Alice and Bob share a Quantum Channel that can be eavesdropped and a classical channel that can be passively eavesdropped. Alice chooses between three possible orientations for measurement: a_0, a_1, a_2, and Bob chooses from two possible orientations: b_1, b_2. An EPR source emits particles that are sent to both Alice and Bob. When Alice chooses a_0 and Bob chooses b_1, the two measurements are perfectly correlated. When other bases are chosen, the results are uncorrelated. This protocol is a bit more efficient than EKERT's one as fewer bases are used. Here $\frac{4}{5}$ of the results are uncorrelated whereas it would have been $\frac{7}{9}$ for EKERT's protocol.

Then the results of measurement when the bases a_1, q_2, b_1, b_2 are used to estimate

$$S = E(a_1, b_1) + E(a_1, b_2) + E(a_2, b_1) - E(a_2, b_2) \tag{3.73}$$

As in Ekert's protocol, the rules of quantum mechanics say that

$$S = 2\sqrt{2}p \tag{3.74}$$

(the state in the quantum channel is $\boldsymbol{\rho} = p\,|\Phi^+\rangle\,\langle\Phi^+| + (1-p)\frac{1}{4}$).

And for classically correlated data, we should find

$$S \leq 2 \tag{3.75}$$

(Note that BELL's inequality is not exactly the same as before, because different angles are chosen).

DIQKD is impossible if the value of S is below the classical limit (i.e. $p \leq \frac{1}{\sqrt{2}}$).

In [45], the protocol was proven to be secure against individual attacks from an eavesdropper that is limited only by the non-signaling principle (which basically states that the measurement of an entangled state cannot transfer information to another observer), and in [41], this protocol was proven secure against collective attacks (Eve applies the same attack to Alice's and Bob's systems).

In the paper, they discussed loopholes in BELL experiments and DIQKD, and especially:

- the locality loophole: no information should travel between the moment one party made the measurement and the other party making the measurement. This means that the measurements should be performed sufficiently far apart and fast, and the basis' choice should be truly random. This loophole is in fact only a problem for BELL experiments and can be overcome in the context of DIQKD by ensuring that Alice's and Bob's physical locations are isolated and that no signal/photon other than the ones needed for the protocol travel from Alice to Bob;

- the detection loophole: detectors must have sufficient detection efficiency to perform BELL tests and DIQKD. According to [41], for the BELL's inequality to be valid it is needed to have $\eta > 82.8\%$. This is a real problem for DIQKD as the measurement apparatus is given by an

untrusted party. A solution proposed in the paper is to use an experiment that is immune to the detection loophole [46].

In 2020, a slight modification of this protocol was proposed to make it more robust and seems to look like EKERT's protocol [47]. Instead of using 1 possibility to have correlated outcomes, there are proposing a protocol, where Alice has the choice between 2 orientations and Bob has the choice between 4 orientations, and the 2 orientations of Alice are in the set of Bob's orientations. In this way, there are 2 possibilities that give correlated outcomes. In the first proposal of DIQKD, a potential eavesdropper would know which orientation gives the correlated outcomes and can optimise here attack knowing this fact, i.e. maximise the amount of information she gained when this set of measurements is chosen from Alice and Bob. In the other case, there are two possibilities, as the two measurements can both maximise Eve's gain of information, there will at least one that will not maximise Eve's gain of information. This protocol was proven to be secure against collective attacks.

We won't discuss it in this paper, but there is also research to do other quantum cryptographic tasks device independent including the tasks that we will see in the following chapters.

Chapter 4

Public key, digital signatures and fingerprinting

In this chapter, we examine how the law of Quantum Mechanics can help to design unconditionally secure public crypto-systems : public-key, fingerprinting and digital signatures.

4.1 Digital fingerprints

4.1.1 Classical fingerprinting

A fingerprint, or digest, or checksum, or hash of a message or string is a smaller message or string produced by a cryptographic hash function. This cryptographic hash function is a function that is one way (see the discussion in the next section but it is basically a function that is easy to compute and hard to invert) and has ideally the following properties (see [48]):

1. A given message has always the same hash;

2. It is impossible to compute the original message from its hash (pre-image resistance);

3. The hash is unique, meaning that two different messages have different hash (2nd pre image resistance). Also, given a hash value, it is not possible to find a message such that its hash is the given hash value;

4. Two different messages should have hashes that differ significantly (near-collision resistance);

5. A small change in the message should result in a significant change in the hash (non-correlation and avalanche effect);

6. Even if the message can be any size, the hash has always the same size.

Some cryptographic hash functions may have slightly-less strong properties in practice.

Today, there are several cryptographic hash functions that are used: md5, the SHA family, or BLAKE (2 or 3), the third version being very recent (January 2020).

Let's take the md5[1] scheme, for example. It was designed in 1991 by Ronald RIVEST and was standardised in RFC1321 [49].

It is a scheme that produces a fingerprint of 128 bits for any given message. It is today considered as insecure because some properties that are given above were proved to be not fulfilled but the scheme is still widely used.

Let's take the message

---

[1]MD stands for Message Digest

IMPERIAL COLLEGE LONDON

The fingerprint produced by the `md5` algorithm is

1ba7043ffcc86c417c072aa74d649202

If we introduce a slight modification in the string, for instance

IMPERIAL COLLGE LONDON

we obtain a totally different hash:

242de8694a5ffac6bd993e12322f8d8a

(the digests are printed in hexadecimal format).

If we compare the two results byte to byte, we find that they have no bytes in common, even with all permutations of bytes allowed. If we compare the two digests bit to bit, we find that they have 59 bits in common over 128 (two random strings would have approximatively $128/2 = 64$ bits in common), so the two digests appear to be decorrelated.

4.1.2 Application of such fingerprints

Fingerprints have several applications, that we try to list and explain below:

- Ensuring the integrity of a message. If you receive a large message and you want to ensure that there were no errors during transmission or that nobody tricked you by making you download a message that only looks like the legitimate message. You could calculate the hash and your own and compare it to a publicly available hash. For example, if you go to the Arch Linux download page, https://www.archlinux.org/download/, you will find 4 different checksum, one being the `md5` checksum: cd918e38b3d468de98c1a523990500ef[2]. This is useful to check that you ave indeed the right file (as they have several distribution servers, one file may have been corrupted for example, or could be corrupted during download);

- Storing password. Instead of storing the plain-text password in databases, that have disastrous consequences when the database is compromised, it is possible to store the hash of the password. Each time a user inputs a password, you calculate the hash and compare it to the hash in the database. If they are the same, the user has entered the good password. This method is no longer considered secure because of the so-called rainbow tables (they are tables of correspondence between commonly used passwords and their hash, which allow malicious parties to just do a search in a list to find a password). The recommended method for storing the password is salted-hashing.

- Produce a more readable verifier for humans. `PGP` keys are based on a trusted network. In order to construct this trust network, people gather at Key Signing Parties (KSP) where you verify that a person does indeed the key he claims to hold (by sending him an encrypted message and verify he can decrypt it) and verify he is the man he claims to be (by verifying an identity card with a photograph). In the beginning, to verify that you are signing and trusting the good key, you compare the fingerprints of the keys.

  The fingerprint of my `PGP` key is

  438C 07EC 20B6 45D0 5BBE 4C83 DC24 C578 7C94 3389

  and my complete public `PGP` key can be found at https://nanoy.fr/media/yoann_pietri.asc but has over 15500 characters it;

---

[2]This checksum was extracted on August, 22nd 2020. A different checksum could be available at the download page at later times.

- Fingerprints can be used as file identifiers. For example, `git` the versioning software used hashes to uniquely identify files and commits. In the same idea, it can be used to solve the Two Generals' problem by creating a unique identifier for a message that needs to be only interpreted once [50].

- Verify that a shared secret is the same. Imagine that Alice and Bob want to verify that they share the same secret string. They can do it by comparing the fingerprint of their respective strings;

- Fingerprints are used in digital signature schemes. We speak of them in section 4.3.

In the end, such identifiers are very important and there are used in lots of different situations.

4.1.3 Quantum fingerprinting

In analog to the **CNOT** gate, we can define the Controlled-SWAP or **CSWAP** or FREDKIN gate, which is a gate that takes 3 qubits as input and outputs 3 qubits, where the second and third qubit are swapped if the first qubit is $|1\rangle$. The first qubit is not changed.

$$\begin{aligned} \textbf{CSWAP}(|0\rangle |\phi\rangle |\psi\rangle) &= |0\rangle |\phi\rangle |\psi\rangle \\ \textbf{CSWAP}(|1\rangle |\phi\rangle |\psi\rangle) &= |1\rangle |\psi\rangle |\phi\rangle \end{aligned} \tag{4.1}$$

and if $|c\rangle = \alpha |0\rangle + \beta |1\rangle$ is a general control qubit

$$\textbf{CSWAP}(|c\rangle |\phi\rangle |\psi\rangle) = \alpha |0\rangle |\phi\rangle |\psi\rangle + \beta |1\rangle |\psi\rangle |\phi\rangle \tag{4.2}$$

This can be represented by the matrix

$$\textbf{CSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \left(\begin{array}{c|c} \mathbb{1}_4 & \mathbf{0}_4 \\ \hline \mathbf{0}_4 & \text{SWAP} \end{array} \right) \tag{4.3}$$

in the $\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}$ basis.

In quantum circuits, we represent it by
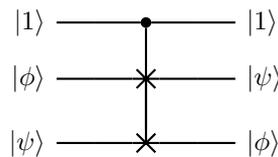
Figure 4.1: Representation of the **CSWAP** gate

    The SWAP test is a test that allows us to see how two states are different, i.e. to estimate the overlap $|\langle\psi|\phi\rangle|^2$.

    Let's consider two states $|\phi\rangle$ and $|\phi\rangle$ and the following quantum circuit:
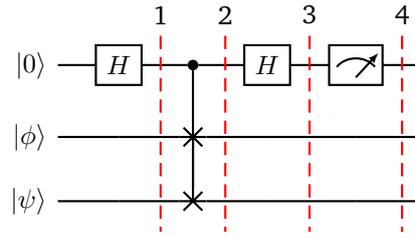
Figure 4.2: SWAP test circuit

Just before, measurement, in step 3, the system is in the state

$$\underbrace{(\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1})}_{\text{2 to 3}} \underbrace{(\textbf{CSWAP})}_{\text{1 to 2}} \underbrace{(\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1})}_{\text{Init to 1}} \underbrace{|0\rangle \, |\phi\rangle \, |\psi\rangle}_{\text{Init}} \tag{4.4}$$

and with some computations:

$$
\begin{aligned}
(\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1})(\textbf{CSWAP})(\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1}) |0\rangle \, |\phi\rangle \, |\psi\rangle &= \frac{1}{\sqrt{2}} (\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1})(\textbf{CSWAP})(|0\rangle \, |\phi\rangle \, |\psi\rangle + |1\rangle \, |\phi\rangle \, |\psi\rangle) \\
&= \frac{1}{\sqrt{2}} (\boldsymbol{H} \otimes \mathbb{1} \otimes \mathbb{1})(|0\rangle \, |\phi\rangle \, |\psi\rangle + |1\rangle \, |\psi\rangle \, |\phi\rangle) \\
&= \frac{1}{2} (|0\rangle \, |\phi\rangle \, |\psi\rangle + |1\rangle \, |\phi\rangle \, |\psi\rangle + |0\rangle \, |\psi\rangle \, |\phi\rangle - |1\rangle \, |\psi\rangle \, |\phi\rangle)
\end{aligned}
$$

and hence, the final state before the measurement is

$$\frac{1}{2} \Big(|0\rangle \left(|\phi\rangle \, |\psi\rangle + |\psi\rangle \, |\phi\rangle \right) + |1\rangle \left(|\phi\rangle \, |\psi\rangle - |\psi\rangle \, |\phi\rangle \right) \Big) \tag{4.5}$$

Hence the probability of measuring 1 as an outcome is

$$
\begin{aligned}
p &= \frac{1}{2} (\langle\phi| \, \langle\psi| - \langle\psi| \, \langle\phi|) \frac{1}{2} (|\phi\rangle \, |\psi\rangle - |\psi\rangle \, |\phi\rangle) \\
&= \frac{1}{4} (1 - |\langle\psi|\phi\rangle|^2 - |\langle\phi|\psi\rangle|^2 + 1) \\
&= \frac{1}{2} - \frac{1}{2} |\langle\phi|\psi\rangle|^2
\end{aligned} \tag{4.6}
$$

As for the probability of measuring 0, it is

$$1 - p = \frac{1}{2} + \frac{1}{2} |\langle\phi|\psi\rangle|^2 \tag{4.7}$$

It means that if the two states are equal to each other $\langle\phi|\psi\rangle = 1$, then the result of the measurement will only be 0.

If you repeat the SWAP test n times, and store each result as r_i, the average value of the $r_i's$ should be the expected value of the SWAP test which is p ($(1-p) \times 0 + p \times 1$) i.e.

$$\frac{1}{n} \sum_{i=1}^{n} r_i \simeq \frac{1}{2} - \frac{1}{2} |\langle\phi|\psi\rangle|^2 \tag{4.8}$$

that can be rewritten

$$|\langle\phi|\psi\rangle|^2 \simeq 1 - \frac{2}{n} \sum_{i=1}^{n} r_i \tag{4.9}$$

and in a more formal way

$$| \langle \phi | \psi \rangle |^2 = 1 - \lim_{n \to \infty} \frac{2}{n} \sum_{i=1}^{n} r_i \qquad (4.10)$$

The SWAP test was introduced in a paper called *Quantum Fingerprinting* published in 2001 by BUHRMAN *et al.* [51]. It seems to be the only paper, as this day, treating of the subject of quantum fingerprinting. In this paper, the problem was introduced as follow (it is called simultaneous message passing and it was introduced in [52]): Alice and Bob have each a string of n bits, let denote them by x and y and the goal is, for a third party, called the referee, to get the right value of $f(x, y)$ with

$$f(x, y) = \begin{cases} 1 & \text{if} \quad x = y \\ 0 & \text{if} \quad x \neq y \end{cases} \qquad (4.11)$$

i.e. to know whether Alice and Bob share the same string or not. Of course, this could be done if Alice and Bob send their respective string to the referee, but this method is nor optimal nor private.

The basic idea is the following:

1. Apply an error correction code to the strings;

2. Build a quantum fingerprint of the string;

3. Send the fingerprint to the referee;

4. The referee performs the SWAP test to know whether the fingerprints are the same or not.

Let's try to understand deeper and especially why we need an error correction code.

4.1.4 Error correction codes

This subsection is here to recap about error correction codes. It will cover simple properties in order to understand the fingerprinting protocol. We consider error correction codes on strings of bits, and we follow standard notations that may not agree with [51].

We will only consider linear block codes here, meaning that a whole message is split into smaller messages, each of the same size. Each message is then transformed into a longer message called a block.

An error correction code C is a map

$$C : \{0, 1\}^k \to \{0, 1\}^n \qquad (4.12)$$

with k, and n being integers. k is called the message length and it is the number of bits per block. n is called the block size. C is required to be injective (two different messages can't have the same code block).

The rate R is the ratio

$$R = \frac{k}{n} \leq 1 \qquad (4.13)$$

The quantity of information coded into one block increases as R increases.

Now let's define the minimum distance of a code. For that, we need the concept of HAMMING distance. We consider two strings of bits, of the same length. For each position of bit, we can have either the same bit on the two strings or two different bits. The HAMMING distance is the number of differences.

$$\begin{array}{c} 0110110 \\ 0101010 \end{array} \qquad (4.14)$$

From the example above, the HAMMING distance is the number of bits in red, so it is 3.

The HAMMING distance between two strings s_1 and s_2 is denoted $\Delta(s_1, s_2)$ and we have

$$0 \leq \Delta(s_1, s_2) \leq \text{length}(s_1) \tag{4.15}$$

where the left equality occurs when $s_1 = s_2$ and the right when s_1 is the exact opposite, bit to bit, of s_2.

Now we define the minimal distance d of an error correction code C by

$$d = \min_{\substack{s_1, s_2 \in \{0,1\}^k \\ s_1 \neq s_2}} \Delta(C(s_1), C(s_2)) \tag{4.16}$$

i.e. the minimal HAMMING distance between two different code blocks. As C is injective, d is at least one.

Ok let's take an example. We consider a very simple error correction code which is a parity code. We consider $k = 1$ and $n = 3$ and the following map:

$$\begin{aligned} p(1,3) &: \{0,1\} \to \{0,1\}^3 \\ p(1,3)(0) &= 000 \\ p(1,3)(1) &= 111 \end{aligned} \tag{4.17}$$

We are just repeating the bit three times. This will make the whole message much longer (3 times longer actually) but more resistant to errors if we considerer that the probability of two errors happening during the transmission is very less than the probability of one error happening.

Indeed considerer the case, where the initial message is 0, then the transmitted block is 000 and during the transmission, an error makes the second bit flip. After transmission, the block is 010, and assuming that only one error occurred, we find that the initial message was 0. If two or three errors occur, we will guess the wrong initial message.

In this error correction code there are only two codes: 000 and 111 and the Hamming distance between this two codes is 3. Then we deduce that for this code, $d = 3$.

4.1.5 Back to the fingerprinting protocol

Now let's consider Alice and Bob with their respective strings x and y, each of length k. We suppose we have an error correction code

$$C : \{0,1\}^k \to \{0,1\}^n \tag{4.18}$$

with n bigger than k and we denote by d the minimal distance of C.

For i between 1 and n we denote by $C_i(x)$ the i-th bit of $C(x)$.

Now, we define the quantum fingerprint

$$|h_x\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |C_i(x)\rangle \tag{4.19}$$

Now let's compute the overlap $\langle h_y | h_x \rangle$

$$\langle h_y | h_x \rangle = \left(\frac{1}{\sqrt{n}} \sum_{j=i}^{n} \langle j | \langle C_j(y) | \right) \left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} | i \rangle | C_i(x) \rangle \right)$$

$$= \frac{1}{n} \sum_{i,j=1}^{n} \langle j | i \rangle \langle C_j(y) | C_i(x) \rangle$$

$$= \frac{1}{n} \sum_{i,j=1}^{n} \delta_{ij} \langle C_j(y) | C_i(x) \rangle$$

Hence

$$\langle h_y | h_x \rangle = \frac{1}{n} \sum_{i=1}^{n} \langle C_i(y) | C_i(x) \rangle \tag{4.20}$$

But $C_i(x)$ is either 0 or 1 and same for $C_i(y)$ and then the overlap $\langle C_i(y)|C_i(x)\rangle$ has only two possible values:

$$\langle C_i(y) | C_i(x) \rangle = \begin{cases} 1 & \text{if} \quad C_i(x) = C_i(y) \\ 0 & \text{if} \quad C_i(x) \neq C_i(y) \end{cases} \tag{4.21}$$

Hence, we have

$$\sum_{i=1}^{n} \langle C_i(y) | C_i(x) \rangle = \text{number of bit-to-bit equalities between } C(x) \text{ and } C(y)$$
$$= n - \text{number of bit-to-bit differences between } C(x) \text{ and } C(y) \tag{4.22}$$

and remembering the definition of the HAMMING distance given above

$$\langle h_y | h_x \rangle = \frac{1}{n} (n - \Delta(C(x), C(y))) = 1 - \frac{\Delta(C(x), C(y))}{n} \tag{4.23}$$

This is an occasion to verify that fingerprints are unique. Let's imagine that $|h_x\rangle = |h_y\rangle$, then we have $\langle h_y | h_x \rangle = 1$ and hence we deduce $\Delta(C(x), C(y)) = 0$ and then $C(x) = C(y)$. Remembering that error corrections code are injective, we deduce $x = y$ and then two different strings will have different fingerprints.

$$x = y \Leftrightarrow |h_x\rangle = |h_y\rangle \tag{4.24}$$

Considering equation 4.23 and the definition of the distance d of C, we infer that, when $x \neq y$

$$\langle h_y | h_x \rangle \leq 1 - \frac{d}{n} \tag{4.25}$$

But we have a way to estimate $\langle h_y | h_x \rangle$ with the SWAP test. If we perform the test on the fingerprints (when $x \neq y$):

$$p \geq \frac{1}{2} - \frac{1}{2} \left(1 - \frac{d}{n} \right)^2 \tag{4.26}$$

This means that when $x = y$, the probability of obtaining 1 is 0 and when $x \neq y$, the probability of obtaining 1 is at least $\frac{1}{2} - \frac{1}{2} \left(1 - \frac{d}{n} \right)^2$.

Hence, the SWAP test with one copy of the fingerprints with the output

"Yes the two strings are identical" if the SWAP test result is 0

"No, the two strings are different" if the SWAP test result is 1

will always give a correct result if $x = y$ and will give a wrong result with probability at most $\frac{1}{2} + \frac{1}{2}\left(1 - \frac{d}{n}\right)^2$ if $x \neq y$.

Now consider the case where Alice and Bob send 2 copies of their fingerprints to the referee. Then we consider the two-round swap test:

"Yes the two strings are identical" if the two SWAP test results are 0

"No, the two strings are different" if one the two SWAP test result is 1

will always gave a correct result if $x = y$ and will fail, in the case $x \neq y$ it the two results of the SWAP tests are 0, i.e., with a probability of at most

$$\left(\frac{1}{2} + \frac{1}{2}\left(1 - \frac{d}{n}\right)^2\right)^2 \tag{4.27}$$

More generally, if Alice and Bob send k copies of their fingerprints, the SWAP test

"Yes the two strings are identical" if the k SWAP test results are 0

"No, the two strings are different" if one the k SWAP test result is 1

have the following probability of failure:

$$\begin{cases} 0 & \text{if} \quad x = y \\ \left(\frac{1}{2} + \frac{1}{2}\left(1 - \frac{d}{n}\right)^2\right)^k & \text{if} \quad x \neq y \end{cases} \tag{4.28}$$

This probability can be made arbitrarily smaller than ε for any $\epsilon > 0$ if we allow enough copies of fingerprints.

In fact, in order to have

$$\left(\frac{1}{2} + \frac{1}{2}\left(1 - \frac{d}{n}\right)^2\right)^k < \varepsilon \tag{4.29}$$

we would require

$$k > \frac{1}{a}\log\left(\frac{1}{\varepsilon}\right) \tag{4.30}$$

where $a = -\log\left(\frac{1}{2} + \frac{1}{2}\left(1 - \frac{d}{n}\right)^2\right) > 0$.

4.2 Public key

In this section, we first review the goals of public-key encryption and how it is implemented today. Then we proceed to quantum versions of public-key cryptosystems.

4.2.1 Goal of public-key encryption

We already explained in the introduction section the basics of public-key encryption, also referred to as asymmetric cryptography.

Each people can generate a private secret key from a key generation program, usually using or generating a large random number as input. The key generation program can also generate a public key using the private key.

The private key must stay secret and is usually encrypted using a password and standard encryption. On the other hand, the public key can be distributed to the open world without compromising the security of the secret key.

The public key can be used to encrypt a message. On the other hand, the private key is used to decrypt the message. The private cannot be recovered from the public key and the public key is insufficient to decrypt a message.

There are two main applications of public-key encryption:

- Exchange encrypted data;

- Use it as a digital signature (authentication method) to verify the authenticity of the sender of a message for instance.

Public-key encryption has many advantages (easy to use and understand, scalable, secure) but is computationally heavy, especially when there is a need to encrypt large files. To encrypt a large amount of data (a document, an email, etc...), it has to do a lot more computations than using symmetric keys. Often, protocols are then hybrid: they exchange a one-time secret key (i.e. a secret key that will only be used for this exchange) and exchange the key using public-key encryption.

Let's suppose, Alice wants to send an email to Bob. This goes like this:

1. Alice gets Bob's public key (on public-key open servers for example);

2. Alice generates a completely new and random symmetric key;

3. Alice encrypts the email using the symmetric key;

4. Alice encrypts the symmetric key using Bob's public key;

5. Alice sends the encrypted email and the encrypted symmetric key to Bob.

To decrypt the email, Bob performs the following operations:

1. He decrypts the symmetric key using his private key;

2. He decrypts the email using the symmetric key.

This kind of hybrid cryptosystems is used for example with the PGP (to exchange encrypted emails), SSH (to authenticate on remote servers), and SSL/TLS (widely used for example for web browsing) protocols.

If we forget about the email, we see that au public-key encryption algorithm can allow two parties to share a common symmetric key.

There are some security issues with the use of public-key encryption.

If the size of the key is too small, brute force attacks can be used. But as long as large keys are used and that quantum computers are not available, there is no way to deduce the private secret key in a reasonable amount of time.

One way to cheat is a man-in-the-middle attack or authenticity issues when the public key is exchanged. Imagine that Mr. X posts a public key claiming it to be Mr. Y's public key. Then Mr. X can read all encrypted messages sent to Mr. Y (and Mr. Y can't read the messages).

4.2.2 Quantum public-key

Classically, public cryptosystems are implemented using one-way functions. They are functions such that they are "easy" to compute, and "hard" to invert, and this is basically the case for integer factorisation. It means that you can compute the function in polynomial time, but that there is a negligible probability that a polynomial-time algorithm succeeds in inverting the function.

In fact, what is used in classical cryptosystems are trapdoor one-way functions. They are functions that are "easy" to compute and "hard" to invert unless you have additional information. Hence, you can encrypt a message using this function and no one can invert it with high probability unless he has the trapdoor information, i.e. the secret key.

We can define, following [53, 54], similar concepts in Quantum Cryptography:

Definition (Quantum One-Way function): A quantum one way function (sometimes abbreviated quantum OWF, QOWF or QOW) $f : x \mapsto |\phi_x\rangle$ is a function that takes an input string and outputs a quantum state such that it is "easy" to compute but "hard" to invert, in the following sense:

- We can find a quantum circuit that given the input x, compute the output $|\phi_x\rangle$ in a polynomial time;

- Given $|\phi_x\rangle$, there is no polynomial quantum algorithm that recovers x with a non-negligible probability, or written in a more mathematical sense:

$$\mathbb{P}(f(y) = f(x)) < \frac{1}{n^c} \tag{4.31}$$

  y being the output of the polynomial algorithm, Pr the probability, n the size of x, and c any integer.

Note that even if the basic idea of a quantum one-way function is always the same, the exact definition varies between publications.

Now, as in the classical case, a quantum one-way function is not very useful because neither eavesdroppers nor legitimate users can decrypt messages. We need to keep the hard inversion for non-legitimate users and make the inversion easy for legitimate users.

Definition (Quantum Trapdoor One-Way function): A quantum trapdoor one way function $f : x \mapsto |\phi_x\rangle$ is a function that takes an input string and outputs a quantum state such that it is "easy" to compute but "hard" to invert, in the following sense:

- We can find a quantum circuit that given the input x, compute the output $|\phi_x\rangle$ in a polynomial time;

- Given $|\phi_x\rangle$, and additional information called trapdoor information, we can recover the input x in polynomial time;

- Given $|\phi_x\rangle$, and without the trapdoor information mentioned above, there is no polynomial quantum algorithm that recovers x with a non-negligible probability, or written in a more mathematical sense:

$$\mathbb{P}(f(y) = f(x)) < \frac{1}{n^c} \tag{4.32}$$

  y being the output of the polynomial algorithm, Pr the probability, n the size of x, and c any integer.

Now the whole challenge is to find some practical quantum trapdoor one-way functions.

Now before examining the candidates for such functions, let's define a quantum public-key encryption scheme:

Definition (Quantum Public-Key Encryption scheme): A quantum public key encryption is composed of three quantum functions, that we can call G, E and D. E can be seen as quantum trapdoor one way function and D its inverse:

- G is a function that takes the size of the key n and produces a secret key s and a public key p;

- E is the encryption function, that takes as an input p and a message m and outputs an encrypted message $E(p, m)$;

- G is a decryption function that takes e and an encrypted message $E(p, m)$ and outputs m.

First, we will see general ideas on quantum public-key cryptosystems, then research that has been done on the subject. I cite a slide presentation from Daniel GOTTESMAN for this first part [55] (who was one of the first working on the subject):

- The public key will be a quantum state, meaning that a quantum network infrastructure is needed to send the public key;

- Another implication of the fact that the public key is a quantum state is the fact that quantum memory may be needed, at least during the time needed to encrypt data. The owner of the public key doesn't require quantum memory as the key can be regenerated from the private key (which is classical), and the public key can be sent to the sender just before encryption to minimise the time needed for quantum memory;

- Another remark that we can draw from the public key being a quantum state is the fact that the information that can be obtained about the private key contained in the public key is bounded, by virtue of Holevo's theorem. However, the more copies of the public key in circulation, the more information on the private key a malicious party has. This means that there is a maximum number of copies of the public key that we can distribute before it is needed to regenerate one. This is a key difference with classical public schemes with publishing a new copy of the public key does not provide any supplementary information;

- Quantum public key does not seem more efficient than classical public-key cryptosystems(and even seems less efficient) but will be more secure to a post-quantum attacker.

In [55], a sketch of a quantum public-key scheme is provided:

Alice starts by generating a random secret key k. This key is used to generate a unitary operator U_k, but the knowledge of U_k does not give the full knowledge of k (easy to compute, hard to invert).

Then the public key is given by

$$(\mathbb{1} \otimes U_k)(|00\rangle + let11) \tag{4.33}$$

This state is sent to Bob through a quantum channel (even if they are not discussed here, keep in mind the authentication problem).

Now let's say that Bob went to send the state $|\phi\rangle$ (it may contain classical data encoded in a quantum way but this is not the issue we address here). To encrypt, Bob teleports the state using the public key. It will obtain one of four results, giving one of the Pauli matrices σ. Bob ends up with the encrypted state

$$U_k \rho |\phi\rangle \tag{4.34}$$

in the second register of the state that was previously the public key. Next, Bob sends the state to Alice, along with the Pauli matrix that was obtained during the teleportation and Alice can decrypt the data, first by applying U_k^{-1} then the Pauli operator σ (remember that they square to $\mathbb{1}$).

Note that the copy of the public key is destroyed in the process.

There were somme additional developments in the subject [54, 56, 57].

Let's see an example of a public key scheme, that was described in [56]:

Alice generates her private key by choosing a random function $F : \{0,1\}^n \to \{0,1\}^n$. She randomly generates $s \in \{0,1\}^n$ and computes $k = F(s)$. k is a n-bit string and we denote $k_1, \ldots k_n$ the bits of k. Alice randomly generates an even n-bit string $i_1, i_2, \ldots i_n$ (i.e. the bits that compose the string sum to 0, modulo 2) and encodes it in the $\mathcal{B}_z$ basis in the state $|i\rangle = |i_1\rangle \otimes \ldots \otimes |i_n\rangle$. Alice applies $\boldsymbol{H^k} = \boldsymbol{H^{k_1}} \otimes \boldsymbol{H^{k_2}} \otimes \ldots \otimes \boldsymbol{H^{k_n}}$ to $|i\rangle$. Here $\boldsymbol{H}$ is the Hadamard gate, when the bit is 0 it stays ine the $\mathcal{B}_z$ basis and change to $\mathcal{B}_x$ when the bit from k is 1. $(s, \boldsymbol{H^k}|i\rangle)$ is a copy of Alice's public key. She sends it to Bob.

Bob receives the public key. He wants to send $m = 0/1$ to Alice. Bob then generates a n-bit string $j_1, \ldots, j_n$ with $j_1 \oplus j_2 \oplus \ldots \oplus j_n = m$. He applies $\boldsymbol{\sigma_y^j} = \boldsymbol{\sigma_y^{j_1}} \otimes \boldsymbol{\sigma_y^{j_2}} \otimes \ldots \otimes \boldsymbol{\sigma_y^{j_n}}$ to $\boldsymbol{H^k}|i\rangle$. Bob sends $(s, \boldsymbol{\sigma_y^j H^k}|i\rangle)$ to Alice.

Alice receives $(s, \boldsymbol{\sigma_y^j H^k}|i\rangle)$ and applies again H^k to $\boldsymbol{\sigma_y^j H^k}|i\rangle$.

Now we are interested in the value of

$$\boldsymbol{H^k \sigma_y^j H^k} \tag{4.35}$$

Let $1 \leq \alpha \leq n$, we are interested in $\boldsymbol{H^{k_\alpha}} \times \boldsymbol{\sigma_y^{j_\alpha}} \times \boldsymbol{H^{k_\alpha}}$. If $j_\alpha = 0$, then this is equal to $\left(H^2\right)^{k_\alpha} = \mathbb{1}$ since $\boldsymbol{H^2} = \mathbb{1}$. If $j_\alpha = 1$, then we distinguish between $k_\alpha = 0/1$. If $k_\alpha = 0$, this is trivially equal to σ_y. Finally, if $k_\alpha = 1$, then we compute $\boldsymbol{H} \times \boldsymbol{\sigma_y} \times \boldsymbol{H} = -\boldsymbol{\sigma_y}$. Hence, we have, up to a sign, the same expression as $\boldsymbol{\sigma_y^j}$. The sign is the number of times when we have $k_\alpha = j_\alpha = 1$ and so

$$\boldsymbol{H^k \sigma_y^j H^k} = (-1)^{j \cdot k} \boldsymbol{\sigma_y^j} \tag{4.36}$$

σ_y will flip the bit, up to a phase and so σ_y^j will flip the bit of $|i\rangle$ every time the bit of j is 1, and hence, the final state is up to a global phase $|i \oplus j\rangle$. As i was chosen to be even, the parity of $i \oplus j$ is m, the one-bit message that Bob wanted to send.

Let's draw some general remarks:

- F is chosen between $(2^n)^{2^n}$ functions. This can be done by generating 2^n random bits;

- The public key is not unique, as opposed to the classical case. It depends on a precomputed secret that will change for each secret key. This will help to reduce the amount of information on F that is given in one copy of the public key;

- Here the quantum one-way trapdoor function is the one that maps j to $\boldsymbol{\sigma_y^j H^k}|i\rangle$. The knowledge of k is needed to recover j;

- This scheme was proven to be information-theoretically secure or ciphertext-indistinguishable under quantum chosen plaintext attack. Without giving a proper definition, here is what it means. Imagine a game where the malicious party, which has access to the public key, gives 2 plaintext bit strings to the owner of the key, who performed encryption on one of the two strings, randomly. The ciphertext is then sent back to the malicious party. We say that the scheme is ciphertext-indistinguishable under quantum chosen plaintext attack if the information owned by the malicious party (the two plain texts) only give him a negligible advantage over a random guess;

- One copy of the public key was needed to encrypt one bit of classical data.

In [56], a scheme based on entanglement was also proposed.

Now before closing, this section, I would like to come back on a point that seems to be one of the key differences between classical and quantum public-key encryption which is the fact that there is a maximal number of copies allowed.

First, we need to emphasize that a malicious party can have two different goals: the first one is to decrypt a message, i.e. recovering the plaintext from the ciphertext, and the second one is to obtain the private key. The accomplishment of the second task immediately makes the first one trivial but might be much more difficult.

In [57], a quantum public-key scheme based on qubit rotations. The basic idea is again the same, Alice generates a private key and a copy of the quantum public key and sends this copy to Bob. Bob makes an interaction between his message and the public key and sends the public key back to Alice, who can apply operations bases on the private key to recover the original message.

In this paper, the authors proved that the VON NEUMANN entropy of the states of τ copies of one public qubit of the key was bounded, and hence by virtue of HOLEVO's bound, the information that Eve could access. It was proven that the bound is

$$I \leq \log_2(\tau + 1) \tag{4.37}$$

It was also proven that the map that takes the private key to generate τ copies of the public key was quantum one way at the condition that $n >> \log_2(\tau + 1)$ where n is a security parameter, that is chosen at the beginning of the protocol.

It was nevertheless proven in the article that even if Eve gains a negligible amount of information on the private key, due to this bound, she could successfully decrypt a message, which is an easier task.

4.2.3 Are quantum public-key cryptosystems really useful ?

As said in the introduction and at the beginning of this chapter, symmetric cryptosystems are more efficient than asymmetric cryptosystems and secure enough if there are used rights. One of the main issues was to distribute the secret key without someone eavesdropping on it. In the past, it was done with guys traveling with a case handcuffed to their hand and, today is it done, as we explained with hybrid cryptosystems, using public keys. But with Quantum Key Distribution, we can distribute secret keys without bothering for public keys. So why are we investigating public keys at all?

First, even if the security of QKD is theoretically secure, there are flaws in practical realisations.

Also as explained before, secret key cryptography does not scale well and if we want to have secure communication between two users, they need a secret key for each message and also a way to distribute the secret key between two end-users, this means having very complex quantum networks, in order to be able to perform QKD end-to-end. On the other hand, quantum public-key cryptography can use classical channels.

4.3 Digital signatures

4.3.1 What are digital signatures ?

A digital signature scheme is a cryptographic scheme that allows a sender to sign a document, or a message and the receiver to verify the signature, verifying that the message was issued by the good person (at least by the detainer of a private key that should be in practice verified, for instance, in Key Signing Parties (see the previous section)) and to verify the integrity of the document or message (that the message was not altered in any way, intentionally or not).

In practice, if Alice wants to send a signed document to Bob:

1. Once the document is finished, Alice generates a signature of the document using her private key;

2. She sends her document and the signature to Bob;

3. Bob verifies that the document and signature match. This will use some additional information (Alice's public key).

4. If the document and signature don't match, then Alice and Bob will proceed to a new exchange. If they match, Bob knows that: the integrity of the message is correct and the sender was in possession of Alice's private key.

Two remarks:

- One might be worried if only the signature is altered. In fact, if the signature is altered, the document and signature won't match and Alice and Bob will proceed to a new exchange anyway. One could be worried that the document and the signature are altered in such a way that there is a match anyway, but that is very not likely (see formal definition);

- This looks like a fingerprint, and for that reason, they are in the same chapter, but there is one key difference: the use of public and private keys. In fingerprinting schemes, any party can run the algorithm that produces the signature. Hence, Eve could place herself between Alice and Bob and replace Alice's message by a completely different one, with a different document and a valid fingerprint. Doing that with a digital signature scheme will only work if Eve had Alice's private key to generate the signature.

This document was signed using a `PGP` key (see appendix A for more detail)

But. If someone gains access to someone else's private key, he could impersonate that person and send a message. Well yes. The private key must at all cost remains secret. Sometimes, it can't be done. That's why, when creating a key, you also generate revoking certificates. They should be kept secret but are less sensitive than the private key. Also, the owner of the private key must keep several copies of these revoking certificates. Thus, if the key is lost or stolen, the original owner revokes the key (by sending the certificates to key servers) and all messages signed or encrypted with this private key from this date should be considered insecure. However note that if this key was used as a public key cryptosystem private key, the thief will be able to decrypt any messages encrypted with the corresponding public key.

One could say that digital signatures are a mix between fingerprinting and public-key encryption.

Also, remember how it is important, for QKD for instance, to have an authenticated quantum channel (even if quantum digital signature may not be a good solution. See [25] for a discussion an authentication of quantum channels for QKD, using a pre-shared secret, that is coupled to the photon used for QKD).

4.3.2 Quantum digital signature

As for public-key encryption, we have a formal definition of what is a quantum digital signature scheme.

> **Definition (Digital signature scheme):** A digital signature schema is composed of 3 polynomial functions (G, S, V):
>
> - G is called the key generator. It generates a private key s and the associate public key p on input 1^n.
>
> - S is the signing function. On an input message m and the private key s, it returns a signature $t = S(m, s)$
>
> - V is the verification function, that takes as input the message m, the public key p and the signature t and outputs 1 if everything is correct and 0 otherwise.

A digital signature scheme (G, S, V) is correct if, for any message m

$$\mathbb{P}\left(V(m, p, S(m, s)) = 1\right) = 1$$
$$s, p = G(1^n) \tag{4.38}$$

In the slides that were cited for public-key encryption [55], there was also some ideas on quantum digital signature. The first statement is that it is impossible to sign an unknown quantum state. There is no equivalent classically as it is always possible to know a message.

We will give an example of a quantum signature scheme, that was designed in 2001 by GOTTES-MAN and CHUANG [58]. The scheme works as follows.

Let's say Alice wants to send a signed bit $b = 0/1$. Alice, and all other parties involved, have at their disposal the same quantum one-way function $f : k \mapsto |f_k\rangle$. Alice start by generating M pairs composed of 2 L-bit private keys: $\{k_0^i, k_1^i\}$ for $1 \leq i \leq M$. Then Alice compute the correspondent public keys $\left|f_{k_0^i}\right\rangle$, $\left|f_{k_1^i}\right\rangle$. We suppose that the keys are then available to each party involved (we will see how there are distributed later).

Then, to send the signed bit, Alice sends the classical message, over a classical channel, that is not authenticated: $(b, k_b^1, b_b^2, \ldots, k_b^M)$. Then the other participants can use the quantum one-way function to compute the public key and verify that the message was indeed sent by someone that possessed the private keys prior to the exchange.

Here, we supposed that everyone has a perfect copy of Alice's public keys, which may not be the case if the channels are noisy. It is possible, in the protocol to define thresholds to accept or refuse a message (and even something in between where the message is accepted but is marked unsafe to be transferred).

After the exchange, Alice discards all the private keys (used or unused).

Now, how Alice sends the copies of the public key. First option: she has an authenticated channel with all the parties involved, in which case, a digital signature scheme isn't really needed. Second option: a key distribution center where a party, which is supposed truthful has an authenticated channel with everyone. Then Alice can send her public keys to the key distribution center, enough copies for all participants, and then the kye distribution center sends the public keys to the parties. The key distribution center may perform swap tests to verify that Alice has indeed sent the same keys for everyone. But in that case, Alice can send a message to the key distribution center through the authenticated, and the message is then sent to the other parties through an authenticated channel. In this case, the distribution center acts more like a certification authority that proves the authenticity of the message.

In any case, it is possible to do quantum digital signatures and there is research on the subject. However, it needs a lot of resources and has common issues with quantum public-key encryption.

Chapter 5

Other quantum components for cryptography

In this last chapter, we review how quantum mechanics can be applied to other cryptographic tasks. First, in section 5.1, we see how we can use quantum mechanics to generate randomness. True randomness, as we saw in this paper is a prerequisite for both classical and Quantum Cryptography. Next, in sections 5.2 and 5.3, we see historic applications of Quantum Cryptography. Finally, we see how we can design quantum voting scheme in section 5.4.

5.1 Quantum randomness

I think it is pretty clear from what we saw in this paper why we need true random generators, in order to perform both classical and Quantum Cryptography.

A random generator is a device, either physical or computational, that generates random numbers (or characters) that cannot be predicted better than the random guesses.

The term pseudo-random random generator is used whenever we have something that looks likes random numbers but is in fact deterministic. A simple example is to use the decimal of π is a random generator for numbers between 0 and 9. If we look at the repartition over the first 1000000 of decimals we have

0	1	2	3	4
0.099959 %	0.099758 %	0.100026 %	0.100229 %	0.10023 %
5	6	7	8	9
0.100359 %	0.099548 %	0.0998 %	0.099985 %	0.100106 %

Table 5.1: Repartition of numbers in the first 1000000 of π

It is close to a uniform distribution [1].

Now let's consider a random generator, that chooses at random the first decimal (for instance the 145098th decimal) and goes one by one to generate random numbers. It will look random but it is totally deterministic.

To have a suitable random numbers generator for cryptography, there exist requirements for Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). There are two main requirements:

- First, the algorithm must pass the next-bit test. It the test for randomness and it works like this: if you have k bits of a random sequence generated by the CSPRNG, it passes the test if

---

[1] A real number that verifies the property of having its decimal uniformly distributed is called a normal number. π has not be proven to be a normal number but numerical experiments tends to confirm that.

there is no polynomial-time algorithm that guesses the (k+1)th bit with a probability greater than $\frac{1}{2}$

- Secondly, the algorithm must hold against attacks, even if the initial state or the current state is revealed. Suppose that the state of the CSPRNG is given for the k-th bit. Then the test is passed if it's impossible to reconstruct the sequence of $(k-1)$ bits that precede the revealing.

But, with quantum mechanics, we can use a non-deterministic process to generate true randomness. Quantum mechanics is already used to generate random numbers, see for example https://qrng.anu.edu.au/.

For instance, if we want to represent a random generator using quantum gates we could do the following:

1. Initialize the qubit in the $|0\rangle$.

2. Apply the Hadamard gate. The qubit will end up in the $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ state.

3. Measure in the $\{|0\rangle, |1\rangle\}$ basis. This will lead to the result 0 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$.
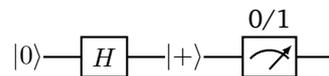

Figure 5.1: Quantum randomness using a Hadamard gate

In fact, just remember the conjugate coding principle: if we send a photon diagonally polarised and we make a measurement in the rectilinear basis, we should have a completely random outcome.

IDQ was the first company to develop a quantum random number generator ship and they today sell those kinds of chips. In this chip, the method used is photon counting [59]. On the website https://qrng.anu.edu.au/, the method uses the vacuum fluctuations [60]. Other methods are radioactive decay, noise, branching path, time of arrival, attenuated pulse, the phase noise of lasers, amplified spontaneous emission, RAMAN scattering, optical parametric oscillators, noise in electronics components, measurement on trapped ions, spin noise (for a full review of quantum random number generator, see [61]).

Without entering more in detail, let's just note that detectors' imperfections and errors may disturb the primary source of randomness, and often some kind of filters are used to purify the random data. Also, note that there are specifications and tests to verify random number generation schemes (some of them are published by the NIST for instance) and that some quantum chips have already passed those tests (some quantum schemes are even self-testing).

On an amusing note, IDQ and SK Telecom released in 2020 a smartphone equipped with a quantum random number generator chip: the Galaxy A quantum [62].

Hence, quantum random number generators has been extensively studied and is being used in real applications today. As we saw, a first smartphone was released with a quantum chip to generate a random number, which means that the chip was small enough to get into a smartphone. We can expect a democratisation of quantum random number generators in PCs and smartphones in the next few years. Will it have a real impact on our everyday life? Probably not.

5.2 Quantum Coin Flipping

5.2.1 The first-ever protocol

When we described the BB84 protocol and quantum key distribution in general, we assume implicitly that Alice and Bob *trust each other*. The idea of coin-flipping by telephone was first introduced

by Manuel BLUM in 1981 [63]. The basic idea is to share information between 2 parties that do not trust each other, for example, Alice flips a coin and asks Bob to make a guess over the phone. But as Bob doesn't see the actual result of the flip, Alice could cheat easily. Here, we are interested in a quantum version of coin-flipping.

We make the same assumptions as for the QKD protocol except for the first one. This protocol was also proposed by BENNETT and BRASSARD in 1984. It goes as follow:

1. Alice chooses a random basis $\mathcal{B}_z$ or $\mathcal{B}_x$.

2. Alice chooses a random bit string, encodes it in the chosen basis, and send it to Bob over the quantum channel.

3. At each qubit received, Bob chooses a random basis $\mathcal{B}_z$ or $\mathcal{B}_x$ and measures the qubit in this basis. He keeps the results in two separate columns (one for each basis).

4. Over the classical channel, Bob makes a guess on the basis used by Alice to encode her bits[2]. Alice says if Bob is right or wrong.

5. To prove she is not lying, she sends, over the classical channel, the bit string. The string should agree on Bob's column of the basis she claims have used.

Here is an example of how it can go:

Alice's Basis	$\mathcal{B}_x$						
Alice's Bits	0	1	0	0	1	1	0
Qubits	$\lvert+\rangle$	$\lvert-\rangle$	$\lvert+\rangle$	$\lvert+\rangle$	$\lvert-\rangle$	$\lvert-\rangle$	$\lvert+\rangle$
Bob's basis	$\mathcal{B}_z$	$\mathcal{B}_z$	$\mathcal{B}_x$	$\mathcal{B}_z$	$\mathcal{B}_x$	$\mathcal{B}_x$	$\mathcal{B}_x$
$\mathcal{B}_z$ register	0	0		1			
$\mathcal{B}_x$ register			0		1	1	0

Table 5.2: Example of realisation of quantum coin flipping

We see that, as expected, the input bits and the $\mathcal{B}_x$ register agree, as the $\mathcal{B}_x$ basis was used to encode the bits:

Alice's Bits	0	1	0	0	1	1	0
$\mathcal{B}_x$ register			0		1	1	0

Table 5.3: Comparison of input bits and $\mathcal{B}_x$ register

On the contrary, the $\mathcal{B}_z$ register and the input bits will probably not be equal:

Alice's Bits	0	1	0	0	1	1	0
$\mathcal{B}_z$ register	0	0		1			

Table 5.4: Comparison of input bits and $\mathcal{B}_z$ register

For each bit, the probability that they are different is $\frac{1}{2}$ and so, if Bob measures k times in the $\mathcal{B}$) basis, the probability that the input bits and the register are different is

$$1 - \left(\frac{1}{2}\right)^k \tag{5.1}$$

Now imagine that Bob makes the guess $\mathcal{B}_x$. He is right but Alice tries to cheat and says Bob is wrong, Now she has to provide a string that matches Bob's $\mathcal{B}_z$ register. But this register is basically a random string, and so Alice would have to do a completely random guess.

---

[2]Note that the measurements made by Bob don't give him any information on the basis used by Alice

5.2.2 Can Alice and Bob cheat ?

In this protocol, there is a simple way for Alice to cheat. Instead of using a single-photon source, she uses an EPR source and send one particle of an entangled pair to Bob and keeping the other particle to herself, using quantum memory. Once Bob has made his guess, she can do all the measurements in the opposite basis, and create a string that will match Bob's register.

In fact, this is not an isolated case. Quantum coin flipping was studied after the original proposal from BENNETT and BRASSARD, and unfortunately, things are not going well for quantum coin-flipping. In 1998 LO and CHAU proved that secure ideal quantum coin tossing (and also bit commitment which is a cryptographic task close to coin tossing) was not possible [64]. They did not only disprove the security of existing schemes but of all possible schemes. Basically, those schemes are not robust against EPR attacks. However, the authors posed the question of a non-ideal quantum coin tossing protocol. For that we have to identify three types of coin tossing:

- Ideal coin tossing: Each party has a probability of 0 of success when they try to bias the outcome;

- Weak coin tossing with bias ϵ: Alice cannot bias the outcome by more than ϵ in her favor and Bob cannot bias the outcome by more than ϵ in his favor;

- Strong coin tossing with bias ϵ: Alice and Bob cannot bias the outcome by more than ϵ in any direction.

Mathematically this can be defined as follow:

Definition (Weak quantum coin tossing protocol): A weak quantum coin tossing protocol with bias ϵ is one that verifies:

- If Alice and Bob are honest, then $\mathbb{P}(\text{Alice wins}) = \mathbb{P}(\text{Bob wins}) = \frac{1}{2}$;

- If Alice cheats and Bob is honest, then $\mathbb{P}(\text{Alice wins}) \leq \frac{1}{2} + \epsilon$;

- If Bob cheats and Alice is honest, then $\mathbb{P}(\text{Bob wins}) \leq \frac{1}{2} + \epsilon$;

Note: here we have defined a balanced protocol, in the sense that if Alice and Bob are honest they have the same chance of success. It is also possible to define unbalanced weak coin tossing protocols.

Definition (Strong quantum coin tossing protocol): A weak quantum coin tossing protocol with bias ϵ is one that verifies:

- If Alice and Bob are honest, then $\mathbb{P}(\text{Alice wins}) = \mathbb{P}(\text{Bob wins}) = \frac{1}{2}$;

- If Alice cheats and Bob is honest, then $\max(\mathbb{P}(\text{Alice wins}), \mathbb{P}(\text{Bob wins})) \leq \frac{1}{2} + \epsilon$;

- If Bob cheats and Alice is honest, then $\max(\mathbb{P}(\text{Alice wins}), \mathbb{P}(\text{Bob wins})) \leq \frac{1}{2} + \epsilon$;

(see [65] for the definitions for instance).

Ideal quantum coin tossing is not possible, but weak and strong quantum coin tossing are possible. Weak quantum coin tossing can be performed with arbitrarily small bias [66] whereas strong quantum coin tossing can be performed with a minimal bias which is $\frac{\sqrt{2}}{2} - \frac{1}{2}$ [67], but the bias can be made arbitrarily close to this value of $\frac{\sqrt{2}}{2} - \frac{1}{2}$ [68]. In [67], it was however proven that, even if weak quantum coin tossing can be performed, this is not very efficient as at least $\exp\left(\Omega(\frac{1}{\sqrt{\epsilon}})\right)$ rounds of communication are needed for bias ϵ.

5.3 Quantum money

In the first paper of Quantum Cryptography, written by Stephen WIESNER [16], which we discussed previously as it introduces conjugate coding, there was a proposal for quantum money or money

that is physically impossible to copy.

According to the Bank of England [69], a small fraction of the banknotes circulating in the UK are counterfeits (around 0.02%), representing around 10 million pounds of fake and worthless money.

Another reason to be interested in Quantum Money is the potential security breach in crypto-moneys. The most famous crypto-money is Bitcoin, and it has been more and more used in the last few years. The issue is that in the end, the Bitcoin (an other crypto-moneys) relies on public-key cryptosystems that could be become insecure with quantum computational power.

We will briefly see several implementations of quantum money here, but all of them have the requirement of quantum memory that can store the quantum states for a long time (the lifetime of the banknote or credit card), which is not yet available (and not believed to be in the near future).

There are two categories of quantum money:

- private quantum money: only the bank can verify that the money is no counterfeit;

- public quantum money: anyone can verify that the money is no counterfeit;

The schemes of these two categories are composed of three functions:

- a generating function G that generates a key (private quantum money) or a public/private keys pair (public quantum money);

- a mint function M that is used to create new money (the function uses either the key (private quantum money) or the private key (public quantum money))

- a verify function V that is used to verify that the money is no counterfeit (it uses either the key (private quantum money) or the public key (public quantum money)).

There are several quantum money schemes that have been designed over the years, since WIESNER's first proposition, with very little experimental implementations. Also, it is possible to note that the idea of quantum money is related to fingerprinting (since they have common goals).

5.3.1 WIESNER's scheme

The first proposal of quantum money, by WIESNER, is a private quantum money scheme. The money is issued by the bank and can only be verified by the bank.

A banknote is composed of three different parts:

- classical financial information (the value, the currency, the bank that issued the banknote, etc...);

- a classical serial number;

- a quantum key, that for now, is composed of only one qubit.

The quantum key is composed one qubit, encoded in one the two bases $\mathcal{B}_z$ or $\mathcal{B}_x$. When creating the banknote, the bank chooses a random bit and encodes it in a basis chosen at random. This classical information (bit and basis) is stored with the serial number, securely at the bank.

Whenever a banknote needs to be verified, the bank looks at the serial number record and makes a measurement according to the basis in the record. If the measurement outcome is the same as in the record, they declare the banknote legit, and they declare it counterfeit otherwise.

A malicious party cannot create a banknote with a completely new serial number but can try to copy one. To maximise the chance of creating a legit copy, the malicious party makes a measurement in a basis chosen at random and encodes the outcome measurement in the same basis.

- The good basis is chosen with probability $\frac{1}{2}$ (assuming that both the bank and the malicious party choose the basis with equal probability). In this case, the banknote will always pass the test of the bank.

- The wrong basis is chosen with probability $\frac{1}{2}$. In this case, the banknote will pass the test of the bank with probability $\frac{1}{2}$.

Overall, the copy operation has a success probability of $\frac{3}{4}$. This is pretty high but, like in the BB84 protocol when Alice and Bob reveal bits trying to detect a potential eavesdropper, the bank can store k qubits in the banknote, and the then the overall probability of success when trying to copy a banknote is $\left(\frac{3}{4}\right)^k$ and can therefore be made arbitrarily small.

5.3.2 BOZZIO *et al.*'s scheme

In 2018, BOZZIO *et al.* proposed a quantum money scheme close, presented as a quantum credit card [70]. We do not handle the payment mechanism, but we are interested in a scheme that doesn't allow a copy of the card.

The bank emits a prepaid card, which has a unique card number. Then a quantum key is stored on the card. The quantum key is a qubit pair, the two qubits begin encoded in different bases. The bit encoded can be 0 or 1 so there are eight possibilities. This eight possibilities can be encoded on 3 classical bit $\{b, c_0, c_1\}$. c_0 and c_1 are the values of the encoded bits and $b = 0$ if the first basis is $\mathcal{B}_z$ and the second basis is $\mathcal{B}_x$ and $b = 1$ if the first basis is $\mathcal{B}_x$ and the second $\mathcal{B}_z$. There eight of those pairs, each with a corresponding secret key:

Pair	$|0+\rangle$	$|0-\rangle$	$|1+\rangle$	$|1-\rangle$	$|+0\rangle$	$|+1\rangle$	$|-0\rangle$	$|-1\rangle$
Code	$\{0,0,0\}$	$\{0,0,1\}$	$\{0,1,0\}$	$\{0,1,1\}$	$\{1,0,0\}$	$\{1,0,1\}$	$\{1,1,0\}$	$\{1,1,1\}$

Table 5.5: 8 possible pairs for the quantum card code

The quantum pair is stored on the card. The classical code is stored at the bank, alongside with the serial number of the card.

Now let's see how the card is verified. When a transaction is to be done, the seller, through a payment terminal, verifies the card with the following steps:

1. The payment terminal randomly chooses a challenge between two possible challenges: Q_{zz} or Q_{xx}. The Q_{zz} challenge is the challenge that always gets the correct bit for the one encoded in the $\mathcal{B}_z$ whereas Q_{xx} is the one that always gets the correct bit for the one encoded in the $\mathcal{B}_x$ basis. Q_{zz} is implemented by the $\boldsymbol{\sigma_z} \otimes \boldsymbol{\sigma_z}$ measurement and Q_{xx} is implemented through $\boldsymbol{\sigma_x} \otimes \boldsymbol{\sigma_x}$;

2. When the measurement is performed, the payment terminal ends up with two classical bits. There are sent (along with information about the card like the serial number, that is read classically) to the bank. The chosen challenge is also sent to the bank;

3. The bank checks the bits that should match (it depends on b and the chosen challenge). If they match, the card is declared correct, and if not it is counterfeit.

Let set an example:

1. The bank issues the card with the quantum code $|0-\rangle$, which corresponds to the classical code $\{0, 0, 1\}$;

2. The legitimate user uses the card. The payment terminal chooses the Q_{xx} challenges and ends up with the result $\{1, 1\}$. The result is sent to the bank;

3. The bank knows that the qubit encoded in the $\mathcal{B}_x$ basis is the second one, so it verifies that the second bit of the classical code and of the measurement outcome. They match and the bank sends the answer to the payment terminal, where the transaction can take place.

Now let's see how a malicious party would try to copy the key. The basic idea to do that is to guess the value of b (i.e. randomly make an assumption whether $b = 0$ or $b = 1$) and make the measurements. With probability $\frac{1}{2}$, the value of b is correct and then, the malicious party has a perfect copy of the card. In the other case, the card will be encoded using the wrong bases and hence, the card is detected to be wrong with probability $\frac{1}{2}$ at each transaction.

One of the main advantages of this protocol is that it only requires classical communication to verify a card, whereas, in WIESNER's protocol, the bank should have access to the qubits, and hence it needed some sort of quantum communication.

As before, the number of quantum pairs stored and the card can be augmented to lower the probability of successfully copying a card.

5.3.3 Semi quantum money

In the first subsection, where we reviewed WIESNER's scheme, we saw a scheme that requires:

- a quantum bank;

- a quantum user;

- quantum communication for minting and verifying money.

In the second subsection, where we reviewed BOZZIO *et al.*'s scheme, we saw a scheme that requires:

- a quantum bank (the bank only needs to be quantum for the minting process);

- a quantum user;

- a classical channel for verifying money;

- a quantum channel for minting and sending money to the user.

In 2019 Roy RADIAN and Or SATTATH proposed the idea of semi quantum money [71]. In a public semi quantum scheme:

- the user is quantum;

- the bank is classical;

- the quantum money is minted by the user;

- only classical communications are needed for minting and verification.

There are several advantages of using only classical channels: the first being that we already have the infrastructure for classical communication, also we have robust error corrections on classical channels, whereas losses and errors on quantum channels may yield invalid banknotes and hence loss of money.

We won't describe in detail semi quantum money schemes as a strong background is needed.

To conclude on quantum money, even if we research have been done and protocols exist, it requires long-term quantum memory, which we don't have at our disposal today, and for some protocols, a heavy quantum communication infrastructure.

5.4 Quantum voting

5.4.1 Electronic voting

Electronic voting refers to two types of voting methods:

- First, when electronic devices are placed at voting locations, i.e. under the responsibility of government officials;

- Secondly, when internet voting is used, i.e. citizens can vote from wherever they want.

Two main requirements for voting were identified in [72, 73], and were proved to be not fulfilled with electronic voting:

- First, the vote needs to be anonymous. It must not be possible to identify how an individual voted afterward;

- Secondly, the vote needs to be trustworthy. An individual must know that his vote was taken into account.

It is impossible to fulfill these requirements with electronic voting, even more, when vote over the internet is used.

5.4.2 Quantum voting

It is clear that quantum voting will not be used for large scale votes before a pretty long time, and that for two reasons:

- It would very expensive in quantum resources,

- It is not possible to expect the public to understand quantum physics, and they cannot trust something they don't understand.

Nevertheless, quantum voting can be interesting to study. Also, it shows how far can quantum principles be used in cryptographic protocols.

Following [74, 75], we are going to specify in more detail the requirements of a voting scheme. An open ballot system is a voting scheme where users sign their ballot or make them identifiable in some way. In this system, each voter can know what an other voter has voted. For instance, a "show of hands" vote in a general assembly or when MPs cry "aye" or "no" in the UK Parliament are examples of open ballot systems. Opposed to those systems are the closed ballot systems, or anonymous systems, that have 3 main requirements:

1. Correctness: a vote that is considered valid (i.e. with no identifying mark and from a valid voter) should be counted. A vote that is no considered valid should not be counted;

2. Anonymity: Votes are anonymous (i.e. no identifying mark on the ballot);

3. Receipt-free: there should not be any way to know how a voter voted or even if he has voted after the election.

Often, an additional requirement is that a valid voter should be able to vote only once.

Several quantum voting schemes have been designed [74, 75, 76, 77] and we will review one of them in more details (the scheme for OKAMOTO *et al.*).

In [75], the authors presented a voting scheme based on conjugate coding and uses the same bases and states that the BB84 protocol. This voting scheme requires:

- a counter, or scrutinizer C. He will verify and count votes. He needs to be trusted.

- an administrator A. He will issue blank ballots. There is no need for total trust in the administrator as he can be supervised by C, or even split into several parties (each party can forge a part of the ballot)

- v voters V_i $(i = 1, \ldots, n)$

In the beginning, the administrator A forges blank ballots for every voter. We now explain what a blank ballot is:

We denote by n a security parameter which is the length of a blank piece. A blank ballot is composed of several blank pieces.

At the beginning, the administrator randomly creates a secret K. This secret is the choice of $n + 1$ bases chosen between $\mathcal{B}_x$ and $\mathcal{B}_z$:

$$K = (\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n+1}) \tag{5.2}$$

To forge a blank piece, the administrator randomly chooses n bits $b_1, b_2, \ldots, b_n$, and b_{n+1} is defined to be $b_{n+1} = b_1 \oplus_2 b_2 \oplus_2 \cdots \oplus b_n$. The bits $(b_1, \ldots, b_{n+1})$ are encoded into quantum states using the bases of the secret K.

For instance, if $n = 2$, with $K = (\mathcal{B}_x, \mathcal{B}_z, \mathcal{B}_x)$. A valid blank piece would be, using $b_1 = 1$, $b_2 = 0$,

$$(\lvert - \rangle, \lvert 0 \rangle, \lvert - \rangle) \tag{5.3}$$

To construct a blank ballot for voter V_i, the administrator constructs m blank pieces with $r_1^i, \ldots, r_m^i$,

$$r_j^i = (b_{j,1}^i \ldots, b_{j,n+1}^i) \tag{5.4}$$

and $b_{j,n+1}^i = b_{j,1}^i \oplus_2 \cdots \oplus_2 b_{j,n}^i$ for $j = 1, \ldots, m$.

Next, one blank ballot is sent to each voter[3]

When the voter receives his ballot, he will first randomize the ballot. This step is here to preserve the anonymity of the voter. This is done as follows: for each state of each blank piece of the blank ballot, the voter will apply either the identity or the gate $\sigma_x \sigma_z$. For the n first states of the blank piece, the gate to apply is chosen at random. For the last bit of the blank piece, the gate is the identity, if the gate $\sigma_x \sigma_z$ was applied an even number of times on the n first states and $\sigma_x \sigma_z$ if the gate $\sigma_x \sigma_z$ was applied an odd number of times, this will make the blank piece stay valid. The gate $\sigma_z \sigma_x$ flips the state without changing its basis (the gate may change the overall phase but this cannot be measured).

$$\begin{aligned}
\sigma_x \sigma_z \lvert 0 \rangle &= \sigma_x \lvert 0 \rangle = \lvert 1 \rangle \\
\sigma_x \sigma_z \lvert 1 \rangle &= -\sigma_x \lvert 1 \rangle = -\lvert 0 \rangle \\
\sigma_x \sigma_z \lvert + \rangle &= \sigma_x \lvert - \rangle = -\lvert - \rangle \\
\sigma_x \sigma_z \lvert - \rangle &= \sigma_x \lvert + \rangle = \lvert + \rangle
\end{aligned} \tag{5.5}$$

By doing this step, the basis for each state remains the same, but the value if random. Also, the blank pieces remain valid.

Now that the voter has to cast his vote. We suppose that the set of all possible votes is a subset of $\{0, 1\}^m$. To write his vote, the voter applies either the identity (if he wants to encode a 0) or the gate $\sigma_x \sigma_z$ (if he wants to encode a 1) to the last state of each blank piece.

Finally, the voter sends his vote to the counter C[4]. Once the counter has received all votes, the administrator sends K to the counter through a secure classical channel. For each ballot, he measures each piece using K. The value of the bit for the piece is retrieved by summing all the results. The counter C verifies that the results are indeed a possible vote. If it is, the vote is counted, and if not, the vote is discarded.

The security of this voting scheme relies on the fact that a malicious party cannot create a ballot with a valid choice with high probability without knowing the secret K. In fact, the probability

---

[3]Here note that the channel used to send the ballot can be passively eavesdropped but is authenticated on both sides.
[4]Here, the channel can be passively monitored, but must be authenticated on the side of the receiver (C)

for a ballot that was forged without K to be accepted is $\frac{\#\text{candidates}}{2^m}$. As the number of candidates is constant, the probability can be made arbitrarily small by adding pieces to the ballot.

Let's see a quick example before concluding this section. Let $n = 3$ and $m = 3$ with two possible candidates: 010 and 101. The secret key is $K = (\mathcal{B}_z, \mathcal{B}_x, \mathcal{B}_x, \mathcal{B}_z)$.

The administrator A forges the ballot

$$
\begin{aligned}
&(\left|1\right\rangle, \left|+\right\rangle, \left|+\right\rangle, \left|1\right\rangle) \\
&(\left|0\right\rangle, \left|-\right\rangle, \left|+\right\rangle, \left|1\right\rangle) \\
&(\left|0\right\rangle, \left|-\right\rangle, \left|-\right\rangle, \left|0\right\rangle)
\end{aligned}
\tag{5.6}
$$

(i.e $r_1 = (1, 0, 0, 1), r_2 = (0, 1, 0, 1), r_3 = (0, 1, 1, 0)$).

The ballot is randomised by the voter:

$$
\begin{aligned}
&(\left|1\right\rangle, \left|+\right\rangle, \left|+\right\rangle, \left|1\right\rangle) \\
&(\left|0\right\rangle, \left|+\right\rangle, \left|+\right\rangle, -\left|0\right\rangle) \\
&(\left|1\right\rangle, \left|-\right\rangle, \left|+\right\rangle, \left|0\right\rangle)
\end{aligned}
\tag{5.7}
$$

applying the following operators:

$$
\begin{aligned}
&\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
&\mathbb{1} \otimes \boldsymbol{\sigma_x\sigma_z} \otimes \mathbb{1} \otimes \boldsymbol{\sigma_z\sigma_x} \\
&\boldsymbol{\sigma_x\sigma_z} \otimes \mathbb{1} \otimes \boldsymbol{\sigma_x\sigma_z} \otimes \mathbb{1}
\end{aligned}
\tag{5.8}
$$

The voter wants to vote for 101 so he will flip the last states for the first and last pieces (the blank ballot corresponds to 000), and the final ballot is hence

$$
\begin{aligned}
&(\left|1\right\rangle, \left|+\right\rangle, \left|+\right\rangle, -\left|0\right\rangle) \\
&(\left|0\right\rangle, \left|+\right\rangle, \left|+\right\rangle, -\left|0\right\rangle) \\
&(\left|0\right\rangle, \left|-\right\rangle, \left|+\right\rangle, \left|1\right\rangle)
\end{aligned}
\tag{5.9}
$$

applying the following operators

$$
\begin{aligned}
&\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \boldsymbol{\sigma_x\sigma_z} \\
&\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \\
&\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \boldsymbol{\sigma_x\sigma_z}
\end{aligned}
\tag{5.10}
$$

The counter C then makes measurements using the bases of K and finds the following results:

$$
\begin{aligned}
&(1, 0, 0, 0) \quad \bigoplus{}_2 = 1 \\
&(0, 0, 0, 0) \quad \bigoplus{}_2 = 0 \\
&(0, 1, 1, 1) \quad \bigoplus{}_2 = 1
\end{aligned}
\tag{5.11}
$$

where $\bigoplus_2$ is the sum modulo 2 over all the results of the measurements of one piece. C finds the vote 101 which is a correct vote and count it.

This protocol uses conjugate coding to implement a voting scheme. It is unconditionally guaranteed anonymous and has the property of correctness if the one-more unforgettability. This assumption is the following: there is no polynomial-time quantum algorithm that can create $l + 1$ blank pieces out of l blank pieces with high probability. However, to be implemented it requires quantum memory.

Also, the counter must be absolutely trusted. A version can remove this assumption: where a voter sends his vote to all other voters. The secret K is shared with every voter after all the votes have been cast. This is however much heavier in quantum resources.

Another protocol was proposed in [77], using GREENBERGER–HORNE–ZEILINGER state and entanglement. In this scheme, the administrator A and the counter C supervise each other (at the end, one of them must be however trusted).

Conclusions

In this review, we saw the principles allowing the creation of unconditionally secure protocols using Quantum Physics. We have first reviewed quantum key distribution, which is historically the first application of quantum physics in cryptography if we omit the unnoticed propositions of WIESNER. Several schemes were reviewed based on conjugate coding, entanglement, and BELL's inequality violation. The issue with the distance and fiber losses was treated and solutions including quantum relays and repeaters were proposed. One aspect that was not reviewed and which has been experimentally tested is satellite-to-ground quantum key distribution with propagation in free space [78]. A key exchange using a stellite was performed between two points on Earth separated by 7600 km [79] Finally, schemes based on BELL's inequality violation were proven useful to do device independent quantum key distribution.

Next, we went beyond QKD and explore quantum public cryptographic tasks: public-key, digital signatures, and fingerprinting. It shows the range of possibilities and how we can extend Quantum Cryptography. However, it seems that these tasks require a lot of resources, and sometimes devices that are not available today.

Finally, we presented other quantum cryptographic tasks. Quantum randomness is already commercialised while the other tasks do not seem attainable today (quantum money requires long term quantum memory for example). One thing we could talk about in this section are quantum games [80], which is an extension of the classical game theory, and have some interconnections with Quantum Cryptography.

Appendix A

End of document notes

To stay in line with cryptography, this paper was digitally signed without data encryption using PGP signature 438C07EC20B645D05BBE4C83DC24C5787C943389.

You can find the signature of this document, along with my public key, at the address

https://signatures.nanoy.fr

(note that DNSSEC and secure http are enabled).

This is document

QFFF Dissertation - Quantum Cryptography

revision

25.09.2020.1

The quantum circuits were drawn using quantikz. As asked in the documentation, I cite their paper because I found it useful [81].

The simulation were mainly coded with Simulaqron which is an open source quantum network simulator available at http://www.simulaqron.org/[82].

Appendix B

List of simulation codes

The simulation codes are available at <https://github.com/nanoy42/qfff-dissertation-simulation> :

- bb84/bb84_simple : simulation without Eve and without losses;

- bb84/bb84_with_eve : simulation with Eve and without losses;

- bb84/bb84_losses : simulation with Eve and with losses;

- quantum_randomness : quantum randomness concept using cirq

Note that all the code is distributed under the GPLv3 license. More simulation may be added afterwards including error correction and privacy amplification.

Bibliography

1. SUETONIUS. *The Lives Of The Twelve Caesars* [De Vita XII Caesarum]. 121. Available from: http://thelatinlibrary.com/suetonius/suet.caesar.html#56

2. Micka P. Letter frequency (English). Available from: http://en.algoritmy.net/article/40379/Letter-frequency-English

3. WhatsApp. WhatsApp Encryption Overview. Tech. rep. WhatsApp, 2017 Dec. Available from: https://scontent.whatsapp.net/v/t61.22868-34/68135620_760356657751682_6212997528851833559_n.pdf/WhatsApp-Security-White-paper.pdf

4. Rivest RL, Shamir A, and Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 1978 Feb; 21:120–6. DOI: 10.1145/359340.359342. Available from: https://doi.org/10.1145/359340.359342

5. Pomerance C. A tale of two sieves. NOTICES AMER. MATH. SOC 1996; 43:1473–85

6. Benioff P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. Journal of Statistical Physics 1980; 22:563–91. DOI: 10.1007/bf01011339

7. Feynman RP. Simulating physics with computers. International Journal of Theoretical Physics 1982; 21:467–88. DOI: 10.1007/bf02650179

8. GROVER L. A fast quantum mechanical algorithm for database search. Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96 1996. DOI: 10.1145/237814.237866. Available from: http://dx.doi.org/10.1145/237814.237866

9. SHOR P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing 1997 Oct; 26:1484–509. DOI: 10.1137/s0097539795293172. Available from: http://dx.doi.org/10.1137/S0097539795293172

10. VAN METER R, ITOH KM, and LADD TD. ARCHITECTURE-DEPENDENT EXECUTION TIME OF SHOR'S ALGORITHM. Controllable Quantum States 2008 Oct. DOI: 10.1142/9789812814623_0029. Available from: http://dx.doi.org/10.1142/9789812814623_0029

11. Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, Biswas R, Boixo S, Brandao FGSL, Buell DA, and al. et. Quantum supremacy using a programmable superconducting processor. Nature 2019; 574:505–10. DOI: 10.1038/s41586-019-1666-5

12. Martín-López E, Laing A, Lawson T, Alvarez R, Zhou XQ, and O'Brien JL. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. Nature Photonics 2012; 6:773–6. DOI: 10.1038/nphoton.2012.259. Available from: https://dx.doi.org/10.1038/nphoton.2012.259

13. CHEN L, STEPHEN J, LIU YK, MOODY D, PERALTA R, PERLNER R, and SMITH-TONE D. Report on Post-Quantum Cryptography. 2016 Apr

14. Bernstein DJ. Introduction to post-Quantum Cryptography. 2009. Available from: http://www.pqcrypto.org/www.springer.com/cda/content/document/cda_downloaddocument/9783540887010-c1.pdf

15. Wootters WK and Zurek WH. A single quantum cannot be cloned. Nature 1982; 299:802–3. DOI: 10.1038/299802a0. Available from: https://doi.org/10.1038/299802a0

16. Wiesner S. Conjugate Coding. SIGACT News 1983 Jan; 15:78–88. DOI: 10.1145/1008908.1008920. Available from: https://doi.org/10.1145/1008908.1008920

17. Einstein A, Podolsky B, and Rosen N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? Physical Review 1935; 47:777–80. DOI: 10.1103/physrev.47.777. Available from: https://doi.org/10.1103/physrev.47.777

18. Schrödinger E. Discussion of Probability Relations between Separated Systems. Mathematical Proceedings of the Cambridge Philosophical Society 1935; 31:555–63. DOI: 10.1017/s0305004100013554. Available from: https://doi.org/10.1017/s0305004100013554

19. Clauser JF, Horne MA, Shimony A, and Holt RA. Proposed Experiment to Test Local Hidden-Variable Theories. Physical Review Letters 1969; 23:880–4. DOI: 10.1103/physrevlett.23.880

20. Freedman SJ and Clauser JF. Experimental Test of Local Hidden-Variable Theories. Physical Review Letters 1972; 28:938–41. DOI: 10.1103/physrevlett.28.938

21. Bennett CH, Brassard G, Crépeau C, Jozsa R, Peres A, and Wootters WK. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Phys. Rev. Lett. 1993 Mar; 70(13):1895–9. DOI: 10.1103/PhysRevLett.70.1895. Available from: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895

22. VON NEUMANN John 11. Mathematical foundations of quantum mechanics. eng. Investigations in physics series ; no.2. Princeton: London ; Princeton University Press ; Oxford University Press, 1955

23. Holevo AS. Some estimates for the amount of information transmittable by a quantum communications channel. Problemy Peredači Informacii 1973; 9:3–11

24. Bennett CH and Brassard G. Quantum Cryptography: Public key distribution and coin tossing. Theoretical Computer Science 2014; 560. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84:7–11. DOI: https://doi.org/10.1016/j.tcs.2014.05.025. Available from: http://www.sciencedirect.com/science/article/pii/S0304397514004241

25. Kosloski JT. QKD Quantum Channel Authentication. 2006. arXiv: quant-ph/0603101 [quant-ph]

26. Rijsman B. A Cascade Information Reconciliation Tutorial. 2020. Available from: https://hikingandcoding.wordpress.com/2020/01/15/a-cascade-information-reconciliation-tutorial/

27. Brassard G, Lütkenhaus N, Mor T, and Sanders BC. Limitations on Practical Quantum Cryptography. Physical Review Letters 2000 Aug; 85:1330–3. DOI: 10.1103/physrevlett.85.1330. Available from: http://dx.doi.org/10.1103/PhysRevLett.85.1330

28. Collins D, Gisin N, and Riedmatten H de. Quantum Relays for Long Distance Quantum Cryptography. 2003. arXiv: quant-ph/0311101 [quant-ph]

29. Gisin N, Marcikic I, Riedmatten H de, Tittel W, and Zbinden H. Quantum communications with time-bin entangled photons: long distance quantum teleportation and quantum repeaters. 2003. arXiv: quant-ph/0301181 [quant-ph]

30. Bennett CH, Bessette F, Brassard G, Salvail L, and Smolin J. Experimental Quantum Cryptography. *Advances in Cryptology — EUROCRYPT '90*. Ed. by Damgård IB. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991 :253–65

31. Brassard G and Salvail L. Secret-Key Reconciliation by Public Discussion. *Advances in Cryptology — EUROCRYPT '93*. Ed. by Helleseth T. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994 :410–23

32. Bennett CH, Brassard G, and Robert JM. Privacy Amplification by Public Discussion. SIAM Journal on Computing 1988; 17:210–29. DOI: 10.1137/0217014. Available from: https://doi.org/10.1137/0217014

33. Bennett CH, Brassard G, Crepeau C, and Maurer UM. Generalized privacy amplification. IEEE Transactions on Information Theory 1995; 41:1915–23. DOI: 10.1109/18.476316. Available from: https://doi.org/10.1109/18.476316

34. Carter J and Wegman MN. Universal classes of hash functions. Journal of Computer and System Sciences 1979; 18:143–54. DOI: 10.1016/0022-0000(79)90044-8. Available from: https://doi.org/10.1016/0022-0000(79)90044-8

35. Huttner B and Ekert AK. Information Gain in Quantum Eavesdropping. Journal of Modern Optics 1994; 41:2455–66. DOI: 10.1080/09500349414552301

36. Lütkenhaus N. Security against eavesdropping in Quantum Cryptography. Physical Review A 1996; 54:97–111. DOI: 10.1103/physreva.54.97

37. Fuchs CA, Gisin N, Griffiths RB, Niu CS, and Peres A. Optimal Eavesdropping in Quantum Cryptography. I. 1997. arXiv: quant-ph/9701039 [quant-ph]

38. Dixon AR, Dynes JF, Lucamarini M, Fröhlich B, Sharpe AW, Plews A, Tam W, Yuan ZL, Tanizawa Y, Sato H, and al. et. Quantum key distribution with hacking countermeasures and long term field trial. Scientific Reports 2017; 7. DOI: 10.1038/s41598-017-01884-0. Available from: https://dx.doi.org/10.1038/s41598-017-01884-0

39. Ekert AK. Quantum Cryptography based on Bell's theorem. Phys. Rev. Lett. 1991 Aug; 67(6):661–3. DOI: 10.1103/PhysRevLett.67.661. Available from: https://link.aps.org/doi/10.1103/PhysRevLett.67.661

40. Bennett CH, Brassard G, and Mermin ND. Quantum Cryptography without Bell's theorem. Physical Review Letters 1992; 68:557–9. DOI: 10.1103/physrevlett.68.557. Available from: https://doi.org/10.1103/physrevlett.68.557

41. Pironio S, Acin A, Brunner N, Gisin N, Massar S, and Scarani V. Device-independent quantum key distribution secure against collective attacks. New Journal of Physics 2009 Apr; 11:045021. DOI: 10.1088/1367-2630/11/4/045021. Available from: https://doi.org/10.1088%2F1367-2630%2F11%2F4%2F045021

42. Briegel HJ, Dür W, Cirac JI, and Zoller P. Quantum repeaters for communication. 1998. arXiv: quant-ph/9803056 [quant-ph]

43. Lee SW and Jeong H. Bell-state measurement and quantum teleportation using linear optics: two-photon pairs, entangled coherent states, and hybrid entanglement. 2013. arXiv: 1304.1214 [quant-ph]

44. Acin A, Gisin N, and Masanes L. From Bell's Theorem to Secure Quantum Key Distribution. Physical Review Letters 2006; 97. DOI: 10.1103/physrevlett.97.120405

45. Acin A, Massar S, and Pironio S. Efficient quantum key distribution secure against no-signalling eavesdroppers. New Journal of Physics 2006 Aug; 8:126–6. DOI: 10.1088/1367-2630/8/8/126. Available from: https://doi.org/10.1088%2F1367-2630%2F8%2F8%2F126

46. Matsukevich DN, Maunz P, Moehring DL, Olmschenk S, and Monroe C. Bell Inequality Violation with Two Remote Atomic Qubits. Physical Review Letters 2008; 100. DOI: 10.1103/physrevlett.100.150404

47. Schwonnek R, Goh KT, Primaatmaja IW, Tan EYZ, Wolf R, Scarani V, and Lim CCW. Robust Device-Independent Quantum Key Distribution. 2020. arXiv: 2005.02691 [quant-ph]

48. Al-Kuwari S, Davenport J, and Bradford R. Cryptographic hash functions: recent design trends and security notions. English. *Short Paper Proceedings of 6th China International Conference on Information Security and Cryptology (Inscrypt '10)*. The 6th China International Conference on Information Security and Cryptology (Inscrypt 2010), 20-23 October 2010, Shanghai, China. Science Press of China, 2010 :133–50

49. Rivest R. The MD5 Message-Digest Algorithm. RFC 1321. RFC Editor, 1992 Apr :1–21. Available from: https://tools.ietf.org/html/rfc1321

50. Scott T and Sean E. The Two Generals' Problem. YouTube. 2019. Available from: https://www.youtube.com/watch?v=IP-rGJKSZ3s

51. Buhrman H, Cleve R, Watrous J, and Wolf R de. Quantum Fingerprinting. Physical Review Letters 2001 Sep; 87. DOI: 10.1103/physrevlett.87.167902. Available from: http://dx.doi.org/10.1103/PhysRevLett.87.167902

52. Yao A. Some complexity questions related to distributive computing(Preliminary Report). *STOC '79*. 1979

53. Nikolopoulos GM. Applications of single-qubit rotations in quantum public-key cryptography. Physical Review A 2008 Mar; 77. DOI: 10.1103/physreva.77.032348. Available from: http://dx.doi.org/10.1103/PhysRevA.77.032348

54.   Okamoto T, Tanaka K, and Uchiyama S. Quantum Public-Key Cryptosystems. *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO '00. Berlin, Heidelberg: Springer-Verlag, 2000 :147–65

55.   Daniel G. Quantum Public Key Cryptography with Information-Theoretic Security. Available from: https://www.perimeterinstitute.ca/personal/dgottesman/Public-key.ppt [Accessed on: 2020 Sep 23]

56.   Yang L, Yang B, and Pan J. Quantum public-key encryption protocols with information-theoretic security. 2012. DOI: 10.1117/12.922444

57.   Seyfarth U, Nikolopoulos GM, and Alber G. Symmetries and security of a quantum-public-key encryption based on single-qubit rotations. Physical Review A 2012 Feb; 85. DOI: 10.1103/physreva.85.022342. Available from: http://dx.doi.org/10.1103/PhysRevA.85.022342

58.   Gottesman D and Chuang I. Quantum Digital Signatures. 2001. arXiv: quant-ph/0105032 [quant-ph]

59.   Fürst M, Weier H, Nauerth S, Marangon D, Kurtsiefer C, and Weinfurter H. High speed optical quantum random number generation. Optics express 2010 Jun; 18:13029–37. DOI: 10.1364/OE.18.013029

60.   Symul T, Assad SM, and Lam PK. Real time demonstration of high bitrate quantum random number generation with coherent laser light. Applied Physics Letters 2011; 98:231103. DOI: 10.1063/1.3597793. Available from: https://dx.doi.org/10.1063/1.3597793

61.   Herrero-Collantes M and Garcia-Escartin JC. Quantum random number generators. Reviews of Modern Physics 2017; 89. DOI: 10.1103/revmodphys.89.015004

62.   IDQuantique. ID Quantique and SK Telecom announce the world's first 5G smartphone equipped with a Quantum Random Number Generator (QRNG) chipset. 2020 May. Available from: https://www.idquantique.com/id-quantique-and-sk-telecom-announce-the-worlds-first-5g-smartphone-equipped-with-a-quantum-random-number-generator-qrng-chipset/ [Accessed on: 2020 Sep 20]

63.   BLUM M. Coin Flipping by Telephone. *Advances in Cryptology: A Report on CRYPTO 81*. 1981 Nov :11–5. Available from: https://www.iacr.org/archive/crypto81/11_blum.pdf

64.   Lo HK and Chau H. Why quantum bit commitment and ideal quantum coin tossing are impossible. Physica D: Nonlinear Phenomena 1998 Sep; 120:177–87. DOI: 10.1016/s0167-2789(98)00053-0. Available from: http://dx.doi.org/10.1016/S0167-2789(98)00053-0

65.   Chailloux A and Kerenidis I. Optimal Quantum Strong Coin Flipping. 2009 50th Annual IEEE Symposium on Foundations of Computer Science 2009 Oct. DOI: 10.1109/focs.2009.71. Available from: http://dx.doi.org/10.1109/FOCS.2009.71

66.   Aharonov D, Chailloux A, Ganz M, Kerenidis I, and Magnin L. A Simpler Proof of the Existence of Quantum Weak Coin Flipping with Arbitrarily Small Bias. SIAM Journal on Computing 2016; 45:633–79. DOI: 10.1137/14096387X. eprint: https://doi.org/10.1137/14096387X. Available from: https://doi.org/10.1137/14096387X

67.   Miller CA. The impossibility of efficient quantum weak coin flipping. Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing 2020 Jun. DOI: 10.1145/3357713.3384276. Available from: http://dx.doi.org/10.1145/3357713.3384276

68.   Chailloux A and Kerenidis I. Physical Limitations of Quantum Cryptographic Primitives or Optimal Bounds for Quantum Coin Flipping and Bit Commitment. SIAM Journal on Computing 2017; 46:1647–77. DOI: 10.1137/15M1010853. eprint: https://doi.org/10.1137/15M1010853. Available from: https://doi.org/10.1137/15M1010853

69.   England B of. Counterfeit banknotes. 2020. Available from: https://www.bankofengland.co.uk/banknotes/counterfeit-banknotes [Accessed on: 2020 Sep 7]

70.   Bozzio M, Orieux A, Trigo Vidarte L, Zaquine I, Kerenidis I, and Diamanti E. Experimental investigation of practical unforgeable quantum money. npj Quantum Information 2018; 4. DOI: 10.1038/s41534-018-0058-2. Available from: https://doi.org/10.1038/s41534-018-0058-2

71.   Radian R and Sattath. Semi-Quantum Money. Proceedings of the 1st ACM Conference on Advances in Financial Technologies 2019 Oct. DOI: 10.1145/3318041.3355462. Available from: http://dx.doi.org/10.1145/3318041.3355462

72. Scott T. Why Electronic Voting is a BAD Idea - Computerphile. YouTube. 2014. Available from: https://www.youtube.com/watch?v=w3_0x6oaDmI

73. Scott T and Sean E. Why Electronic Voting Is Still A Bad Idea. YouTube. 2019. Available from: https://www.youtube.com/watch?v=LkH2r-sNjQs

74. Vaccaro JA, Spring J, and Chefles A. Quantum protocols for anonymous voting and surveying. Physical Review A 2007; 75. DOI: 10.1103/physreva.75.012333

75. Okamoto T, Suzuki K, and Tokunaga Y. Quantum voting scheme based on conjugate coding. NTT Technical Review 2008 Jan; 6

76. Hillery M, Ziman M, Buzek V, and Bielikova M. Towards quantum-based privacy and voting. Physics Letters A 2006; 349:75–81. DOI: 10.1016/j.physleta.2005.09.010. Available from: https://dx.doi.org/10.1016/j.physleta.2005.09.010

77. Xue P and Zhang X. A simple quantum voting scheme with multi-qubit entanglement. Scientific Reports 2017; 7. DOI: 10.1038/s41598-017-07976-1. Available from: https://dx.doi.org/10.1038/s41598-017-07976-1

78. Liao SK, Cai WQ, Liu WY, Zhang L, Li Y, Ren JG, Yin J, Shen Q, Cao Y, Li ZP, and al. et. Satellite-to-ground quantum key distribution. Nature 2017; 549:43–7. DOI: 10.1038/nature23655. Available from: https://dx.doi.org/10.1038/nature23655

79. Liao SK, Cai WQ, Handsteiner J, Liu B, Yin J, Zhang L, Rauch D, Fink M, Ren JG, Liu WY, and al. et. Satellite-Relayed Intercontinental Quantum Network. Physical Review Letters 2018; 120. DOI: 10.1103/physrevlett.120.030501

80. Guo H, Zhang J, and Koehler GJ. A survey of quantum games. Decision Support Systems 2008; 46:318–32. DOI: 10.1016/j.dss.2008.07.001

81. Kay A. Tutorial on the Quantikz Package. 2018. arXiv: 1809.03842 [quant-ph]

82. Dahlberg A and Wehner S. SimulaQron—a simulator for developing quantum internet software. Quantum Science and Technology 2018 Sep; 4:015001. DOI: 10.1088/2058-9565/aad56e. Available from: http://dx.doi.org/10.1088/2058-9565/aad56e