TÉCNICO LISBOA

Quantum Backdoor – Performing Electronic Side-Channel Analysis on Quantum Key Distribution Systems

Maria Beatriz Soares Lopes da Costa

Thesis to obtain the Master of Science Degree in

Engineering Physics

Supervisor: Prof. Yasser Rashid Revez Omar
Prof. João Carlos Carvalho de Sá Seixas

Examination Committee

Chairperson: Prof. Luís Humberto Viseu Melo
Supervisor: Prof. Yasser Rashid Revez Omar
Members of the Committee: Prof. Gonçalo Nuno Marmelo Foito Figueira

May 2024

I would like to dedicate this thesis to my parents, for their endless support and love

Acknowledgements

I would like to start by thanking Prof. Yasser Omar, for taking me as his student during my Master Degree and introducing me to the topic of Quantum Information Technologies. I thank you for the opportunity of making what initially sounded like a crazy idea come to life, and for all the knowledge that I have obtained through the journal clubs and the weekly meetings, without which this thesis would be much poorer. I would also like to thank Prof. Paolo Villoresi and Prof. Giuseppe Vallone for their availability in allowing me to use their laboratory and experimental setup. To Prof. Ricardo Chaves, I would like to thank you for your feedback and guidance.

Additionally, I would like to thank the support of project QSNP – Quantum Secure Networks Partnership (GA 101114043) of the Horizon Europe Programme of the European Commission. I would also like to thank the members of the QuTe Lab – Quantum Technologies Laboratory and of the Physics of Information and Quantum Technologies Group, at CeFEMA/LAPMEP and PQI – Portuguese Quantum Institute.

To my parents, I thank you with all my heart for always being by my side and showing me what true kindness, empathy and strength is. Thank you for reminding me of what truly matters in times of doubt, and for holding my hand all the way from Portugal to Italy. To my grandparents, whose hands I will sadly no longer be able to hold, I thank you for my memories and the earnest happiness of my childhood. I thank you grandpa for my love of books, and I thank you grandma for showing me grace.

To my friends, I thank you all. For your moral compass, your tenacity, and kindness, I thank you Leonor for always staying by my side. To Luísa, thank you for your constant support, your understanding, intelligence and warmth. Rafael, thank you for your thoughtfulness and help and for grounding me when I most need it. Baltasar, thank you for offering me little escapes from work, and keeping my nerves at bay. And for all the others that I love, but couldn't fit in this little text, I thank you for your presence.

Resumo

Nas últimas décadas, a Distribuição de Chaves Quânticas (QKD) revelou-se uma solução promissora para garantir comunicações seguras, um assunto premente após a ameaça proposta pelo algoritmo de Shor. Oferecendo uma forma teoricamente segura de partilhar chaves secretas, o estado-da-arte de QKD registou progressos notáveis nos últimos anos. No entanto, embora teoricamente seguro, QKD não é seguro em termos de implementação e até agora, o estudo de vulnerabilidades físicas tem se focado no canal ótico.

O conceito de pirataria de uma configuração criptográfica através das suas falhas físicas, conhecido como análise de canal lateral, foi introduzido pela primeira vez na criptografia clássica, com o trabalho seminal de Paul Kosher. Desde então, a análise de energia e a análise eletromagnética de canal lateral tornaram-se um elemento essencial da criptoanálise clássica. No entanto, estes conceitos quase não foram aplicados a QKD.

Neste trabalho, propomos e implementamos um novo método para piratear um sistema QKD, explorando o consumo de energia do controlador eletrónico que controla os componentes eletro-óticos do transmissor de QKD. Para atingirem uma elevada taxa de transmissão, os módulos QKD requerem normalmente controladores eletrónicos, tais como Arranjos de Porta Programável em Campo (FPGAs). Aqui, mostraremos que o consumo de energia da FPGA pode revelar informações sobre a chave. A análise foi efetuada no transmissor QKD da Universidade de Pádua. Os nossos resultados são consistentes com uma fuga de informação, tendo-se atingido uma precisão máxima de $73,35\%$ para a previsão de chaves aleatórias a uma frequência de repetição de qubit de 100 MHz.

Palavras-Chave: Distribuição de Chaves Quântica, Análise de Canal Lateral, Consumo de Energia, Arranjos de Porta Programável em Campo

Abstract

Over the last decades, Quantum Key Distribution (QKD) has risen as a promising solution for secure communications, a pressing subject in the aftermath of the security threat posed by Quantum Computers and the Shor's Algorithm. Offering a theoretically secure way to share secret keys between parties, QKD state-of-the-art has witnessed remarkable progress in the last years. Nonetheless, although theoretically secure, QKD is not implementation-secure and until now, the study of physical vulnerabilities in QKD setups has mainly focused on the optical channel.

The concept of hacking a cryptographic system via its physical characteristics and associated leakages, known as side-channel analysis, was firstly introduced in classical cryptography, with the seminal work of Paul Kosher. Since then, power and electromagnetic side-channel analysis have become a staple in classical cryptanalysis. However, these concepts have hardly been applied to QKD.

In this work we propose and implement a new method for side-channel analysis to QKD systems, by exploiting the power consumption of the electronic driver controlling the electro-optical components of the QKD transmitter. For high-rate transmission, QKD modules typically require electronic drivers, such as Field Programmable Gate Arrays (FPGAs). Here, we will show that the FPGA's power consumption can leak information about the QKD operation, and consequently the generated key. The analysis was performed on the QKD transmitter at the University of Padua. Our results are consistent and show critical information leakage, having reached a maximum accuracy of 73.35% in the prediction of the generated random keys at 100 MHz qubit repetition frequency.

Keywords: Quantum Key Distribution, Side Channel Attack, Power Consumption, Field Programmable Gate Array

Contents

List of Figures

List of Tables

Glossary

MATLAB	*Software* de cálculo numérico
POGNAC	Polarization Modulator used in the QKD transmitter at the University of Padua
CIRC	Optical Circulator

Acronyms

BRAM	Block RAM Memory
CV	Continuous Variable
DI	Device Independent
DV	Discrete Variable
ENOB	Effective Number of Bits
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
IM	Intensity Modulator
MUB	Mutually Unbiased Basis
PNR	Photon Number Resolving
PNS	Photon Number Splitting
PBS	Polarizing Beam Splitter
PC	Polarization Controller
QBER	Quantum Bit Error Rate
QKD	Quantum Key Distribution
QRNG	Quantum Random Number Generator
RF	Radio Frequency
SPAD	Single Photon Avalanche Photodiode
SPDC	Spontaneous Parametric Down Conversion
SoC	System-on-a-Chip
SNSPD	Superconducting Nanowire Single-Photon Detectors

Chapter 1

Introduction

1.1 Motivation

Protecting information is a need that has followed mankind throughout its development, and presently, the relevancy of information security is ever more pressing. Currently, most of the cryptographic tasks make use of public-key encryption protocols, which rely on certain computational hardness assumptions, such as low factoring efficiency [1]. Until recently, classical algorithms offering efficiency advancements were the main threats to the current cryptography schemes, however, with the rise of quantum computation, the paradigm has shifted.

Proven by Peter Shor in 1994 [2], a large-scale Quantum Computer poses a threat to current public-key encryption protocols. Although this has yet to be achieved, the fast-paced development of the field, coupled with different research facilities reaching the Quantum Supremacy milestone [3, 4, 5], sets a positive precedent for the future of Quantum Computing and magnifies the significance of the privacy threat. These advancements not only underline the future need for full-proof encryption schemes, but the current one, since today's eavesdroppers may acquire cryptograms and store them until a large-scale quantum computer has been built [6].

In this climate, Quantum Key Distribution (QKD) rises as a promising solution. Through leveraging the laws of quantum mechanics, it offers a provably-secure method for two parties to create a shared secret key, that can then be used to encrypt messages. Given so, the field has witnessed remarkable progress in recent years, from pivotal demonstrations increasing distance and key rate [7, 8], to the deployment of long-range QKD networks in classical communication infrastructures [9, 10, 11].

However, although theoretically secure, implementations of QKD systems may hold physical vulnerabilities which can be explored to gain information about the key. Devising Quantum Hacking attacks is therefore crucial for the development of the field, since it is only by finding vulnerabilities that one can create countermeasures to mitigate them. Plenty of attacks have been devised [12, 13, 14, 15], some even implemented [16, 17], and the increasing relevancy of characterizing security breaches in QKD systems drives the search for new physical vulnerabilities.

1.2 Basic Concepts in Cryptography

Confidentiality, one of the main cryptographic tasks, ensures unauthorized parties cannot access private information [18], and can be achieved via encryption schemes.

Let us consider a simple, yet typical setting, where Alice wants to send a message to Bob, and keep it private against eavesdroppers, represented by Eve. This framework can be translated with some definitions. The message m is defined as an element from the message space $\mathcal{M}$, and is typically denoted as the plaintext. In order to guarantee that Eve cannot read m, Alice must encrypt it. Therefore we define the key space $\mathcal{K}$, whose elements $a \in \mathcal{K}$ Alice can use to encrypt m before sending it to Bob. Additionally, we must define an the encryption transformation, E_a such that, after using it to encrypt m, $c = E_a(m)$, Alice computes the cyphertext c, which she then sends to Bob. To retrieve the original plaintext m, Bob must decrypt the cypher text. To do so, he uses a key $b \in \mathcal{K}$ and a decryption transformation D_b, such that $m = D_b(c)$.

Evidently the encryption scheme relies on the initial choice of key pair (a, b) (which can consist of two identical keys, i.e. (a, a)). The necessity for secret keys is attributed to Kerckhoffs' Principle, which states that an encryption scheme should be considered publicly known upon its creation [19], making its confidentiality rely solely on the decryption key's secrecy. According to the properties of the key pair (a, b), encryption techniques can be categorized as symmetric-key and asymmetric-key encryption.

Symmetric-key encryption defines schemes where each key in a pair (a, b) can be determined using the other pair key in a computationally easy manner [18]. Typically, in practical schemes, all parties share the same key, this is $a = b$. Since the encryption and decryption transformations are considered public knowledge, and key b can be easily derived from key a, secret-key sharing between parties rises as a pivotal necessity. Additionally, symmetric-key protocols have the fastest implementations in hardware and software, which make them an enticing choice for large data encryption and emphasises the need for sharing secret keys [20].

On the other hand, asymmetric-key encryption, also known as public-key encryption, defines schemes where the key pair (a, b) is composed of a publicly know encryption key a and a private decryption key b. The two keys must obey the following property: by knowing a, c and E_a, it is computationally infeasible to determine m. Therefore, Bob can create two keys, a and b, publicly share a and keep b private. Any party i with public access to a can then send a cyphertext $c_i = E_a(m_i)$ to Bob over an unsecured channel, given that Eve is unable to decypher c_i without the private key b. Since symmetric-key implementations achieve higher rates for data exchange, but require sharing a secret key between users, public-key schemes can be used for the creation of the secret key.

However, with the rise of Quantum Computation, the threat to public-key encryption protocols [2], spawns the need for new key distribution methods, and Quantum Key Distribution rises a promising solution. We note here that post-quantum cryptography is another research domain which addresses the security threat posed by Quantum Computers. This field aims at developing classical algorithms for which no known quantum computing attack exists. However, these algorithms do not offer long-term security, being based on mathematical conjectures only.

1.3 Quantum Key Distribution

Pioneered by Stephen Wiesner in the early 1970's [21], Quantum Cryptography pertains to all methods which use the fundamental laws of Quantum Mechanics to encrypt information in a provably-secure manner. Quantum Key Distribution (QKD) is the primary example of a Quantum Cryptography task. It encompasses all the ways which leverage the laws of Quantum Mechanics to create a shared secret key between users in a provably-secure way. Importantly, QKD is not a method to transmit messages, and serves to satisfy the necessity for a provably secure way of creating shared secret keys for symmetric-key encryption protocols.

The first QKD protocol, known as the BB84 protocol, takes its name after its authors, Charles H. Bennet and Gilles Brassard, who proposed it in 1984 [22]. Since then, many protocols have been put forth [23, 24, 25, 26, 27], and despite their differences, they all share the same base idea, which is to encode and transfer information by using quantum states. In a typical QKD protocol, Alice and Bob are connected through an unsecured quantum channel which is used by the parties to send quantum states to each other. In this quantum channel, Eve can interact with the quantum states, however, by doing so, she introduces errors in the key. These errors, detected during classical communication between Alice and Bob over authenticated channels, allow them to detect Eve's presence. This error introduction, explained by the rules of quantum mechanics, is the essence of security in QKD [28].

The physical system chosen to encode the quantum state divides QKD protocols into discrete variable (DV) protocols, described by finite-sized Hilbert spaces, and continuous variable (CV) ones, described by infinite-sized Hilbert spaces. This thesis will focus on DV systems, considered the earliest form of QKD, since the BB84 protocol was initially conceived for DV applications. In these systems, the single photon is used as a vessel, with its polarization or phase typically used to encode information.

Another important way of distinguishing between QKD protocols is on whether they rely on distributing entanglement, named entanglement based protocols, or not, meaning prepare and measure protocols. In the latter, exemplified by the BB84 [22], Alice encodes a random string of bits using a pool of non-orthogonal quantum states and sends them to Bob through a quantum channel. Upon receiving them, Bob retrieves the string by measuring the states. These protocols consistently follow an initial two-step procedure, with Alice preparing the state and Bob measuring it, which distinguishes them from the entanglement based framework, where Alice and Bob both measure incoming states. The latter, firstly suggested by Ekert in 1991 [23], requires an entangled state to be shared between the two users. By measuring her system, Alice prepares Bob's system, which allows for both of them to share a key.

Aiming at understanding DV QKD protocols, it is important to start by grasping some quantum mechanics notions for DV systems. With this knowledge, given that this thesis will focus on prepare-and-measure systems, we will move on to the study of the BB84 protocol, and then to the state of the art of current implementations. This last step will conclude the fundamental groundwork necessary to understand DV QKD systems.

1.3.1 Quantum Information Notions

A qubit is a two-dimensional quantum state, in a bi-dimensional Hilbert space $\mathcal{H}_2$, that can be written as a linear combination of basis states [28].

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1 \tag{1.1}$$

where $\mathcal{B}_{01} = \{|0\rangle, |1\rangle\}$ is known as the computational basis and the basis states are vectors given by

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1.2}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.3}$$

Being (θ, ϕ) the polar and azimuthal angle defining a sphere surface, the two degrees of freedom (α, β) can be written as a function of (θ, ϕ), such that every pure qubit state can be mapped to a position in the surface of a sphere. This unit sphere, schematized in Figure 1.1, is denoted the Bloch sphere, and every state $|\psi\rangle$ can be written as

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle \quad \text{with} \quad \theta \in [0, \pi] \;\; \phi \in [0, 2\pi]. \tag{1.4}$$

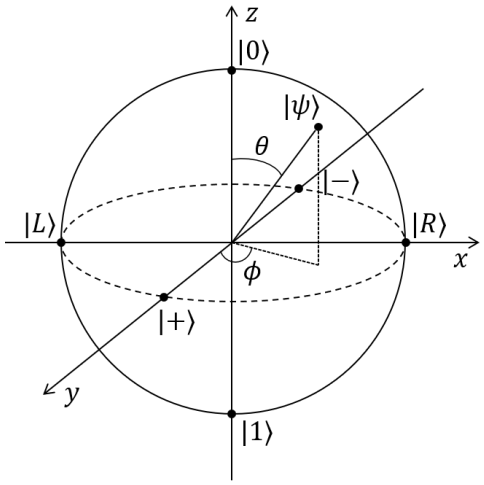


Figure 1.1: Representation of the Bloch Sphere

The poles of the sphere represent the computational basis states ($\theta = 0$ and $\theta = \pi$) and in the equator we can identify two pairs of states which constitute important basis for QKD protocols, which are $\mathcal{B}_{+-} = \{|+\rangle, |-\rangle\}$ and $\mathcal{B}_{LR} = \{|L\rangle, |R\rangle\}$.

The three basis $\mathcal{B}_{01}, \mathcal{B}_{+-}$ and $\mathcal{B}_{LR}$, defined in $\mathcal{H}_2$, have the important characteristic of being mutually unbiased. By definition [29], a pair of orthonormal basis $\mathcal{B}_\gamma = \{|b_1^\gamma\rangle, |b_2^\gamma\rangle, ..., |b_d^\gamma\rangle\}$ and $\mathcal{B}_\sigma = \{|b_1^\sigma\rangle, |b_2^\sigma\rangle, ..., |b_d^\sigma\rangle\}$, in a Hilbert Space of dimension d, $\mathcal{H}_d$, define mutually unbiased basis (MUB) if they obey Equation 1.5.

$$\left|\langle b_i^\gamma | b_j^\sigma \rangle\right|^2 = \frac{1}{d} \;\; \forall i, j \in \{1, ..., d\} \tag{1.5}$$

For describing quantum systems whose state is not completely know, it is useful to introduce the

density matrix ρ [28]. Considering a quantum system that is in one of a number of states $|\psi_i\rangle$, with respective probability p_i, the density matrix of the system is

$$\rho = \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right| . \tag{1.6}$$

The density matrix formalism is mathematically identical to the state vector one, used in Equations 1.4 and 1.1. Nonetheless, the former is more useful for describing mixed states, this is, quantum systems whose state is not completely known. Before, the Block sphere was introduced for pure states, this is, for quantum systems who state is exactly known, via Equation 1.4. However, it can also be generalized for mixed states. In fact, any mixed state can be mapped to a point in the interior of the Bloch Sphere [28]. Note also that, the density matrix ρ of a pure state $|\psi\rangle$ is written as $\rho = |\psi\rangle\langle\psi|$.

To fully understand prepare and measure protocols, it is important to grasp the concept of measurement of a quantum system. Here, we will introduce the measurement postulate of Quantum Mechanics using the state vector formalism, nonetheless, it can be reformulated using the density matrix formalism, for which we refer to [28]. Measuring a quantum state $|\psi\rangle$ is an operation which can be described by a set of operators $\{M_m\}$, acting on the state space, where m are the number of possible outcomes for the measurement [28]. Each measurement outcome has a probability of occurring given by Equation 1.7, and the state of the system after the measurement $|\psi'\rangle$ is be determined by Equation 1.8.

$$p_m = \langle\psi|M_m^\dagger M_m|\psi\rangle \tag{1.7}$$

$$|\psi'\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} \tag{1.8}$$

For most applications of DV QKD, the primary concern is with measurements described by an observable M, an Hermitian operator acting on the state space of the qubit [28], with spectral decomposition given by Equation 1.9.

$$M = \sum_i m P_m \tag{1.9}$$

$$\sum P_m = I \tag{1.10}$$

$$P_m P_{m'} = \delta_{mm'} P_m \tag{1.11}$$

P_m are the projectors onto the eigenspace of M, and they obey the properties given by Equation 1.10 and Equation 1.11. In this case, each outcome, corresponding to eigenvalue m, has a probability given by Equation 1.12. The final state $|\psi'\rangle$ is determined using Equation 1.13.

$$p_m = \langle\psi|P_m|\psi\rangle \tag{1.12}$$

$$|\psi'\rangle = \frac{P_m\,|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}} \tag{1.13}$$

Performing a measurement in the $\mathcal{B}_{01}$ basis corresponds to performing a projective measurement using the set of operators $\{|0\rangle\langle 0|\,, |1\rangle\langle 1|\}$ and so on for $\mathcal{B}_{+-}$ and $\mathcal{B}_{LR}$. Let us now consider we want to measure state $|b_i^\gamma\rangle$, which corresponds to one of the two states ($i \in \{0,1\}$) of one of the three previously discussed basis ($\gamma \in \{01, +-, LR\}$). When measuring this state in a MUB $\mathcal{B}_\sigma$ ($\sigma \neq \gamma$, with $\sigma, \gamma \in \{01, +-, LR\}$), we now see from Equations 1.5 and 1.12 that it will produce one of the eigenstates of $\mathcal{B}_\sigma$ with equal probability.

$$p_{b_j^\sigma} = \langle b_i^\gamma|P_{b_j^\sigma}|b_i^\gamma\rangle = \langle b_i^\gamma|b_j^\sigma\rangle\langle b_j^\sigma|b_i^\gamma\rangle = \frac{1}{2} \tag{1.14}$$

To finish this overview, it is important to tackle entanglement, a purely quantum mechanical resource which also plays an important role in DV QKD. A composite system is said to be entangled if it cannot be written as a product of states of its component systems [28]. In other words, a state $|\psi\rangle$ is entangled if and only if its Schmidt Rank, this is, the number of Schmidt coefficients of its Schmidt decomposition, is strickly greater than one. For example, the following state $|\psi\rangle$ represents a bipartite entangled state

$$|\psi\rangle_{AB} = \frac{|00\rangle_{AB} + |11\rangle_{AB}}{\sqrt{2}}. \tag{1.15}$$

The Bell states are maximally entangled bipartite states which are of crucial importance for entanglement-based QKD protocols. They can be written in the computational basis as

$$\left|\Phi^+\right\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle_{AB} + |11\rangle_{AB}) \tag{1.16}$$

$$\left|\Phi^-\right\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle_{AB} - |11\rangle_{AB}) \tag{1.17}$$

$$\left|\Psi^+\right\rangle_{AB} = \frac{1}{\sqrt{2}}(|01\rangle_{AB} + |10\rangle_{AB}) \tag{1.18}$$

$$\left|\Psi^-\right\rangle_{AB} = \frac{1}{\sqrt{2}}(|01\rangle_{AB} - |10\rangle_{AB}) \tag{1.19}$$

From the definition of entanglement one sees that the composite system states which make up an entangled state cannot be described independently from one another. Measuring one of these composite system states causes changes in the other. To see this, let us imagine that Alice and Bob share a Bell state $|\Phi^+\rangle_{AB}$ and Alice measures her state in the $\mathcal{B}_{01}$ basis.

$$p_0 = {}_{AB}\langle\Phi^+|0\rangle_{AA}\langle 0|\Phi^+\rangle_{AB} = \frac{1}{2} \tag{1.20}$$

$$|\Phi'\rangle_{AB} = \frac{|0\rangle_{AA}\langle 0\,|\Phi^+\rangle_{AB}}{\sqrt{p_0}} = \frac{1}{2}\,|00\rangle_{AB} \tag{1.21}$$

As expected, there is a 50% probability for Alice to measure a $|0\rangle_A$ or a $|1\rangle_A$ state, which is also true for Bob. However, if Alice performs the measurement and obtains a $|0\rangle_A$ state, Bob's state will collapse into a $|0\rangle_B$ state, as it can be seen in Equation 1.21, which computes the final state shared by Alice and Bob after Alice's measurement. Now, if Bob measures his state, there is a 100% probability of measuring $|0\rangle_B$.

This phenomenon, is particularly important for entanglement-based protocols, since Alice and Bob, by measuring in the same basis, can know which state each one got.

1.3.2 BB84 Protocol

In a typical BB84 protocol, Alice and Bob, who wish to share a secret key, are connected through an unsecured quantum channel and an authenticated classical channel. Through the quantum channel, Alice sends qubits to Bob, and uses two mutually unbiased basis, in a 2-dimensional Hilbert Space $\mathcal{H}_2$, to encode information.

Without loss of generality, let us assume Alice chooses $\mathcal{B}_{01} = \{|0\rangle, |1\rangle\}$ and $\mathcal{B}_{+-} = \{|+\rangle, |-\rangle\}$. She starts by creating a random string of bits s_1, where each bit encodes the basis choice for each photon, and a second random string s_2 encoding the basis state choice for each photon. Considering $b_{i,1}$ as a bit encoding information about photon i in the first string s_1 and $b_{i,2}$ as one in the second string s_2, then the association between basis and state choice for each photon and the bit pair $(b_{i,1}, b_{i,2})$ must be previously agreed between Alice and Bob. Table 1.1 represents a possible convention choice.

$b_{i,1}$	Basis Choice	$b_{i,1}$	Basis State Choice
0	$\mathcal{B}_R$	0	$\|0\rangle$
0	$\mathcal{B}_R$	1	$\|1\rangle$
1	$\mathcal{B}_D$	0	$\|+\rangle$
1	$\mathcal{B}_D$	1	$\|-\rangle$

Table 1.1: Example of a possible association between bit pair $(b_{i,1}, b_{i,2})$ and qubit state

Upon receiving photons from Alice, Bob must decide on which basis to perform the measurement in. Therefore, he creates a string s_3 where each bit $b_{i,3}$ encodes a basis choice for the measurement performed on each upcoming photon, obeying the convention in Table 1.1. As mentioned before, Bob's measurement in each basis can be described by two sets of projective operators $\{P_0, P_1\}$ and $\{P_+, P_-\}$ [28].

$$P_0 = |0\rangle\langle 0| \tag{1.22}$$

$$P_1 = |1\rangle\langle 1| \tag{1.23}$$

$$P_+ = |+\rangle\langle +| \tag{1.24}$$

$$P_- = |-\rangle\langle -| \tag{1.25}$$

When measuring a photon, Bob has a 50% probability of performing the projective measurement in the basis Alice chose, this is $b_{i,1} = b_{i,3}$. From Equation 1.12, one sees that, in this case, he has 100% probability of measuring the received photon in the state prepared by Alice, and retrieving the correct bit.

$$\begin{cases} p_0 = \langle 0|P_0|0\rangle = \langle 0|0\rangle\langle 0|0\rangle = 1 \\ p_1 = \langle 0|P_1|0\rangle = \langle 0|1\rangle\langle 1|0\rangle = 0 \end{cases} \tag{1.26}$$

$$\begin{cases} p_0 = \langle 1|P_0|1\rangle = \langle 1|0\rangle\langle 0|1\rangle = 0 \\ p_1 = \langle 1|P_1|1\rangle = \langle 1|1\rangle\langle 1|1\rangle = 1 \end{cases} \tag{1.27}$$

However, if they choose different basis, since they are mutually unbiased, from Equation 1.14 we see that Bob has 50% probability of measuring the qubit in either state of his chosen basis, and thus randomly retrieve a bit.

$$\begin{cases} p_+ = \langle 0|P_+|0\rangle = \langle 0|+\rangle\langle +|0\rangle = \frac{1}{2} \\ p_- = \langle 0|P_-|0\rangle = \langle 0|-\rangle\langle -|0\rangle = \frac{1}{2} \end{cases} \tag{1.28}$$

$$\begin{cases} p_+ = \langle 1|P_+|1\rangle = \langle 1|+\rangle\langle +|1\rangle = \frac{1}{2} \\ p_- = \langle 1|P_-|1\rangle = \langle 1|-\rangle\langle -|1\rangle = \frac{1}{2} \end{cases} \tag{1.29}$$

After measuring, Bob and Alice share the strings s_1 and s_3 over an authenticated classical channel. This way, they can know when they chose the same basis $b_{i,1} = b_{i,3}$, and thus Bob is sure of which state Alice prepared, and when they chose different basis $b_{i,1} \neq b_{i,3}$, and so he cannot gain any information about Alice's state. In the end, they keep the bits corresponding to the $b_{i,1} = b_{i,3}$ cases, and discard the others. Table 1.2 represents a possible communication between Alice and Bob.

s_1	0	1	0	1	0	1	1	0	1	1
s_2	0	0	1	0	0	1	1	1	0	1
Alice's Photon	$\|\uparrow\rangle$	$\|\nearrow\rangle$	$\|\downarrow\rangle$	$\|\nearrow\rangle$	$\|\uparrow\rangle$	$\|\searrow\rangle$	$\|\searrow\rangle$	$\|\downarrow\rangle$	$\|\nearrow\rangle$	$\|\searrow\rangle$
s_3	0	0	1	1	1	0	1	0	1	0
Bob's Photon	$\|\uparrow\rangle$	$\|\downarrow\rangle$	$\|\nearrow\rangle$	$\|\nearrow\rangle$	$\|\searrow\rangle$	$\|\uparrow\rangle$	$\|\searrow\rangle$	$\|\downarrow\rangle$	$\|\nearrow\rangle$	$\|\downarrow\rangle$
Sifted Key	0			0			1	1	0	

Table 1.2: Example of communication between Alice and Bob during the BB84 protocol.

If Alice had an initial key of size N, since Bob has a 50% probability of choosing the same basis as her, then after this last step they will share a key with half the initial size $N/2$.

As initially mentioned, the safety of the QKD protocols comes from the introduction of errors in the key when in the presence of eavesdropping. Therefore, after sharing a sifted key between them, Alice and Bob must go through a classical post-processing of the key. Firstly, they use part of the raw data to estimate the error rate in the quantum channel, by calculating parameters such as channel noise and transmissivity. Depending on this parameter, the protocol either aborts or, if it doesn't, Alice and Bob perform error correction and privacy amplification to respectively detect and correct errors in the key and to reduce Eve's possible stolen information to a minimum.

To understand the impact of an eavesdropper, one can start with picturing a very simple hacking strategy, depicted in Figure 1.2. Here, Eve intercepts photons from the quantum channel, measures them in a randomly chosen basis, and resends them to Bob. Like Bob, Eve has a 50% probability of choosing the same basis as Alice. When the wrong basis is chosen, her measurement collapses the qubit's state into one of the MUB states.

$$\begin{cases} \frac{P_+|0\rangle}{\sqrt{\langle 0|P_+|0\rangle}} = |+\rangle \text{ with } p = 50\% \\ \frac{P_-|0\rangle}{\sqrt{\langle 0|P_-|0\rangle}} = |-\rangle \text{ with } p = 50\% \end{cases} \tag{1.30}$$

$$\begin{cases} \frac{P_+|1\rangle}{\sqrt{\langle 1|P_+|1\rangle}} = |+\rangle \text{ with } p = 50\% \\ \frac{P_-|1\rangle}{\sqrt{\langle 1|P_-|1\rangle}} = |-\rangle \text{ with } p = 50\% \end{cases} \tag{1.31}$$

If Bob now chooses the same basis as Alice, instead of retrieving the correct bit with an 100% probability, he now only has a 50% probability, because instead of receiving the qubit encoded in the correct basis, he is receiving it encoded in the MUB one. Of course when Eve chooses the same basis as Alice, her measurement collapses the photon in its initial state and so no errors are introduced in the key. Table 1.3 summarizes Eve's error introduction in the sifted key, and from it one concludes that with this strategy, she introduces a $0.5 * 0.5 = 0.25$ error rate.

Alice	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$
Eve	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$
Bob	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$
Error fraction	0	0.5	0.5	0

Table 1.3: Error fraction introduced in the key for the four possible cases which make up the sifted key.

In this simplified example where there aren't other sources of errors, Alice and Bob, after determining the error rate, would detect the presence of Eve and discard the sifted key, restarting the protocol after removing the eavesdropper.

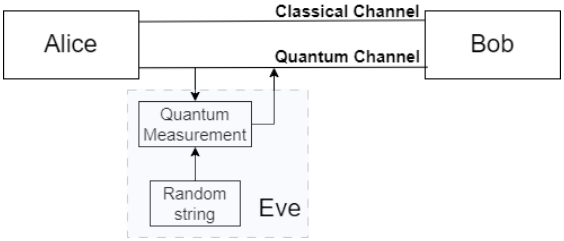


Figure 1.2: Simple Intercept and Resend Attack.

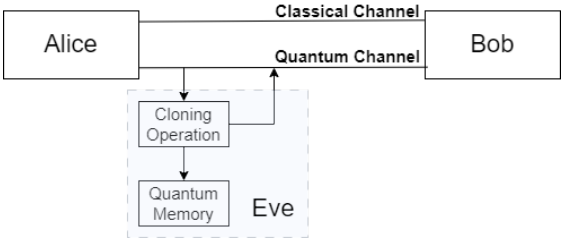


Figure 1.3: Intercept and Resend Attack with a Cloning Operation.

It's worth noticing that, if Eve was able to copy each qubit state into an ancilla system, then she could elevate the intercept and resend attack to get full information about the key without being noticed. As depicted in Figure 1.3, she would simply retrieve the photons from the quantum channel, copy their state to another quantum system, and wait for Bob to reveal his basis choices over the classical channel to make her measurements on the stored photons. This copying of quantum states is, however, prohibited by the No-Cloning Theorem [30], which states that a single unitary operation U cannot clone an arbitrary quantum state. Therefore, theoretically, any attempt from Eve to get information about the key increases the error rate.

Even though the intercept-and-resend strategy is an intuitive example for the claim that an eavesdropper must introduce errors in the key in order to retrieve information, a broader argument can be made in order to justify it. We start by considering Eve which, similarly to the intercept-and-resend strategy, retrieves the photons Alice is sending to Bob over the quantum channel. But now, instead of measuring them in a random basis, she attaches an ancillary system $|E\rangle$ to each of them [31]. For example, from two arbitrary Alice states $|\psi\rangle$ and $|\phi\rangle$, Eve would get $|E\rangle\,|\psi\rangle$ and $|E\rangle\,|\phi\rangle$. Given that her objective is to gain information about the states $|\psi\rangle$ and $|\phi\rangle$ without disturbing them, she applies a unitary operation to the system which only changes her ancillary state:

$$\begin{aligned} U\,|E\rangle\,|\psi\rangle &= |E_\psi\rangle\,|\psi\rangle \\ U\,|E\rangle\,|\phi\rangle &= |E_\phi\rangle\,|\phi\rangle \end{aligned} \tag{1.32}$$

However, from the scalar product calculated in Equation 1.33, we can conclude that with this strategy, Eve's states $|E_\psi\rangle$ and $|E_\phi\rangle$ are identical, and Eve cannot gain any information about Alice's state

$$\langle E|E\rangle\,U^\dagger U\,\langle\psi|\phi\rangle = \langle E_\psi|E_\phi\rangle\,\langle\psi|\phi\rangle \Leftrightarrow \langle E_\psi|E_\phi\rangle = 1 \tag{1.33}$$

Let's now consider a unitary operation which also disturbs Alice's states:

$$\begin{aligned} U\,|E\rangle\,|\psi\rangle &= |E_\psi\rangle\,|\psi'\rangle \\ U\,|E\rangle\,|\phi\rangle &= |E_\phi\rangle\,|\phi'\rangle \end{aligned} \tag{1.34}$$

Now from the scalar product in Equation 1.35, we can see that Eve's states $|E_\psi\rangle$ and $|E_\phi\rangle$ are no longer identical for $|\psi\rangle \neq |\phi\rangle$, thus Eve can gain information about the key.

$$\langle E|E\rangle\,U^\dagger U\,\langle\psi|\phi\rangle = \langle E_\psi|E_\phi\rangle\,\langle\psi'|\phi'\rangle \Leftrightarrow \langle E_\psi|E_\phi\rangle = \frac{\langle\psi|\phi\rangle}{\langle\psi'|\phi'\rangle} \tag{1.35}$$

To get more information, Eve must increase the distinguishability between her states, this is, decrease $\langle E_\psi|E_\phi\rangle$. However by doing this, she increases $\langle\psi'|\phi'\rangle$, which means introducing more disturbance in Alice's states. Therefore, from this simple theoretical exercise, one can understand how retrieving information from the quantum channel implies disturbing the initial states.

Nevertheless, implementing the BB84 protocol presents challenges not considered in this simple theoretical explanation. The impact they have on the protocol's security will be covered in later sections

but before, it is necessary to understand what these challenges comprise of and, consequently, how to implement a DV QKD system.

1.3.3 Implementation of Discrete Variable QKD systems

In an ideal implementation of the BB84 protocol, Alice uses a perfect single photon source, which emits indistinguishable photons into a given spatio-temporal mode with 100% probability and with arbitrary emission frequency. Then, she sends the photon to Bob through a lossless quantum channel, who in turn uses ideal single photon detectors to measure them. Unfortunately, these idealizations are not practical and in an experimental implementation of QKD, both parties must find a compromise between theoretical requirements and what is achievable given the state of the art.

Single photon emission techniques can be divided into probabilistic and deterministic [32]. As conveyed by their name, the former rely on probabilistic phenomena, such as Spontaneous Parametric Down Conversion (SPDC) and Four-Wave Mixing, where a laser excitation of a non-linear optical material creates pairs of correlated photons. This pair production is very useful because one of the photons of the pair can be used to herald the photon produced by the source. The X^2 non-linearity needed for SPDC can be found in crystals [33] or in single-mode waveguides [34], whilst X^3 non-linearity required for four-wave mixing is typically found in single mode optical fibers [35]. Even though commercially available, the disadvantages of probabilistic sources have prevented a widespread implementation in QKD setups. Their statistical nature, combined with multiple pair emission probability increasing when single pair emission probability is increased, means that they have reduced emission frequencies. Since this has a direct impact on the overall key rate of the setup, higher rate solutions are preferred.

On the other side, deterministic sources rely on quantum systems with two energy levels. Single neutral atoms [36] and quantum dots [37], among many others, are examples of physical systems which can be externally controlled and put into an excited state, producing only a single photon upon returning to the ground energy level. However, different issues, such as decoherence in neutral atom sources, scalability issues and cryogenic temperature requirements in quantum dots [32], have prevented these sources from being a staple in QKD implementations.

With the current state of art, most QKD setups have resorted to weak coherent states created by lasers as an approximation to single photons. In order to understand coherent states, and their implications in the security of QKD protocols, we must first introduce the concept of Fock States. Quantum states with a well-defined number of particles can be described by Fock States, quantum states in a Fock space [38], which comprise the occupancy number basis. An arbitrary Fock state is written as $|n_{k_1}, n_{k_2}, ..., n_{k_i}\rangle$, where n_{k_i} are the number of particles in the state k_i, and it is an eigenstate of the number operator N_{k_i}

$$N_{k_i} |n_{k_1}, n_{k_2}, ..., n_{k_i}\rangle = n_{k_i} |n_{k_1}, n_{k_2}, ..., n_{k_i}\rangle \tag{1.36}$$

Therefore, states produced by single photon sources can be described by the Fock State $|1\rangle$. However, electromagnetic waves created inside an optical laser cavity cannot be described this way, hence, their study requires the introduction of coherent states $|\alpha\rangle$.

$$|\alpha\rangle = \sum_n c_n |n\rangle = \sum_n \frac{\alpha^n}{\sqrt{n!}} e^{-\frac{|\alpha|^2}{2}} |n\rangle \qquad (1.37)$$

$$P_n = |c_n|^2 = \frac{\mu^n}{n!} e^{-\mu} \qquad (1.38)$$

Coherent states are a weighted superposition of Fock States [39], as explicit in Equation 1.37. They can be used to ideally describe coherent light produced by lasers, and the probability P_n of having a state with n photons follows the Poisson distribution (Equation 1.38), where $\mu = |\alpha|^2$ is the average number of photons.

As aforementioned, in QKD setups, single photon sources are typically replaced by coherent state sources, such as lasers. To do so, increasing the probability of having one photon per beam P_1 whilst decreasing $P_{n>1}$ is crucial. From Figure 1.4 one can see that, to achieve this, the coherent wave must be highly attenuated, such that $\mu \ll 1$. Consequently, the probability of having 0 photons per state is high $P_0 = e^{-\mu} \approx 1 - \mu$, reducing the efficiency of weak coherent state sources.

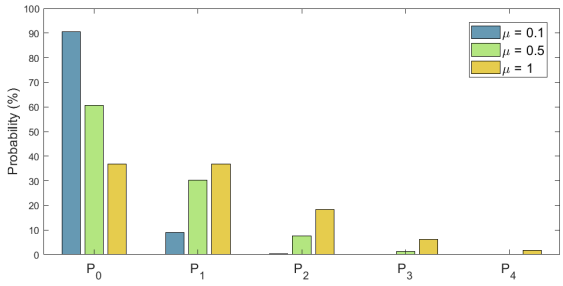


Figure 1.4: Poisson Statistics for different μ values.

Additionally, the probability of having more than one photon per state, albeit small for $\mu \ll 1$, is non-vanishing, independently of how attenuated the beam is, as seen by Equation 1.39. As we will see later, this was one of the first vulnerabilities to be explored by a side channel attack.

$$P(n > 1 | n > 0) = \frac{1 - P_1 - P_0}{1 - P_0} \approx \frac{\mu}{2} \qquad (1.39)$$

After preparing her photon, Alice must send it to Bob over a quantum channel. The choice over this channel is extremely important, given that it has a direct impact on the trade-off between the key rate and the distance of the key exchange. Any point to point QKD link is limited by the Pirandola-Laurenza-Ottaviani-Banchi (PLOB) bound [40]. This upper bound states that, given a bosonic lossy channel with transmissivity η, and two parties which may apply arbitrary local operations, assisted by unlimited two-way classical communication, the maximum achievable rate for key generation is $-\log_2(1 - \eta)$. As the transmissivity of the channel decreases with distance, this bound constraints the key rates for long-distance QKD.

However, channel loss doesn't solely depend on the distance between parties. Among different factors, it depends on the wavelength of the shared photons, the choice of information support (e.g. polarization versus phase encoding), and the type of channel (e.g. free-space versus fiber). For example, even though polarization encoding seems like a logical choice, the birefringence of optical fiber, arising mainly from external stresses applied to the fiber, but also from small asymmetries in the fiber's core cross-section, combined with a possible wide spectral-range of incoming photons, creates rapid depolarization.

Therefore, in order to achieve QKD for large distances, phase encoding is typically preferred. In optical fiber quantum channels, telecommunication wavelengths are implied, and the associated losses are wavelength dependent: typically 2 dB/km for $\lambda = 850$ nm, 0.3 dB/km for $\lambda = 1300$ nm and 0.15 dB/km for $\lambda = 1500$ nm. On the other hand, in free-space links, photons with smaller wavelengths, in the $700 - 900$nm range, are typically preferred.

Upon receiving the photon, Bob must detect it. For this, he requires a single photon detector, which plays a crucial role on the performance of the key exchange. These detectors are characterized by a set of parameters, typically temperature and wavelength dependant, whose value allow for an overall view of the detector's performance. Among them we identify the following:

- Detector's efficiency: translates the probability that an incident photon is detected.
- Dark count rate: caused by the detector firing when no photon is present.
- Dead time: time after an event when the detector is unable to detect any photon.
- Timing jitter: fluctuation between events of the time interval between the photon's arrival and the electrical output signalling a detection
- Photon number resolution: whether the detector is able to distinguish the number of photon in incoming pulses.

Photomultipler tubes pioneered the field of single photon detectors. In these detectors the incoming photon pulse is directed onto a photocathode causing the emission of a single electron. As depicted in Figure 1.5, this primary electron is then accelerated towards the first dynode through the focusing electrode, where it produces more electrons via secondary emission, which are then accelerated to the second dynode, and so forth. With low dark count rates, timing jitters and dead time, the routine deployment of these detectors is hindered by their efficiency rates [32].

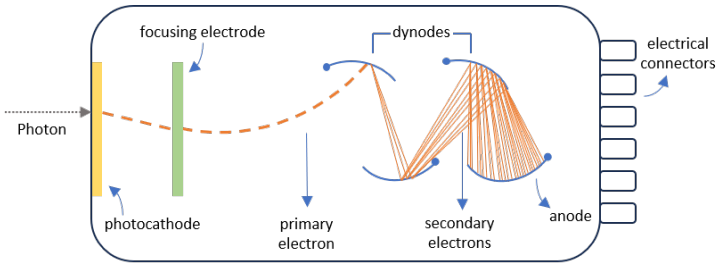


Figure 1.5: Schematics of a Photomultiplier Tube.

Superconductor-based technologies, such as Superconducting Nanowire Single-Photon Detectors (SNSPDs), offer a promising solution for high efficiency and fast response detection, with ultra-low dark counts, in the infrared and visible wavelengths [41]. In these detectors, nanowires typically made out of NbN, and with widths around 100 nm, are biased just below critical current density. Individual photons impinging on the superconductor disrupt hundreds of Cooper pairs, forming a small hotspot region. The current flows around this higher resistance region, which in turn increases the current density in the adjacent areas above the critical point, forming a resistance barrier across the width of the nanowire. The

fast resistance increase causes a voltage spike, representing the detection of a photon. Operating at 4.2K, SNSPDs are usually cooled by immersion in liquid Helium, which is a hazardous and expensive procedure, keeping these high efficiency detectors from common deployment in QKD networks.

Until now, the single photon avalanche photodiode (SPAD) has been the mainstream solution for detection in QKD due to its compact size, cost-effectiveness and easy operation [42]. Using a pn junction in Geiger mode, this is, biased well above the breakdown voltage, when an incoming photon strikes the detector, it produces an electron-hole pair, as schematized in Figure 1.6. This electron is then accelerated to a kinetic energy above the ionization energy of the bulk material. Therefore, a photon inserted in the depletion region generates a self-sustained avalanche current, which must be terminated by lowering the bias voltage below the breakdown value. This quenching is typically accomplished by an external circuit which generates a well synchronized output pulse upon recognizing the onset of the avalanche current, and it is crucial for the detector's performance.

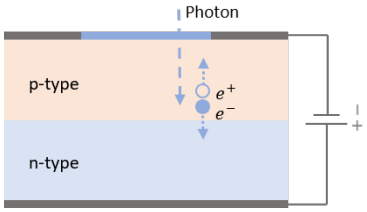


Figure 1.6: Schematics of a SPAD

The absorption material, namely its energy gap E_g, dictates the photon wavelengths for which the detector is sensitive to. Detection at the telecommunication wavelength, $\lambda = 1550$nm, requires lower band-gap semiconductors like InGaAs, with $E_g = 0.75eV$, which are typically matched with a InP multiplication layer. Commercially available InGaAS-InP SPADs have a detection efficiencies around 20% and dark count rates of 500 cps, small compared to the efficiencies of SNSPDs, which have reached record values of 93% [43]. They suffer also from afterpulsing, a non-ideal behaviour caused by carriers getting trapped in certain sites in the multiplication layer during the avalanche. If these trap sites are not given time to depopulate after the quenching, then these carrier can be released, causing fake avalanches, thus, fake detections. The afterpulsing probability increases the waiting time needed before rebiasing the detector, causing increased dead times [32]. Nonetheless, continuous efforts to enhance InGaA-InP SPAD's performances have culminated in record values such as 60% efficiencies [43].

Having summarized the main devices and implementation problems in DV-QKD, it is clear that translating the theoretical simplicity of protocols to practical systems is a difficult task. Nevertheless, the field has witnessed extraordinary progress. The first QKD implementation, done by Bennet and Brassard, with the help of three students, aimed at disproving the skepticism surrounding it, used a free-space quantum channel with 30cm to extract a final key of 403 bits from 64516 light pulses [44]. In recent years, 1002 km distances [7] and Mbit/s rates [8] have since been reached, and efforts for improvement are ongoing. However, vulnerabilities introduced by physical flaws, known as side-channels, are still an important concern. Opening security breaches not initially considered in the theoretical security proofs, side-channel attacks are a ubiquitous problem to QKD, and will be further discussed in the next chapter.

Chapter 2

Security in QKD

Theoretically proving the security of QKD protocols is crucial for their implementation and role in current cryptographic settings. For the purpose of this thesis, it is important to understand the role assumptions on physical devices play in security proofs of QKD protocols, and how they can lead to unexpected security breaches. Therefore, this chapter will cover different security aspects regarding QKD and will overview state-of-the-art quantum hacking attacks for prepare-and-measure protocols.

2.1 Security Definition

Intuitively, to prove that a QKD protocol is secure, one wants to assure Eve's information on the key is negligible when taking into account the most powerful attacks she can do. Naively, this can lead to quantifying security via the mutual information between K, a classical variable representing the key shared between Alice and Bob, and W, a random variable describing Eve's measurement outcomes [31]

$$\max_{W} I(K:W) \leq \epsilon. \tag{2.1}$$

A crucial requirement for QKD protocols is universal security, which determines that the key is secure in arbitrary contexts. Importantly, since keys are used by symmetric encryption protocols, universal security entails composability, meaning that the key can be used as a part of a bigger system. Unfortunately, the criterion in Equation 2.1 does not guarantee universal security, so another approach must be taken.

Typically, security claims for QKD protocols are phrased within the framework of Abstract Cryptography [45], which is mostly independent from the modelling of devices, and can be used in classical and quantum settings. In this framework, security is established by defining an ideal protocol $\mathcal{I}$, perfectly secure by construction, and proving that the real protocol $\mathcal{R}$ is indistinguishable from it. This is computed by the distinguisher, an agent who has access to all inputs and outputs of Alice, Bob and Eve, and can use any strategy to distinguish between the real and the ideal protocol. Let us assume we have two states, ρ and σ, which are given to a distinguisher with 50% probability, to distinguish. It can be proven that the amount

by which the distinguisher can surpass random guessing is given by the distinguishing advantage

$$D(\rho,\sigma)=\frac{1}{2}\|\rho-\sigma\|_1, \tag{2.2}$$

where $\|\rho\|_1=\mathrm{Tr}\left(\sqrt{\rho^\dagger\rho}\right)$ is the trace norm of the state ρ. For the real and ideal protocols, it is said that $\mathcal{R}\approx_\epsilon\mathcal{I}$ if they have a distinguishing advantage of at most ϵ,

$$\mathcal{R}\approx_\epsilon\mathcal{I}\quad\text{if}\quad p_{\text{distinguish}}(\mathcal{R},\mathcal{I})=\frac{1}{2}+\frac{1}{2}\epsilon. \tag{2.3}$$

This is called the distinguishing advantage criterion, equivalent to writing $d(\mathcal{R},\mathcal{I})\leq\epsilon$, and it means the protocol is ϵ-secure. This criterion can be written in terms of ρ_{ABE} and $\tilde{\rho}_{ABE}$, the states shared between Alice, Bob and Eve in the real and the ideal protocol, respectively,

$$D(\rho_{ABE},\tilde{\rho}_{ABE})\leq\epsilon. \tag{2.4}$$

Note that this definition of security englobes composability since given two protocols, one ϵ_1-secure and the other ϵ_2-secure, it can be proven that together they are $(\epsilon_1+\epsilon_2)$-secure, for both sequential and parallel composition [31]. Typically, for a QKD protocol, the criterion in Equation 2.4 is divided into two parts, the correctness of the key and the secrecy.

For the definition of correctness, we consider K_A and K_B, random variables which describe Alice's and Bob's respective key bit strings, whose realizations k_A and k_B belong to the space $\mathcal{K}\cup\perp$, where $k_A=k_B=\perp$ represent an aborted protocol, $\mathcal{K}=\{0,1\}^\ell$ the key space, and ℓ the key size when the protocol is not aborted. The protocol is said to be ϵ_{corr}-correct if the probability that Alice and Bob do not abort the protocol whilst having different generated keys is small,

$$\Pr[K_A\neq K_B]\leq\epsilon_{\text{corr}}. \tag{2.5}$$

Defining secrecy requires comparing the real key to an ideal key, a key which is uniformly distributed and independent of the adversary's information. To do so, we consider the distinguishing advantage between the classical-quantum state shared between Alice and Eve in the real scenario ρ_{AE} and in the ideal one $\tilde{\rho}_{AE}$, $D(\rho_{AE},\tilde{\rho}_{AE})$. Note that this analysis can be also done for Bob, since his key is ϵ_{corr} likely to be identical to Alice's.

Firstly, we consider the two possible cases within the protocol, this is, either it is aborted, with probability $p_\perp$, either it is not. For example, the shared state in the real scenario can be written as $\rho_{AE}=p_\perp\rho_{AE}^\perp+(1-p_\perp)\rho_{AE}^\top$, where $\top$ represents the scenario when the protocol is not aborted. As aforementioned, when the protocol is aborted, $k_A=|\perp\rangle$ for both the ideal and real scenarios. Regarding Eve, Alice and Bob aborting the protocol does not affect her information, and so her state is also the same for both scenarios. Therefore $\rho_{AE}^\perp=\tilde{\rho}_{AE}^\perp=p_\perp|\perp\rangle\langle\perp|\otimes\rho_E^\perp$.

Considering the protocol is not aborted, in the real scenario, Eve's system, described by the quantum state $\rho_E^{k_A}$, depends on the realization k_A of Alice's key K_A, which is distributed according to the probability

distribution p_{K_A}. Therefore, we can write

$$\rho_{AE}^{\top} = \sum_{k_A \in \mathcal{K}} p_{K_A} |k_A\rangle_A\langle k_A| \otimes \rho_E^{k_A}. \tag{2.6}$$

In the ideal scenario, Eve's information is independent of Alice's key, and her system described by ρ_E. Alice's key is uniformly distributed, thus described by a maximally mixed state $\rho_U = \sum_{u \in \mathcal{K}} \frac{1}{|\mathcal{K}|} |u\rangle\langle u|$, and the whole system described by

$$\tilde{\rho}_{AE}^{\top} = \sum_{u \in \mathcal{K}} \frac{1}{|\mathcal{K}|} |u\rangle_A\langle u| \otimes \rho_E. \tag{2.7}$$

Finally, we can write

$$\begin{aligned} \rho_{AE} &= p_\perp \, |\perp\rangle\langle\perp| \otimes \rho_E^{\perp} + (1 - p_\perp) \sum_{k_A \in \mathcal{K}} p_{K_A} |k_A\rangle_A\langle k_A| \otimes \rho_E^{k_A}, \\ \tilde{\rho}_{AE} &= p_\perp \, |\perp\rangle\langle\perp| \otimes \rho_E^{\perp} + (1 - p_\perp) \sum_{u \in \mathcal{K}} \frac{1}{|\mathcal{K}|} |u\rangle_A\langle u| \otimes \rho_E. \end{aligned} \tag{2.8}$$

For the distinguishing advantage, we can use the triangle inequality of the trace norm together with the indistinguishability between $\rho_{AE}^{\perp}$ and $\tilde{\rho}_{AE}^{\perp}$ to obtain

$$\begin{aligned} D(\rho_{AE}, \tilde{\rho}_{AE}) = \frac{1}{2}\|\rho_{AE} - \tilde{\rho}_{AE}\|_1 = \frac{1}{2}\|p_\perp(\rho_{AE}^{\perp} - \tilde{\rho_{AE}}^{\perp}) + (1 - p_\perp)(\rho_{AE}^{\top} - \tilde{\rho_{AE}}^{\top})\|_1 = \\ \leq \frac{p_\perp}{2}\|\rho_{AE}^{\perp} - \tilde{\rho_{AE}}^{\perp}\|_1 + \frac{1 - p_\perp}{2}\|\rho_{AE}^{\top} - \rho_U \otimes \rho_E\|_1 \\ \leq \frac{1 - p_\perp}{2}\|\rho_{AE}^{\top} - \rho_U \otimes \rho_E\|_1. \end{aligned} \tag{2.9}$$

Therefore, a protocol is said to be ϵ_{sec}-secret if, for all adversarial strategies, it satisfies

$$\frac{1 - p_\perp}{2}\|\rho_{AE}^{\top} - \rho_U \otimes \rho_E\|_1 \leq \epsilon_{\text{sec}}. \tag{2.10}$$

Equipped with the framework for correctness and secrecy, we can now define the condition for ϵ-security from Equation 2.4 in a more precise manner. In the ideal protocol, k_A and k_B are identical and uniformly distributed, such that the state shared between Alice and Bob is given by $\rho_{UU} = \frac{1}{|\mathcal{K}|} \sum_{u \in \mathcal{K}} |u\rangle_A\langle u| \otimes |u\rangle_B\langle u|$. In the real protocol, the state shared between them depends on the probability distribution $p_{K_A,K_B}(k_a, k_b)$, entailing how the two key variables are distributed, and Eve's state is given by $\rho_E^{k_A,k_B}$. The security condition can be then written as a function of the state $\rho_{ABE}^{\top}$, which is given by

$$\rho_{ABE}^{\top} = \frac{1}{1 - p_\perp} \sum_{k_A,k_B \in \mathcal{K}} p_{K_A,K_B}(k_a, k_b) |k_A\rangle_A\langle k_A| \otimes |k_B\rangle_B\langle k_B| \otimes \rho_E^{k_A,k_B}. \tag{2.11}$$

A protocol is ϵ-secure if, for all adversary strategies, it satisfies

$$\frac{1 - p^{\perp}}{2}\|\rho_{ABE}^{\top} - \rho_{UU} \otimes \rho_E\|_1 \leq \epsilon. \tag{2.12}$$

From the condition in Equation 2.12 it can be proven, as done in [31], that if a protocol is ϵ_{corr}-correct and

ϵ_{sec}-secret, then it is ϵ-secure with $\epsilon = \epsilon_{\text{corr}} + \epsilon_{\text{sec}}$. Also, this condition allows for an intuitive comparison between the ideal and real key, depicted in Figure 2.1. The ideal key U has a uniform distribution p_U, and so each realization u has a probability 2^{ℓ} of happening, where ℓ is the size of the key. For a real key K, generated by an ϵ-secure protocol, not all realizations k happen with the same probability. The ϵ parameter describes the amount by which we must shift the probabilities $p_K(k)$ to retrieve the uniform distribution.

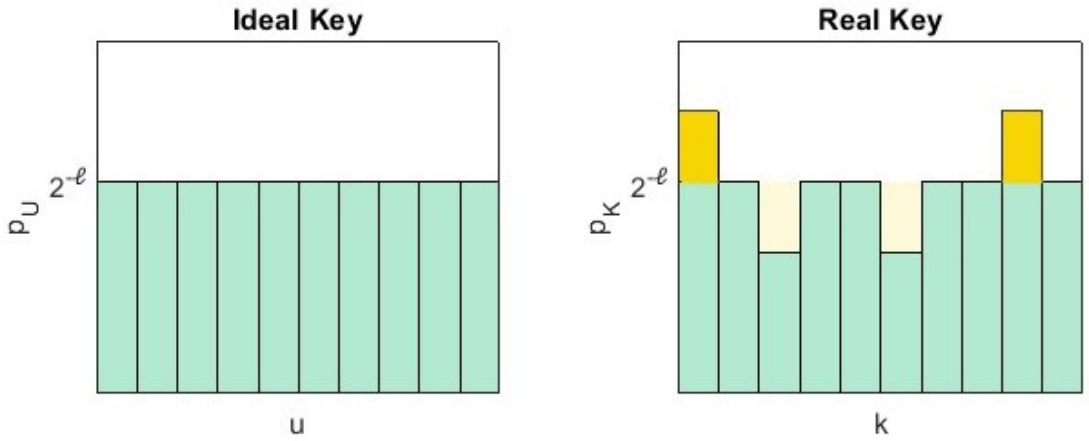


Figure 2.1: Schematics of key realizations in an ideal protocol and in a real protocol.

2.2 Assumptions

As we have seen, security proofs for QKD protocols rely on the modelling of the real protocol, so that indistinguishability from the ideal protocol can be assured. More than assuming that Quantum Mechanics is correct and complete, and that authenticated communication between parties is possible, assumptions regarding the physical implementation of the protocol are typically required. Knowing what these are is crucial to understand the feasibility of the security proof, given that any deviation of the theoretical modelling from the actual physical implementation represents a possible security threat.

From the exhaustive list of assumptions typically made by security proofs, here we present a short list, taken from [31], with some which are commonly found to be different from the actual implementations.

1. State Preparation: it is assumed Alice always prepares the desired quantum states, which in a practical setting, is not always true, and may cause the actual states to differ from the theoretical expected ones.

2. Measurement Devices: typically the modelling of these devices doesn't take into consideration imperfections such as losses or the measuring of states outside the desired Hilbert space.

3. Timing: in order for Alice and Bob to compare their strings bit by bit, which as we will see, is necessary for post processing, they need a shared timing method that allows them to associate the sent state to the correct measurement outcome.

4. Isolation of Alice and Bob's Laboratories: it is assumed that Eve is unable to tamper with state preparation and measurement devices, and cannot access any of their outputs except for the state in the quantum channel.

5. Post Processing: as we will in the next section, in the post processing phase, the amount of information which has been leaked to Eve is estimated. If the security proof does not account for differences between the physical implementation and the modelling, it can lead to an underestimation.

2.3 Post-Processing

After the Quantum transmission, where Alice shared with Bob M symbols over a quantum channel, they are each left with a key that is partially-secret and partially-correct. However, as seen in the security definition, they wish to share an ϵ_{corr}-correct and ϵ_{sec}-secret key. To achieve this, they resort to classical post-processing techniques.

The first step is parameter estimation, where they estimate the error rate in the key in order to understand how much information has been leaked to Eve during the quantum transmission. This step can be considered in the *asymptotic-key* scenario, where one considers an infinite number of shared symbols $M \to \infty$, or in the more realistic *finite-key* scenario, where $M < \infty$. In the finite key scenario, which will be considered for the purpose of this discussion, one defines the secret key rate as

$$r = \frac{\ell}{M}, \tag{2.13}$$

where ℓ is the length of the secret key. To calculate this value, they use a small sample, with dimension k, of their sifted bit strings, with length $k < M$. For example, Alice sends her string sample, K_A^k, to Bob, who compares it to his sample K_B^k, and calculates the error rate

$$\Lambda_k = \frac{1}{k}|K_A^k \oplus K_B^k|, \tag{2.14}$$

where $|K_A^k \oplus K_B^k|$ represents the Hamming weight, this is, the number of ones in the string $K_A^k \oplus K_B^k$. If this value is greater than a certain threshold λ_{max}, the protocol is aborted, and if not, they use it to estimate the global error rate. Using Sterfling's inequality and Bayes Theorem, it can be proven that the error rate λ_n, in the remaining $n = M - k$ bits, has an exponentially small probability in sample size k, of being incompatible with the estimated parameters [31],

$$\text{Pr}[\Lambda_n \geq \Lambda_k + \gamma | \lambda_k < \lambda_{\text{max}}] \leq \frac{e^{\frac{2k^2 n\gamma^2}{(k+1)N}}}{\text{Pr}[\lambda_k \leq \lambda_{\text{max}}]}. \tag{2.15}$$

After parameter estimation, Alice and Bob either abort the protocol, or they have partially correct raw keys of size n, to which they know an estimate of the error rate, but do not know where the errors lie. To remove them, they use classical information reconciliation, which turns two strings with correlations into two identical ones. For simplicity, here we consider direct reconciliation, where Alice sends information

about her string to Bob and he uses it to correct his key. The choice of the error correcting code dictates the amount of communication necessary between both parties. Nonetheless, after this correction, Alice and Bob need to make sure that the procedure was successful. For that they use a two-universal hash function f. These functions belong to $\mathcal{F}$, a family of functions from an alphabet $\mathcal{X}$ to an alphabet $\mathcal{Z}$ and they obey

$$\Pr_{f\in\mathcal{F}}[f(x)=f(x')]\leq\frac{1}{|\mathcal{Z}|},\ \text{ for any }\ x,x'\in\mathcal{X}\ \text{ with }\ x\neq x'. \tag{2.16}$$

To check the correctness of their key string, Alice chooses a two-universal hash function f_{EC}, applies it to her string, sends it to Bob, together with the result she obtained $f_{EC}(K_A)$, and he then applies it to his string $f_{EC}(K_B)$. If Alice chooses f_{EC} from a family $\mathcal{F}$ with output space cardinality $|\mathcal{Z}|=2^{\lceil\log_2\frac{1}{\epsilon_{\text{corr}}}\rceil}$, then it follows from Equation 2.16 that

$$\Pr[f_{EC}(K_A)=f_{EC}(K_B)|K_A\neq K_B]\overset{\text{def}}{\leq}\frac{1}{|\mathcal{Z}|}=2^{-\lceil\log_2\frac{1}{\epsilon_{\text{corr}}}\rceil}\leq 2^{-\log_2\frac{2}{\epsilon_{\text{corr}}}}\leq\epsilon_{\text{corr}}. \tag{2.17}$$

Applying Bayes Theorem, knowing that the protocol aborts if the hashes differ, we retrieve

$$\Pr[K_A\neq K_B|f_{EC}(K_A)=f_{EC}(K_B)]\leq\epsilon_{corr}, \tag{2.18}$$

showing that if the checking procedure succeeds except with probability ϵ_{cor}, then the whole protocol is ϵ_{cor}-correct.

After error correction, Alice and Bob share an ϵ_{corr}-correct, partially secret key. In the final state of post processing, called privacy amplification, the objective is to remove any residual information which might have been leaked to Eve over the quantum channel as well as the authenticated classical channel during information reconciliation. To do so, parties can resort to random extractors. These are functions Ext which receive as input a string X, of length n, as a source of randomness, and a small uniformly random seed S, of length d, to compute the output Z, a string whose length ℓ is bigger that d,

$$\text{Ext}(X,S)=Z. \tag{2.19}$$

In a QKD setting, the randomness extract must fulfill two requirements. Firstly the output string Z must be independent from the seed S, given that Alice must publicly announce S to Bob over the classical channel. Secondly, it must be able to extract randomness against a quantum adversary. Understanding whether this is possible requires quantifying the amount of information Eve has on Alice's key, to know if it is limited. For the classical-quantum state ρ_{XE}, shared between Alice's classical key variable X and Eve's quantum system E,

$$\rho_{XE}=\sum_{x in\mathcal{X}}p_X(x)\,|x\rangle\langle x|\otimes\rho_E^x, \tag{2.20}$$

the conditional min-entropy $H_{\text{min}}(X|E)$ quantifies how much uniform randomness $Z=\text{Ext}(X,S)$ can be extracted from X such that Z is independent from E.

The randomness extractors which fulfill these requirements are called Quantum-proof Strong Randomness Extractors. A function Ext : $\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^\ell$ is a (t,ϵ)-strong quantum-proof randomness

extractor if, for all classical-quantum states ρ_{XE} with $H_{\min}(X|E) \geq t$, we have

$$D(\rho_{\text{Ext(X,S)SE}}, \frac{\mathbb{1}}{2^\ell} \otimes \rho_S \otimes \rho_E) \leq \epsilon, \tag{2.21}$$

in which the state $\frac{\mathbb{1}}{2^\ell} \otimes \rho_S \otimes \rho_E$ represents the ideal case. The output key Z, uniformly distributed such that every string of length ℓ has an equal probability $\frac{1}{2^\ell}$, is independent from the seed state ρ_S and from Eve's system state ρ_E.

The two-universal hash functions aforementioned are quantum proof strong randomness extractors, and so for privacy amplification, Alice can randomly choose a function f_{PA} from $\mathcal{F}$, for which she uses the seed S, and apply it to K_A, to obtain the final key $f_{PA}(K_A)$. Notice that K_A is the product of the error correction procedure, which means it is $1 - \epsilon_{\text{corr}}$ likely to be identical to K_B, thus this is trivially extended to Bob.

Whilst $H_{\min}(X|E)$ assumes output Z indistinguishable from a uniformly distributed variable, it is useful to extend the previous considerations to allow for some errors in the final key. To do so, we require the concept of quantum conditional smooth mean entropy $H^\epsilon_{\min}(X|E)$, which allows the output to be ϵ distant form the uniform distribution. Being $\mathcal{B}(\rho)$ an ϵ-ball around the state ρ, the smooth min-entropy can be written as

$$H^\epsilon_{\min}(X|E) = \max_{\tilde{\rho} \in \mathcal{B}(\rho)} H_{\min}(X|E)_{\tilde{\rho}}. \tag{2.22}$$

The Quantum Leftover Hash Lemma allows us to further generalize our results, by making use of $H^\epsilon_{\min}(X|E)$. Let $\rho_{f_{PA}(K_A)SE}$ be the state shared between Alice's final key $f_{PA}(K_A)$, the seed state and Eve's system, then for every $\epsilon' > 0$ it holds that

$$D(\rho_{f_{PA}(K_A)SE}, \frac{\mathbb{1}}{2^\ell} \otimes \rho_{SE}) \leq 2\epsilon' + \frac{1}{2}\sqrt{2^{\ell - H^{\epsilon'}_{min}(K_A|E)}}. \tag{2.23}$$

Equipped with this Lemma, requiring the protocol is ϵ_{sec}-secure allows us to extract an upper bound for the final key length ℓ,

$$D(\rho_{f_{PA}(K_A)SE}, \frac{\mathbb{1}}{2^\ell} \otimes \rho_{SE}) \leq \epsilon_{\text{sec}} \Leftrightarrow \ell \leq H^{\epsilon'}_{\min}(K_A|E) + 2 - 2\log_2\left(\frac{1}{\epsilon_{\text{sec}} - 2\epsilon'}\right). \tag{2.24}$$

In summary, post-processing shapes the security proof of a QKD protocol, with every step accounting for a security parameter that contributes to the total failure of the protocol. With this, we can now tackle the secret key fraction, and how it is impacted by eavesdropping.

2.4 Secret Key Fraction

In the *asymptotic-key* scenario, where an infinite amount of shared symbols is assumed, the secret key fraction r is defined as

$$r = \lim_{n \to \infty} \frac{\ell}{n}, \tag{2.25}$$

where ℓ is the size of the final key, after post-processing, and n the length of the raw key, this is, the key after parameter estimation. Security proofs must provide an explicit expression for the secret key fraction, which can the be turned into the final secret key rate S_∞ by multiplying it by R_{raw}, the rate at which the raw key is created,

$$S_\infty = R_{\text{raw}} r. \tag{2.26}$$

As previously seen, before error correction the keys are not ϵ_{corr}-correct. At this stage, we can define the Quantum Bit Error Rate (QBER) δ, as

$$\delta = \frac{\text{number of different bits in the sifted key}}{\text{total number of bits in the sifted key}}. \tag{2.27}$$

and use it to describe the channel between Alice and Bob as a binary symmetric channel, where the binary input a is chosen with $\frac{1}{2}$ probability, and the binary output b has a δ probability of being different from a. This comparison simplifies the calculation of the mutual information between Alice and Bob,

$$I(A:B) = H(B) - H(B|A) = -\sum_{b=0,1} p(b)\log_2(p(b)) + \sum_{a,b=0,1} p(a,b)\log_2(p(b|a)) = 1 - h_2(\delta), \tag{2.28}$$

where $h_2(\delta) = -\delta\log_2(\delta) - (1-\delta)\log_2(1-\delta)$ is the Binary Entropy of the QBER. Actually, $I(A:B)$ is an upper bound for the fraction of perfectly correlated symbols that can be extracted from a list of partially correct ones [46]. With this being said, Equation 2.28 takes on a more intuitive interpretation: if the sifted key had no errors, then $h_2(\delta = 0) = 0$ and Alice and Bob would not need to remove any bits from the sifted key. Since this is not the case, they must communicate at least an $h_2(\delta)$ number of bits.

For privacy amplification, considering a direct reconciliation setting, the fraction of bits which must be removed are the ones which Eve has information on. Therefore, they can be calculated by the mutual information between Alice and Eve $I(A:E)$. Note that, for inverse reconciliation, one simply replaces this by $I(B:E)$.

Finally, the secret key fraction, in the *asymptotic-key* scenario, is given by

$$r = I(A:B) - I(A:E). \tag{2.29}$$

From Equation 2.29 it is clear that the more information Eve has on the key, the smaller the secret key fraction is. To calculate r, security proofs must then evaluate $I(A:E)$, which in turn depends on the protocol and the type of attack Eve performs.

2.5 Classification of Attacks

The security of a protocol has to be proven against all attacks Eve can perform to get information about the key. These can be divided into three classes, which are, in increasing order of power, individual attacks, collective attacks and coherent attacks [31]. Eve's extraction of information, regardless of what attack class she chooses, can be generalized as such: not knowing Alice's state, she is left with attaching an

ancilla system $|E\rangle_E\langle E|$ to Alice's system ρ_A; she then performs a unitary transformation to the composite system, leaving her ancilla system in the state

$$\rho_E = \mathsf{Tr}_A\big(U^\dagger(\rho_A \otimes |E\rangle_E\langle E|)U\big); \tag{2.30}$$

to extract a result, she must measure ρ_E, for which she chooses an appropriate POVM $\mathcal{M} = \{M_j\}$, where result j happens with probability $\mathsf{Tr}(M_j\rho_E)$.

In individual attacks, Eve handles each state sent by Alice ρ_A^i individually. Considering n the total number of states, such that $i = 1, ..., n$, Eve attaches $|E\rangle_E\langle E|$ to ρ_A^i, retrieving

$$\rho_E^i = \mathsf{Tr}_A\big(U^\dagger(\rho_A^i \otimes |E\rangle_E\langle E|)U\big). \tag{2.31}$$

Afterwards, she measures each ρ_E^i individually, obtaining the probability distribution $\mathcal{P}_{\mathcal{M}^1}^{\rho_E^1}...\mathcal{P}_{\mathcal{M}^1}^{\rho_E^n}$. Conversely, in collective attacks, Eve measures the composite state $\rho_E^1 \otimes \rho_E^2 \otimes ... \otimes \rho_E^n$, creating a final probability distribution $\mathcal{P}_{\mathcal{M}^n}^{\rho_E^1 \otimes ... \otimes \rho_E^n}$.

Coherent attacks are the most powerful ones, where one assumes Eve has unlimited resources. Here, she attaches $|E\rangle_E\langle E|$ to the composite system formed by all n states, $\rho_A^1 \otimes \rho_A^2 \otimes ... \otimes \rho_A^n$, and performs a global unitary transformation U_G, retrieving

$$\rho_E = \mathsf{Tr}_A\big(U_G^\dagger(\rho_A^1 \otimes \rho_A^2 \otimes ... \otimes \rho_A^n) \otimes |E\rangle_E\langle E|U_G\big), \tag{2.32}$$

to which she applies a global measurement, with probability distribution $\mathcal{P}_{\mathcal{M}^n}^{\rho_E}$.

As aforementioned, the mutual information $I(A:E)$ shared between Alice and Eve depends on the type of attack Eve performs. For individual attacks we have that

$$I(A:E) = \max_{\text{Eve}} I(A:E), \tag{2.33}$$

where the maximization considers all the individual attacks strategies Eve can perform and chooses the strongest one, and $I(A:E)$ is the classical mutual information.

In turn, for collective attacks, we have

$$I(A:E) = \max_{\text{Eve}} I(A:E)_{\rho_{AE}}, \tag{2.34}$$

where the maximization follows the same logic as before, but now $I(A:E)_{\rho_{AE}}$ represents the quantum mutual information associated with the classical-quantum state $\rho_{AE} = \sum_k p_K(k)|k\rangle_A\langle k| \otimes \rho_E^k$.

Finally, to completely validate a given protocol, one must prove its security against the most powerful eavesdropping tactic, coherent attacks. However, these involve a very high dimension global Hilbert Space, including many possible variations, and consequently cannot be efficiently parameterized. Nonetheless, techniques for security proofs against coherent attacks have been developed [31], but are outside the scope of this thesis.

As an example, we show how the secret key fraction r is calculated for an individual attack. The most

general individual attack [31] Eve can perform in a prepare-and-measure protocol consists of

$$\begin{aligned} U\left|0\right\rangle\left|E\right\rangle &= \sqrt{\delta}\left|0\right\rangle\left|E_{00}\right\rangle + \sqrt{1-\delta}\left|1\right\rangle\left|E_{01}\right\rangle \\ U\left|1\right\rangle\left|E\right\rangle &= \sqrt{1-\delta}\left|0\right\rangle\left|E_{10}\right\rangle + \sqrt{\delta}\left|1\right\rangle\left|E_{11}\right\rangle, \end{aligned} \tag{2.35}$$

where δ is the QBER from Equation 2.27, $|E\rangle$ is the initial state of her ancilla system, and $|E\rangle_{ij}$ the states after the unitary transformation.

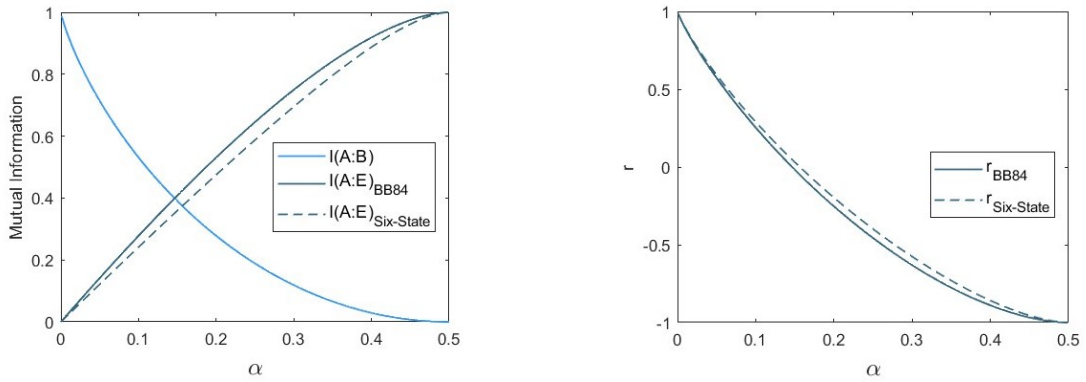
As proven in [47], the mutual information shared between Alice and Eve, $I(A:E)$, when Eve performs this attack in a BB84 implementation can be written as

$$I(A:E) = \frac{1}{2}(1+f(\delta))\log_2(1+f(\delta)) + \frac{1}{2}(1-f(\delta))\log_2(1-f(\delta)), \quad \text{with} \quad f(\delta) = 2\sqrt{\delta(1-\delta)}. \tag{2.36}$$

However, the same attack does not result in the same $I(A:E)$ when considering a different QKD protocol. For example, considering the Six-State protocol, another QKD prepare-and-measure protocol, the mutual information $I(A:E)$ [48] can be written as

$$I(A:E) = 1 + (1-\delta)\big(g(\delta)\log_2(g(\delta)) + (1-g(\delta))\log_2(1-g(\delta))\big), \quad \text{with} \quad g(\delta) = \frac{1}{2}\big(1 + \frac{1}{1-\delta}\sqrt{\delta(2-3\delta)}\big). \tag{2.37}$$

Using $I(A:B)$ from Equation 2.28, which is the same for both protocols, the secret key fraction r from Equation 2.29 can be calculated for each protocol by using the respective $I(A:E)$. In Figure 2.2 **(a)**, we can see the evolution of $I(A:E)$ with the QBER: as the number of errors in the key increases, the more information Eve has on the key and the less information is shared between Alice and Bob. Figure 2.2 **(b)** shows the secret key fraction for both protocols, where it is evident that, for α such that $I(A:B) > I(A:E)$, no key can be extracted, since $r < 0$.



(a) **(b)**

Figure 2.2: Mutual Information $I(A:B)$ and $I(A:E)$ for the BB84 and the Six-State Protocol in **(a)** and their corresponding difference r in **(b)**

In conclusion, the parametrization of Eve's attacks in a security proof is necessary to distil a secret key fraction r, which in turn is crucial for Alice and Bob to know how many key bits they can securely extract from the sifted key. If this analysis is based on naive assumptions, which do not hold in real implementations of the protocol, then the secret key rate might be overestimated, and security may be

breached. In the next section we will summarize different ways in which this has been done.

2.6 Optical Side-Channel Attacks in Discrete Variable QKD

As we have seen, although theoretically secure, QKD is susceptible to eavesdropping. Before being implemented and studied for QKD systems, physical security attacks, also known as side-channel attacks, were introduced in classical cryptography. In both settings, classical and quantum, they exploit implementation-specific characteristics of the cryptographic device in order to extract information.

In a QKD framework, as mentioned in Section 2.2, it is assumed that Eve has total control of the Quantum Channel. Consequently, most hacking strategies that have been proposed so far have explored vulnerabilities via the quantum channel, this is, Eve tries to gain information by manipulating the optical band of the cryptographic setting. In this sense, these attacks can be categorized as Optical Side-Channels.

Here we will focus on DV QKD systems, where attacks they can be categorized into four classes: Photon Number Splitting, Faked States, Trojan Horse and Backflash [49].

2.6.1 Photon Number Splitting Attacks

As seen in subsection 1.3.3, QKD implementations typically resort to highly attenuated weak coherent state sources as a replacement to single photon ones. In this case, the number of photons in the states prepared by Alice obey the probability distribution in Equation 1.38. Let us assume a quantum channel with transmittance $\tau = e^{-\eta}$, where η is the previously introduced transmissivity of the channel. If Alice sends a weak coherent state with average photon number μ through this channel, when Bob receives it, this number will have decreased to $\tau\mu$. Therefore, the probability of him receiving a non-vacuum pulse is given by

$$P(n > 0) = 1 - e^{-\tau\mu}. \tag{2.38}$$

For this attack, Eve replaces the lossy quantum channel with an ideal one, and performs a photon number resolving (PNR) quantum non-demolition measurement to the pulses, which measures their number of photons without disturbing the state where information was encoded, e.g. polarization. The zero photon states are straightforwardly resent to Bob, since they do not hold any information. However, upon receiving a multi-photon pulse, Eve splits off one photon, stores it in a quantum memory, and forwards the remaining to Bob, as schematized in Figure 2.3. When Alice reveals her basis choice over the classical channel, Eve can perform the correct measurements on the stored photons, and gain information about the bits they encoded. Nonetheless, if she resends every single photon pulse to Bob, given that she can't gain information on them, then the probability in Equation 2.38 would increase, because the quantum channel is now ideal ($\tau = 1$). Therefore, to assure she remains undetected, Eve must block some single photon pulses in order to mimic the expected probability distribution for non-vacuum pulses.

To counter Photon Number Splitting Attacks, the Decoy State Protocol was firstly suggested in [50]. Here, Alice employs two weak coherent state sources, with different mean photon numbers μ_1, μ_2, and

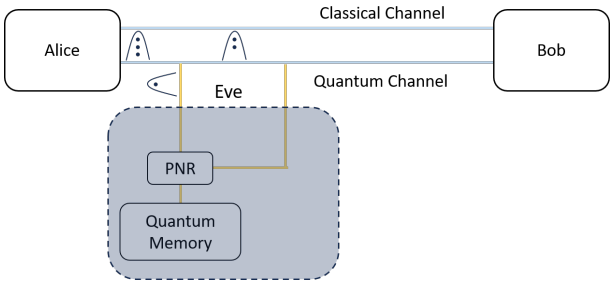


Figure 2.3: Schematics of a Photon Number Splitting Attack

alternates between the two. In the quantum channel, Eve cannot distinguish which source the states come from, thus she treats them equally and implements exactly the same strategy as before. However, now, when she blocks single photon states in order to maintain the probability Bob expects for non-vacuum states, she is correcting for states with a Poisson distribution $\mu_1 + \mu 2$. This means that, after Alice announces which states were created by which source, Bob will detect a higher loss that expected for each state population.

Nonetheless, vulnerabilities in the implementation of Decoy State Protocols have also been found, mainly regarding the assumption of indistinguishability between decoy and signal states. If Eve can distinguish them, she can adjust her strategy to keep the statistic distributions as expected by Bob, and perform a photon number splitting attack whilst undetected. In [51], the authors tested a source which alternated between two intensities via laser diode gain switching, by modulating its pump current. They found, with this setup, that the probability distributions of the signal and decoy state did not overlap, specifically, the higher intensity pulses had a higher probability of being emitted earlier. This is expected since the laser diode reaches, on average, population inversion earlier for higher pump currents. Also in [51], a setup with an external intensity modulator connected to the laser was tested and found to create indistinguishable decoy and signal states.

2.6.2 Faked State Attacks

In this type of attack, instead of trying to gain knowledge whilst remaining undetected, Eve sends designed states through the quantum channel to try and influence Bob's results. According to [12], where these types of attacks were firstly introduced, a faked state attack is an intercept-and-resend attack, where Eve generates light pulses to get detected by legitimate parties in a controlled manner, without being noticed. As schematized in Figure 2.4, Eve cuts into the fiber Alice uses to send photons to Bob and connects it to a copy of Bob's measurement setup; she then prepares faked states, which are setup and even target device dependent, and sends them to Bob.

In [12], two of the four devised attacks exploit the passive-basis choice on Bob's setup, where rather than using a source of true randomness to decide the basis for each incoming photon, depolarized photons impinging on a polarizing beam splitter randomly choose their detection basis. The third attack devised forced detections via parasitic reflections in Bob's setup, and the fourth one explored detector efficiency mismatch. The latter focused on SPADs working in gated mode, a common approach to reduce

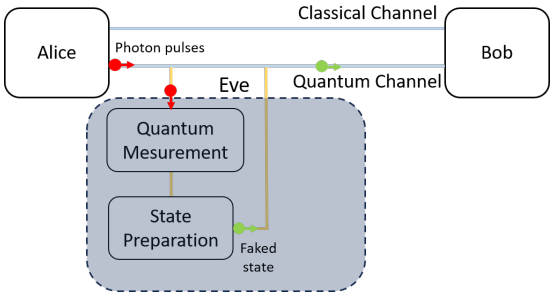


Figure 2.4: Schematics of a Faked States Attack

dark counts, that results in the detectors having a time-dependent efficiency. This vulnerability was further explored in [52]. In a setup with two SPADs, the detection windows will typically overlap, but not coincide completely. Thus Eve can encode states in a well-defined manner in relation to her measurement outcome and time-shift them to the sides of Bob's detector sensitivity curve, taking advantage of the higher sensitivity of one detector regarding the other and vice-versa.

2.6.3 Trojan Horse Attacks

Firstly introduced with the Large Pulse Attack [13], Trojan Horse Attacks also involve sending states into one or more of the legitimate parties devices, but now Eve does not aim at manipulating detections, wanting in turn to gain information about the key.

In the Large Pulse Attack, schematized in Figure 2.5, Eve takes advantage of the reflection coefficient of optical components in Alice's setup and the finite operational time of her internal modulators. For example, in a setup which uses phase modulators to encode the qubit's basis, Eve can send large light pulses to Alice's setup that pass through the modulator while it is still operational and gain the encoding of the signal states sent to Bob. The reflections of these light pulses can then be analysed by Eve to gain information about the key. This might not allow her to know the bit value, and only the encoding basis, nonetheless in this case she can perform an intercept-and-resend attack where she always chooses the same basis as Alice, thus not introducing errors in the key. If, conversely, Eve chooses to target Bob's setup, then she can gain information about his measurement basis and perform a PNS in a BB84 implementation without a quantum memory.

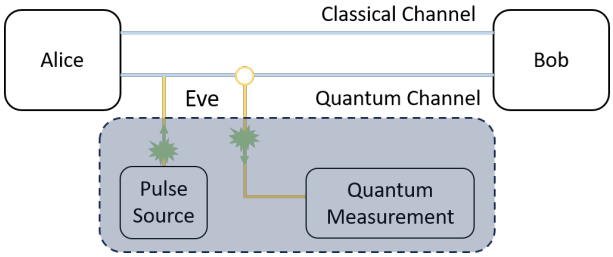


Figure 2.5: Schematics of a Trojan Horse Attack

2.6.4 Backflash Attacks

As mentioned before, avalanche photodiodes have become a preferred choice in practical implementations of QKD. In 1955, Newman reported for the first time a significant emission of light accompanying the avalanche of carriers during photon absorption in silicon p-n junctions [53]. From then on, the emission of light from APD's has been a well known phenomena, being firstly introduced as a vulnerability that could be explored by side-channel attacks in 2001 [54].

The quantum state of the backflash photon is not correlated to the one of the incident photon, however after emission, the photon can pass through security sensitive components of Bob's implementation and take some information back to the quantum channel, where it can be picked up by Eve. For example, considering a polarization encoding with a passive basis-choice scheme, backflash photons emitted by the detector which measures, for example, horizontally polarized photons, will return to the quantum channel with a horizontal polarization. In this way, Eve can obtain information about which detector the photon came from.

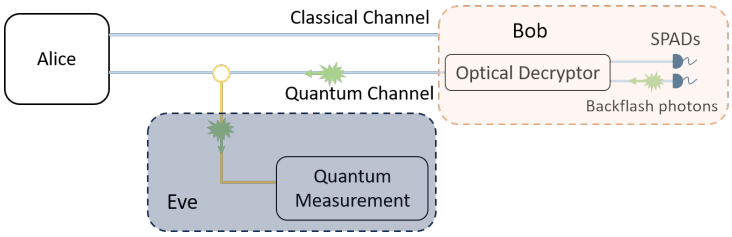


Figure 2.6: Schematics of a Backflash Attack

2.7 Device-Independent QKD

All the side-channels shown in Subsection 2.6 pose a security threat due to the same base problematic: the operation of some device does not follow the assumptions made by the protocol's security proof. These threats can be mitigated by patching the side-channel, however this does not assure other side-channels will not be found by the eavesdropper. Fortunately, Device-Independent (DI) QKD offers a way of mitigating side-channels which exploit device imperfections.

DI QKD protocols rely on the historical Ekert 91 protocol [23], where the Bell State in Equation 1.19 is shared between Alice and Bob. To measure her photon, Alice selects her measurement basis from three options: $A_1 = \mathbb{Z}$, $A_2 = \mathbb{X}$ and $A_3 = \frac{1}{\sqrt{2}}(\mathbb{X} + \mathbb{Z})$, while Bob chooses from $B_1 = \mathbb{Z}$, $B_2 = \frac{1}{\sqrt{2}}(\mathbb{Z} - \mathbb{X})$ and $B_3 = \frac{1}{\sqrt{2}}(\mathbb{Z} + \mathbb{X})$. When they choose the same basis, one of them flips the resulting bit value, and they both use it as part of their key. If they choose different basis, they use their results to calculate the CHSH quantity,

$$S := \langle A_1 B_3 \rangle + \langle A_1 B_2 \rangle + \langle A_2 B_3 \rangle - \langle A_2 B_2 \rangle, \tag{2.39}$$

where, if A_1, A_2, B_2, B_3 are classical variables which can take one of two values $+1$ or -1, then they obtain the CHSH inequality $S \leq 2$ [31]. Conversely, if they are quantum observables, the inequality is violated, and S reaches the maximum violation value of $S = 2\sqrt{2}$ when Alice and Bob share a maximally

entangled state, as the one in Equation 1.19. Therefore, if after the quantum transmission Alice and Bob retrieve $S \leq 2$, they know they shared separable states, which means eavesdropping has occurred, given that a maximally entangled bipartite state cannot be entangled with a third party.

To encompass the DI framework, the idea behind the Ekert 91 protocol is combined with foregoing trust on devices. To circumvent this, and prove security, Alice and Bob self-check their devices during the protocol via their input-output statistics. For example, in the spot-checking CHSH DI QKD protocol, Alice and Bob alternate between key and test rounds. In the key rounds, they measure their photon in the same basis, whilst in the test rounds, they play the CHSH game, schematized in Figure 2.7.

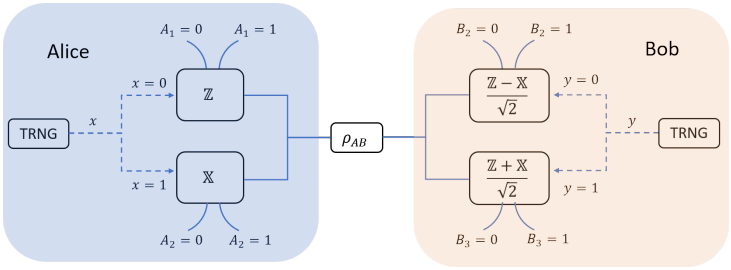


Figure 2.7: Schematics of a CHSH Test Setup

Via the CHSH game, parties calculate the CHSH quantity, in order to keep the devices honest. In an ideal implementation $S = 2\sqrt{2}$, but in reality, some tolerance δ is given to account for physical imperfections, and thus Alice and Bob abort the protocol if $S < 2\sqrt{2} - \delta$.

Given that they rule out many quantum hacking threats, one would expect DI QKD protocols to be the primary choice. However, in practice, their implementation is very challenging [55], since it relies on the sharing of high-quality entanglement and close to perfect quantum measurements. Additionally, device independence does not assure protection against all side-channels. Security in the DI QKD framework relies on the following assumptions [55]:

1. Quantum Mechanics is correct
2. Parties are connected through an authenticated classical channel
3. Parties are able to perform post-processing operations faithfully
4. Parties are able to generate truly local randomness
5. Parties perform their measurements in secure locations, where no unwanted information leakage occurs

This last assumption makes DI QKD vulnerable under the side channel explored in this thesis, as we will see below

2.8 Electronic Side-Channel Attacks in QKD

From Sections 2.2 and 2.7, we conclude that the assumption that Alice and Bob's laboratories are secure locations where no unwanted information is leaked applies to all QKD protocols. Typically,

guaranteeing this can be done by either physically protecting the facilities, or, for plug-and-play devices, by assuring boxes are unopenable, e.g. self-destruct upon opening.

Nonetheless, electronic side-channel attacks might offer a way for Eve to gain information, even when faced with these protective measures. These attacks exploit vulnerabilities related to the electronic components of a cryptographic setting, and were introduced for classical devices in the seminal work [56]. These devices are typically implemented using semiconductor transistors as logic gates, where electrons flow across the silicon substrate when voltage is applied or removed from the gate. This means that their power consumption and emanated electromagnetic radiation is data-dependent, namely, depends on the Hamming weight of the handled bits. Exploring these dependencies has, since then, evolved into the fields of power and electromagnetic side-channel analysis, respectively.

Returning to the quantum framework, electronic devices, such as Field Programmable Gate Arrays (FPGAs), are ubiquitous to high-rate QKD setups [57] and are also based on transistor logic, thus opening the possibility for electronic side-channels. Until now, only two works devising these types of hacking strategies against QKD setups have been put forth, both exploring electromagnetic vulnerabilities [16, 17].

Summarily, in [16] researchers used a magnetic near-field probe, an RF amplifier and an oscilloscope to measure the EM near-field emissions from the control electronics of the QKD transmitter. The module consisted on a BB84 polarization-encoding decoy-capable setup, where the control electronics lied in a (10×10) cm^2 Printed Circuit Board (PCB). With the probe fixed at 10 mm from the unshielded PCB, a machine-learning based attack was implemented on different locations of the board and at the optimal location, a prediction accuracy of 99% was reached. However, for distances between the probe and the PCB above 8 cm, the attack was no longer considered successful, motivating a far-field analysis. Using a log-periodic dipole antenna and the same neural network, the far-field was explored but proved unsuccessful for extracting a key. Nonetheless, in the article it was proved that the far-field emissions had non-zero information.

In [17], researchers explored the Radio Frequency (RF) radiations emitted by the accelerated displacement of electrons in a SPAD. Given their specific location in the laboratory, the radiation emitted by each SPAD creates a unique spectrum, due to the multiple reflections induced by the detectors' physical surroundings. They created a deep learning signal classification neural network to distinguish between two different SPADs via their RF fingerprint. With the detectors placed 20 cm apart, and an antenna 2 m away from the SPAD setup, they found an accuracy above 99%. Afterwards, the same procedure was applied to a BB84 implementation, with four SPADs, a pair per basis choice, and a detector in each pair for each bit value. They found also that, with this procedure, they were able to extract more than 99% of the raw key, from which Eve could then extract the sifted key by listening to Alice and Bob's classical communication.

Equipped with the state-of-the-art on electronic side-channels for QKD systems, we can now better understand the objective of this thesis, which is to perform a power side channel on the QKD transmitter at the University of Padua. As previously mentioned, these attacks are implementation dependent, so the next chapter will briefly cover the main aspects of the QKD implementation at the University of Padua.

Chapter 3

Working Principle of the QKD transmitter

The QKD setup at Padua implements the three-state one-decoy BB84 protocol with polarization encoding. In the transmitter, an Intensity Modulator (IM) [58] is used to create decoy and signal states, followed by a polarization modulator, the iPOGNAC [59], which encodes the polarization of each photon. In Figure 3.1, a schematics of the setup implemented in [60] is shown. Here, the detection at the receiver was achieved with a time-multiplexing scheme, where the measurements on the two BB84 bases were done using only one InGaAs/InP SPAD. Regarding the transmission, in this setup, the classical and quantum communication were established within the same fiber via wavelength multiplexing. The quantum states were emitted at 1550 nm, by a gain-switched distributed feedback laser, with a $f_{\text{rep}} = 50$ MHz repetition rate, whilst the classical communication was held at 1490 nm. This allowed the key exchange to happen using an already stationed telecom dark fiber.

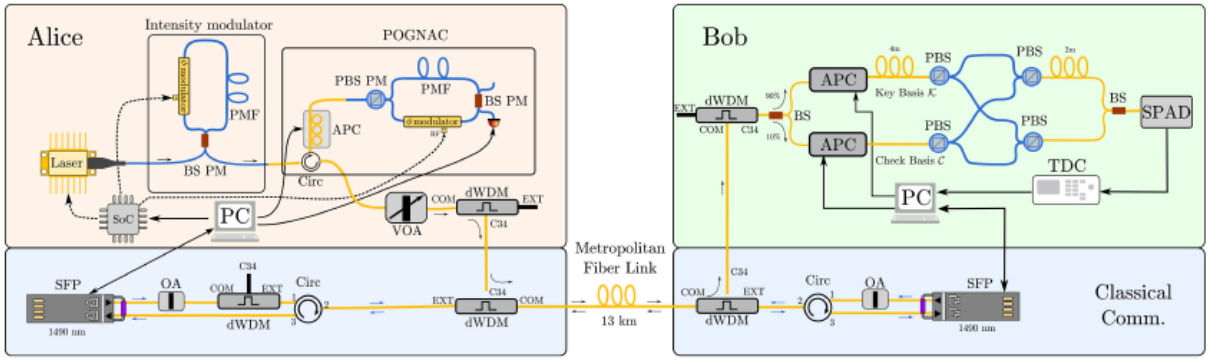


Figure 3.1: Schematic representation of a QKD setup employed by researchers at University of Padua. Taken from [60]

The side-channel attack performed in this thesis was done on the electronic drivers at the transmitter side, therefore, this chapter will primarily focus on explaining the main components of the QKD transmitter. However, we start by outlining the three-state one-decoy BB84 protocol, in order to understand its experimental demands.

3.1 Three-State One-Decoy BB84

The three-state BB84 protocol relies only on 3 qubit states, two which contribute to key generation, and a third which is solely used for parameter estimation. Its security was proven in [61] for a single-photon source and for a coherent state source with a decoy state method, which is the case at the University of Padua. With a polarization encoding, the protocol has the following outline [26]:

1. For each bit, Alice chooses between two basis, $\mathcal{K}$ and $\mathcal{C}$, corresponding to the key and the check basis, with probabilities $p_{\mathcal{K}}^{A}$ and $p_{\mathcal{C}}^{A}$ = 1 - $p_{\mathcal{K}}^{A}$, respectively. Note that $\mathcal{K}$ and $\mathcal{C}$ can be any two MUB, where here we use $\mathcal{K} = \{|L\rangle, |R\rangle\}$ and $\mathcal{C} = \{|D\rangle, |A\rangle\}$ since these are the polarization states prepared at the University of Padua. If $\mathcal{K}$ is chosen, Alice transmits a state randomly chosen between $|L\rangle = \frac{|H\rangle + i|V\rangle}{\sqrt{2}}$ and $|R\rangle = \frac{|H\rangle - i|V\rangle}{\sqrt{2}}$, whilst if $\mathcal{C}$ is selected, she sends $|D\rangle = \frac{|H\rangle + |V\rangle}{\sqrt{2}}$.
2. Bob measures each bit, choosing randomly between the two measurement bases $\mathcal{K}$ and $\mathcal{C}$, with respective probabilities $p_{\mathcal{K}}^{B}$ and $p_{\mathcal{C}}^{B}$ = 1 - $p_{\mathcal{K}}^{B}$.
3. Alice and Bob announce over a classical channel the basis used to measure each bit
4. They discard the bits corresponding to different basis. The bits belonging to the key basis $\mathcal{K}$ are used for key generation, whilst the ones belonging to $\mathcal{C}$ are used for parameter estimation.

As previously mentioned, the QKD setup at Padua uses a coherent-state source, and to protect against PNS attacks, they implement a one-decoy protocol, where for each state sent through the quantum channel, Alice chooses between two intensities μ_1 and μ_2 ($\mu_1 > \mu_2 > 0$) with probabilities p_{μ_1} and $p_{\mu_2} = 1 - p_{\mu_1}$. To implement this, they use an Sagnac-based intensity modulator, described in the next section.

3.2 Intensity Modulator

To modulate each pulse's intensity, between μ_1 and μ_2, a Lithium Niobate ($LiNbO_3$) phase modulator placed inside a Sagnac interferometer is used as a two-level IM [58], as depicted in Figure 3.2

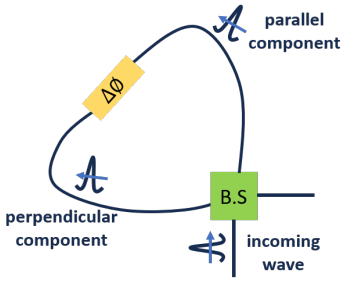


Figure 3.2: Schematic representation of a Sagnac-based IM

The pulses created by the gain-switched laser are directed to the IM, where they are divided into two fiber paths, denoted as parallel and anti-parallel, going anticlockwise and clockwise respectively. Then, a phase shift is applied to the parallel light pulse, via the use of an electro-optical phase modulator. In this

device, the light passes through an electro-optical crystal, where the voltage at its terminals determines its refractive index, and thus, the phase delay of the light pulse. At the output, both light pulses are recombined by the beam splitter into a single pulse, whose intensity obeys

$$I \propto R^2 + T^2 + 2RT\cos(\Delta\phi). \tag{3.1}$$

Noticeably, in the Sagnac interferometer, both parallel and anti-parallel light pulses travel through the same length of fiber, meaning that, any perturbations in the fiber or DC changes in the phase modulator affect both pulses equally, which increases the stability of the interferometer [58].

The coupling ratio of the beam splitter is fixed, and, for example, on the exchange held in [60] it was fixed at 70:30. For fixed coupling ratios, the maximum optical extinction ratio of the IM is achieved when the higher intensity pulses correspond to applying $\Delta\phi = 0$, and the lower intensity ones to $\Delta\phi = \pi$. The voltage required for inducing a π phase change to light pulses passing through a phase-modulator is called the half-wave voltage $\tilde{V}_\pi$. Therefore, to choose between two intensity levels μ_1 and μ_2 which maximize $\frac{\mu_1}{\mu_2}$, one simply chooses between applying V_π or 0 V at the terminals of the phase modulator. If, instead, one wants an arbitrary value for $\frac{\mu_1}{\mu_2}$, they can reach this value by tuning the voltage V applied to the phase-modulator, such that $V < \tilde{V}_\pi$. For example, in [60], the voltage was tuned such that $\frac{\mu_1}{\mu_2} \approx 3.5$.

3.3 Polarization Modulator (POGNAC)

After exiting the IM, the pulses have been encoded with decoy/signal information, but are yet to be encoded with key information. To do so, they are put through the POGNAC, a polarization modulator based on a $LiNbO_3$ phase modulator inside a Sagnac interferometer. In this section we will briefly go through the device's theory of operation, detailed in [59]. Figure 3.3, taken from [59], represents a schematics of the working principle of the POGNAC, where, importantly, single mode fibers are represented in yellow, and polarization mantaining (PM) fibers in blue.

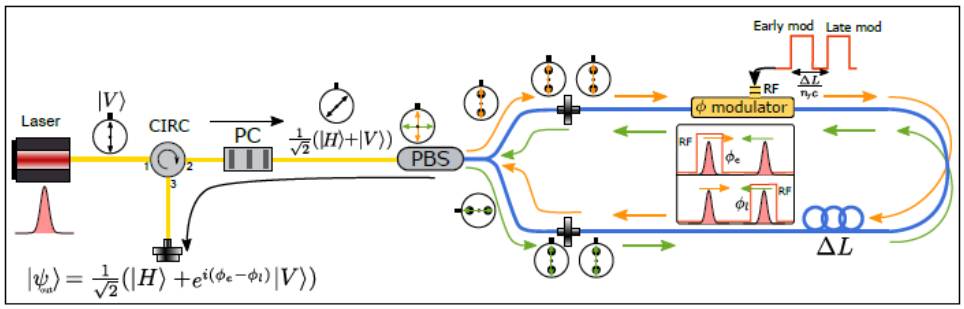


Figure 3.3: Schematic representation of the POGNAC. Taken from [59]

The POGNAC receives linearly polarized light pulses, which enter an Optical Circulator (CIRC) through port 1, and exit through port 2, where they directed to a Polarization Controller (PC). The PC changes their polarization to the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|H\rangle + e^{i\phi_0}|V\rangle)$, where ϕ_0 is an arbitrary relative phase, inconsequential to the polarization state of the light output. Afterwards, the pulses go through a Polarizing Beam Splitter

(PBS) which splits the two polarization states, $|H\rangle$ and $|V\rangle$, into two different spatial modes. The PBS marks the beginning of the Sagnac loop, where the two orthogonally polarized light pulses travel through the slow axis of a PM fiber in opposite directions. The $|V\rangle$ polarized light travels through the clockwise direction, going first through the phase modulator, where it is applied a phase ϕ_e, and then through a PM fiber delay line, of length ΔL. The $|H\rangle$ polarized light does the opposite path, in the counter clockwise direction, reaching the phase modulator after passing through the delay line, where it is given a phase ϕ_ℓ. In this way, both light pulses exit the interferometer at the same instance, where their polarization state is recombined to the final state

$$\left|\psi_{\text{out}}^{\phi_e,\phi_\ell}\right\rangle = \frac{1}{\sqrt{2}}[|H\rangle + e^{i(\phi_e-\phi_l+\phi_0)}\,|V\rangle]. \tag{3.2}$$

The $|H\rangle$ polarized light arrives at the phase modulator with a time delay of $\frac{\Delta L}{n_f c}$, compared to the $|V\rangle$ polarized one. Consequently, one can tune the values ϕ_e and ϕ_l by carefully timing the voltage pulses at the terminals of the phase modulator.

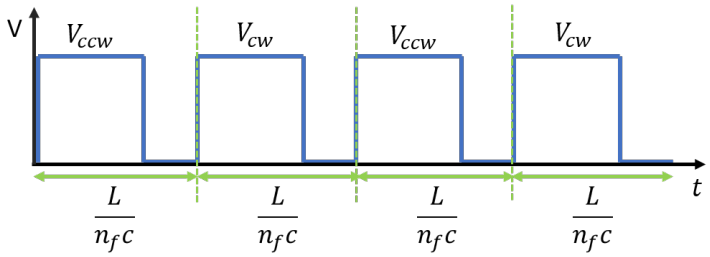


Figure 3.4: Schematic representation of the voltage pulses at the terminals of the phase modulator in the POGNAC

For simplicity, let us assume $\phi_0 = 0$. If equal voltage, i.e V_{CCW} = $V_{CW} = 0$ is applied to both counter propagating pulses, then $\phi_e = \phi_\ell$ and the state 3.2 becomes

$$\left|\psi_{\text{out}}^{0,0}\right\rangle = \frac{1}{2}[|H\rangle + |V\rangle] = |D\rangle\,. \tag{3.3}$$

Conversely, one can create the output states $|L\rangle$ and $|R\rangle$ by accordingly applying $V_{\frac{\pi}{2}}$ to the phase modulator during the passage one of the light pulses,

$$\begin{aligned} V_{CCW} = 0, \quad V_{CW} = V_{\frac{\pi}{2}} \quad &\rightarrow \quad \left|\psi_{\text{out}}^{\frac{\pi}{2},0}\right\rangle = \frac{1}{2}[|H\rangle + i\,|V\rangle] = |L\rangle\,, \\ V_{CCW} = V_{\frac{\pi}{2}}, \quad V_{CW} = 0 \quad &\rightarrow \quad \left|\psi_{\text{out}}^{0,\frac{\pi}{2}}\right\rangle = \frac{1}{2}[|H\rangle - i\,|V\rangle] = |R\rangle\,, \end{aligned} \tag{3.4}$$

where $V_{\frac{\pi}{2}}$ is the voltage required for the device to apply a $\frac{\pi}{2}$ shift. In this way, we can modulate the polarization of each light pulse between the three states required by the Three-State BB84 protocol. After exiting the PBS, the light pulses are guided by a single mode fiber to port 2 of the CIRC, and exit the POGNAC through port 3.

Tracing back to Figure 3.1, after leaving the POGNAC, the light pulses are sent to Bob, whose receiver module irrelevant for our side-channel analysis. Nonetheless, the most relevant component of

the transmitter for this analysis, the System-on-a-Chip (SoC), has yet to be addressed.

3.4 System-on-a-Chip (SoC)

The optical encoder in the transmitter requires three separate electrical signals: one to gate the laser every $\frac{1}{f_{\text{rep}}}$ seconds, and two for the phase modulators, one in the IM and the other in the POGNAC. To create these electrical pulses with the high temporal resolution required by fast repetition rates f_{rep}, a System-on-a-Chip (SoC) is used, comprising a Field Programmable Gate Array (FPGA) and a dual core CPU unit.

The hardware and software architecture used could be implemented as a QKD receiver as well as a quantum random number generator control unit, as described in [57]. For the purpose of this thesis, we will focus on its application as a QKD transmitter. Here, the architecture follows a top-down configuration, where data flows from the user/pc to the quantum system, passing through two different layers, as depicted in Figure 3.5. Firstly it goes through the CPU layer, which is responsible for communication with the outside world and data management operations. Then it reaches the FPGA, which is the lowest layer, where all deterministic and high resolution operations are carried. Finally, from the FPGA, it goes to the chip input-output pins, where the electric pulses are emitted.

Figure 3.5: Schematics of the top-down workflow configuration

To perform the data transfer from an external device, either generated in real time by a quantum random number generator or stored in a PC, Gigabit Ethernet communication is used. However no encryption is done to the raw data, which demands setting a private Local Area Network (LAN) between the SoC and the external device, physically disconnected from other networks. From the CPU, the data must then be transferred to the FPGA. To do so, the software distinguishes between two types of data: configuration parameters, such as the key length, which have a slow refresh rate, and the raw key data. For the first type, the exchange happens with direct communication via the Advanced eXtensible Interface (AXI) protocol. However the raw key data exchange entails a higher refresh rate, thus requiring the use of the onboard DDR-RAM memory, accessible from the CPU, and the Block RAM memory (BRAM), integrated in the FPGA but accessible from the CPU [57].

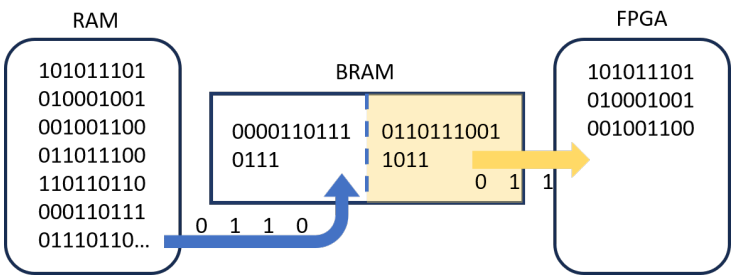


Figure 3.6: Schematics of the raw data transfer from the CPU to the FPGA

The BRAM has a memory length L_{BRAM}, in the order of MBits, which has been split into two halfs. The raw key data, received by the CPU via an external data source, is stored in the CPU's RAM. Then, via the AXI protocol, data strings of length $L_{\text{BRAM}}/2$ are transferred to one of the halves of the BRAM, whilst the FPGA reads data from the other half. When it reaches the end of one of the two halves, a signal is emitted by one of the FPGA's blocks, which signals the CPU to start writing new data in the corresponding half.

Chapter 4

Power Consumption of the System-on-a-Chip

Power consumption traces of cryptographic devices, taken during the cryptographic operation, can leak unwanted information about the encryption key. This is a standard and powerful technique in cryptoanalysis of classical devices implementing symmetric-key algorithms, such as the Data Encryption Standard (DES) [62] and the Advanced Encryption Standard (AES) [63]. However, it has yet to be applied to Quantum Key Distribution systems.

In this chapter, we characterize the power consumption of the SoC used for driving the electro-optical components of the QKD transmitter at the University of Padua. Vulnerabilities found in the course of this characterization are then used to implement a hacking attack, breaking the assumption that Alice performs her operations in secure locations. During the analysis and subsequent attack, the SoC is treated as a pseudo black-box, as it would be by a current eavesdropper with malicious intent, who would expectedly only have access to publicly available information about the device. Therefore, the only knowledge we assume on the working principle of the SoC is the one made public in [57], covered in Chapter 3.

4.1 Experimental Setup

Our target device is the Zynq-7020 SoC mounted on a ZedBoard by Avnet. The ZedBoard receives a 12 V input power supply and comes with an inbuilt 10 mΩ, 1 W in series resistor for power consumption measurements, placed at the high voltage side of the circuit, as schematized in Figure 4.1.

The terminals of the shunt resistor are connected to Header J21 of the ZedBoard [64], such that by measuring the difference between the voltages at the two pins, we can infer the ZedBoard's power consumption. To guarantee enough resolution for measurements regarding the average power consumption values, the 10 mΩ resistance was replaced by a an $1\ \Omega$, 1 W one. The 1Ω value was chosen in order to satisfy two conditions: the voltage at the resistance terminals is not high enough to require a higher voltage input to the Zedboard ($V > 12$ V), but it is high enough to allow resolution in the mV range.

The voltage difference at the terminals of the $1\ \Omega$ resistance translate the power consumption of the

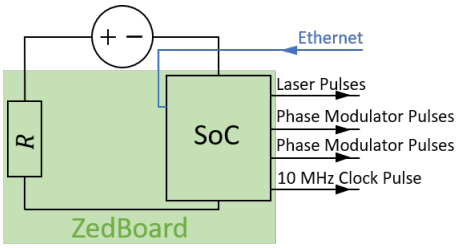
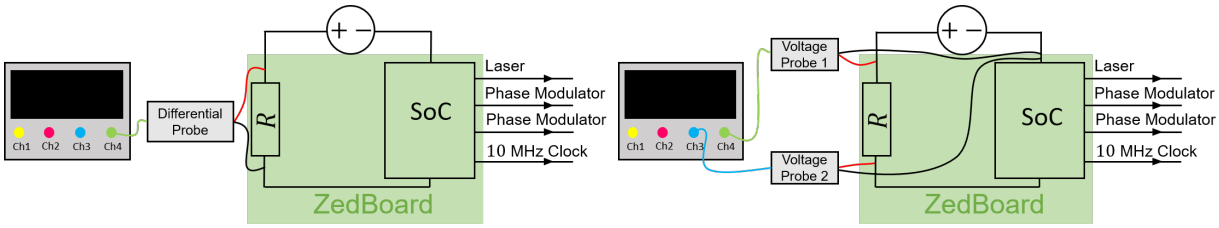


Figure 4.1: Schematics of the ZedBoard highlighting the inbuilt resistance in series with the SoC

SoC. Given that we are not interested in the actual power consumption values, only on how their variation can lead to information leakage, it is sufficient to characterize the voltage difference at the terminals of the resistance. To do so, we acquire power consumption traces, also referred to as power traces, which we define as the voltage difference data at the terminals of the resistance during a certain time interval.

The power traces were acquired using the SIGLENT SDS5104X Oscilloscope, which can reach a sampling frequency $f_{\text{samp}} = 5$ GSa/s, a record length of 250 MSa, and has bandwidth of 1 GHz. Two different probe setups were used to measure the voltage, according to the different requirements of the measurements. The first probe used was a KEYSIGHT N2791A Differential Probe, with a $10\times$ attenuation ratio, which directly computes the voltage difference between the two resistance terminals, as schematized in Figure 4.2 **(a)**. As specified in the probe's datasheet [65], it was connected to the oscilloscope at $1\ M\Omega$ impedance. This probe has a 25 MHz bandwidth, which is insufficient for an analysis of the frequency spectrum of the power traces, leaving this setup for the study of the average power consumption of the SoC. In this case, the sampling frequency of the oscilloscope was set to 500 MSa/s to avoid oversampling given the 25 MHz bandwidth of the differential probe. With this setup, two oscilloscope channels were used, one for the signal measured by the differential probe, and another for the laser pulse signal, which was used to trigger acquisitions.



(a) Setup with low bandwidth differential probe

(b) Differential setup with oscilloscope probes

Figure 4.2: Schematics of the two experimental setups

For the frequency spectrum analysis of the power traces, we used two SIGLENT SP2035A voltage probes with a bandwidth of 350 MHz and $10M\Omega$ impedance, as schematized in Figure 4.2 **(b)**. For each probe, the sensor was placed in one of the resistance terminals, whilst the ground was connected to the ground of the SoC, this way eliminating any parasitic voltage loops that may occur in other configurations. Using the oscilloscope, the power traces were retrieved by computing the difference between the two voltage traces, using a Math function of the oscilloscope. Importantly, given that the resistance is placed in the high voltage end of the circuit, both probes measure an average voltage higher than $5V$, not only

meaning that they must impedance matched at $1M\Omega$ with the oscilloscope, but that, if the signals are DC coupled to the oscilloscope, they require a scaling of at least a $2V/$div. This high scaling results in low resolution in the mV range, making it inappropriate for high precision measurements. To increase the resolution, both probes were AC coupled to the oscilloscope. Finally, in this case, the sampling frequency of the oscilloscope was set to the maximum 2.5 GSa/s allowed when using three channels. The three channels required were two for the oscilloscope probes and one for the laser pulses which were used as a trigger.

4.2 Oscilloscope

The oscilloscope is pivotal for power trace acquisition, and thus preparing and characterizing it is of utmost importance. The SIGLENT SDS5104X oscilloscope was chosen since it can reach high sampling rates, i.e $f_{\text{samp}} = 5$GSa/s, and high bandwidths, i.e. 5GHz. The first task upon setting up the instrument, was create the software to control it remotely.

4.2.1 Controlling the Oscilloscope

Acquiring power traces for different working conditions of the FPGA requires a control over the oscilloscope which can only be obtained by controlling it remotely. For this, we connected a computer to the Oscilloscope via USB, and used MATLAB to establish communication between both. The NI implementation of the Virtual Instrument Software Architecture (NI-VISA) provides a programming interface to control instruments over different communication protocols, USB being one of them. It is supported by MATLAB via the Instrument Control ToolBox, but the control commands are specific to each instrument. For the case of the SIGLENT SDS5104X Oscilloscope, the PG01-E02B Programming Manual offers the list of available commands together with some implementation examples. By consulting this manual, after installing the Instrument Control ToolBox, we used MATLAB to create the 'Oscilloscope_SD' class, where each class method implements different control sequences.

Throughout the data acquisition performed for the purpose of this thesis, we mostly relied on the following class methods:

- connect(obj): connects the computer to the oscilloscope.
- set_trigger(obj, type, source, slope, value): defines the trigger settings.
- measure(obj, n): retrieves the measurement result in the n^{th} slot
- retrieve_data(obj, channel, initial_point, final_point): retrieves the waveform data on a given channel, over a given interval, aligning the data according to the oscilloscopes settings, i.e the vertical and horizontal scale.

4.2.2 Effective Number of Bits (ENOB)

Before any characterization of the voltage traces is done, it is crucial to quantify the resolution we can achieve with our oscilloscope. The oscilloscope's resolution is labeled as 8 bit per data point. However, the analog to digital converter inside the oscilloscope is not ideal, and thus the effective number of bits (ENOB) of each data point will actually correspond to a bit number smaller than 8.

To calculate the ENOB of the SIGLENT SDS5104X Oscilloscope, we used the IEEE definition of ENOB [66]

$$\text{ENOB} = \frac{1}{2}\log_2(\text{SINAD}) - \frac{1}{2}\log_2\left(\frac{3}{2}\right) - \log_2\left(\frac{A}{V}\right), \tag{4.1}$$

where SINAD represents the Signal to Noise and Distortion Ratio, A is the peak to peak amplitude of the sine wave and V the full range of the oscilloscope. The KEYSIGHT Agilent 33220A Arbitrary Waveform Generator, with a maximum frequency generation of 20 MHz, was used to generate sine waves with a well defined amplitude A, and different frequencies. These signals were sent to the oscilloscope, acquired and then transferred to the computer. After storing the traces, the Fast Fourier Transform (FFT) of each waveform was calculated, in order to the compute the SINAD [67]

$$\text{SINAD} = \frac{P_S}{P_{NAD}}, \tag{4.2}$$

where P_S is the power in the FFT bin corresponding to the waveform's frequency and P_{NAD} is sum of powers in all other frequency bins excluding the 0 frequency bin, up to and including the bin at Nyquist frequency.

The input frequency of the sine waves must be chosen such that the sampled waveforms comprise an integer number of periods, this is, the frequency fits in exactly one bin of the FFT, and there are no windowing effects. Therefore, the optimal frequencies f_{opt} obey

$$f_{\text{opt}} = f_{\text{samp}}\frac{J}{M}, \tag{4.3}$$

where M is the number of samples in each waveform and J is the number of periods it comprises. In our analysis, the oscilloscope was set to its highest sampling frequency $f_{\text{samp}} = 5$ GSa/s, since expectedly, higher sampling frequencies correspond to lower ENOBs. Furthermore, we expect higher waveform frequencies to correspond to lower ENOBs, thus we choose to generate sine waves in the MHz range, close to the ceiling of the signal generator's frequency range. Finally, in [66], a minimum sample length L which produces a representative sample in every code bin is defined as

$$L = \pi 2^N, \tag{4.4}$$

where N is the resolution of the oscilloscope, which in our case $N = 8$. The record length chosen was of 10 MPts, which is well above the minimum required value.

The choice regarding the waveform's amplitude A is not specified by the IEEE. From Equation 4.1 we see that lower amplitudes correspond to higher ENOB values, and thus commonly, ENOB values

are given at 90% or 95% full scale. Due to the noisy nature of the voltage traces at the terminals of the resistance, we do not expect them to occupy 95% of the oscilloscope's full range, given that we must leave some room for occasional voltage drops or rises. Therefore, the ENOB was calculated at 90%, where the amplitude of the incoming sine waves was set at $A = 1.44$ V, thus $V = 1.6$ V, which corresponds to a 200 mV/div oscilloscope scale.

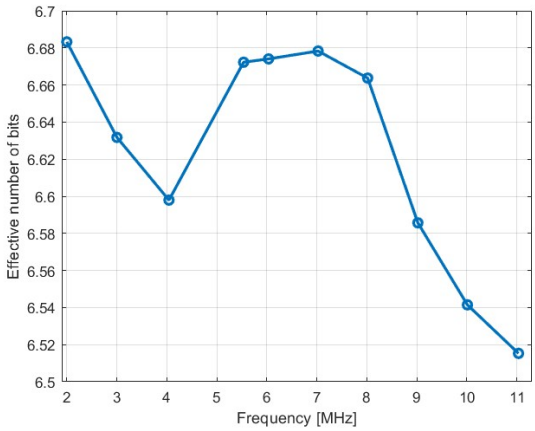


Figure 4.3: ENOBs of the SIGLENT SDS5104X Oscilloscope at 90% full width

Taking all these conditions into account, using the MATLAB oscilloscope class to program the oscilloscope's setting and the data acquisition by the computer, 10 sine waves were collected and their respective ENOBs calculated. The results are shown in Figure 4.3 and from them we extract an average ENOB of

$$\overline{\text{ENOB}} = 6.62 \pm 0.06, \tag{4.5}$$

meaning that, on average, we have 6.62 bits per data value, instead of the expected 8. We now must translate this result into voltage resolution. Considering the oscilloscope has a certain vertical division scale v_{div}, the vertical accuracy is given by

$$\frac{1}{2^{\overline{\text{ENOB}}}} \times \text{v}_{\text{div}} \times 8, \tag{4.6}$$

where 8 is the number of vertical divisions of the oscilloscope. For example, a vertical range $\text{v}_{\text{div}} = 20$ mV/div means a vertical accuracy of 1.6 mV, meaning that the oscilloscope cannot ideally resolve voltage differences smaller than 1.6 mV.

4.3 Field-Programmable Gate Array (FPGA)

As mentioned at the beginning of this chapter, we assume no knowledge of the FPGA's software other than the one made publicly available in [57]. Nonetheless, to control the system's settings, e.g. the qubit repetition frequency of the QKD transmitter, we relied on an application previously developed by the Quantum Future research group, which allows the user to connect to the FPGA and control the different settings. Moreover, it is also through this application that the user initiates or terminates symbol emission,

this is, sends an instruction to the FPGA which starts or ends the emission of electronic signals. Naturally, these signals depend, among others, on the following settings: clock frequency, decoy string, key length and symbol string.

For all the measurements taken in this thesis, the qubit repetition frequency of the QKD transmitter was set to 100 MHz, meaning that the laser pulse signal exiting the FPGA has a 10 ns period. Additionally, the decoy string was also left unchanged at all times, and set to a pre-defined sequence of all $1'$s. In this way, all symbols are programmed by the FPGA as non-decoy pulses, implying that the electronic pulse exiting the FPGA to feed the phase modulator in the IM is periodic (T = 10 ns).

Regarding the key, the symbol encoded by each qubit can be chosen between H, V or D. We note that these states differ from the ones in the explained in Section 3.1. Nonetheless, this difference is simply symbolic, and one associates the state $|H\rangle$ to $|L\rangle$ and the state $|V\rangle$ to $|R\rangle$. The application connecting the user to the FPGA allows for two options concerning the key choice. The first one is using fixed sequences, which consists of an infinite repetition of a fixed sequence of symbols. For example, an Only-H sequence corresponds to having the QKD transmitter sending H symbols indefinitely. Naturally, these sequences would not be used in a practical QKD setting, since to assure theoretical security, each symbol Alice sends must be chosen in a truly random manner, i.e. using a Quantum Random Number Generator (QRNG). Nevertheless, they can be a good starting point for our analysis. Lastly, when using fixed-sequences, a key length must be set by the user, however, this value is arbitrary since the key will be emitted indefinitely. When the emission reaches its end according to the key length setting, it continues again from the start, without ever stopping. To terminate the emission, the user must restart the board. The second option to define the key string is to encode it with a binary file, where an H symbol is encoded via the bit-string $'01'$, whilst a V symbol via $'10'$. The content on the file can be transferred to the FPGA by importing the file to the application. If the length of the key encoded in the file is smaller than the one set by the user in the application, then the remaining pulses are set to 'H' symbols. If they match, the FPGA ends the emission upon reaching the final symbol encoded in the file. This method allows us to closely resemble a QKD setting since we can generate files encoding pseudo-random symbol strings.

4.3.1 Randomize Method

The randomize method was implemented within the control interface of the FPGA in order to achieve statistical relevance for the results regarding fixed sequences. To better understand this, we focus on the case of Only-H/Only-V sequences, but the reasoning can be applied to any fixed sequence. As aforementioned, power traces acquired during the emission of an Only-H sequence correspond to a snippet of an operation that is set to run indefinitely. Therefore, the instant the power trace is acquired was preceded by H-symbol emission and will be succeeded by H-symbol emission. Now let us consider the case were there is a characteristic of power traces from Only-H emissions that differs from the one in power traces from Only-V emissions. If this difference could be solely associated to the percentage of H-symbols in the key, then this would mean that it would be visible in the random key emission scenario, and thus that there is a strong information leakage. However, in fixed-sequences, this association cannot

be done, since the characteristics of the power traces might be influenced by the FPGA emission before and after the acquisition. To reduce this influence, we developed the randomize method.

With the randomize method, the FPGA chooses randomly between five pre-defined sequences: Only-H, Only-HHV, Only-HV, Only-HVV, Only-V and Only-HVV. These sequences were chosen given their different percentage of H-symbols, respectively 100%, 66%, 50%, 33% and 0%. The chosen sequence is emitted during a time interval ΔT, and afterwards, another sequence is chosen and emitted during ΔT, and so forth. The emission stops after N_{seq} have been sent. The parameters ΔT and N_{seq} are defined by the user before the method is initialized. During emission, a register of the random sequence order is written in a txt. file, so that it can be consulted during data analysis. Regarding the acquisition, MATLAB is used to program the oscilloscope such that, at a fixed time interval δt before the end of each sequence emission, the oscilloscope acquires a power trace and transfers the data to the PC. The time span of the acquired power trace is defined by the oscilloscope settings, namely, if its time scale is set to t_{div}, then the power trace will correspond to a $10t_{\text{div}}$ time span, where 10 is the number of horizontal divisions. Figure 4.4 schematizes the timing of the emissions and respective power trace acquisitions during a run of the randomize method. Noticeably, using this methodology, one can associate each power trace to the corresponding sequence, and additionally, know the time each trace was acquired.

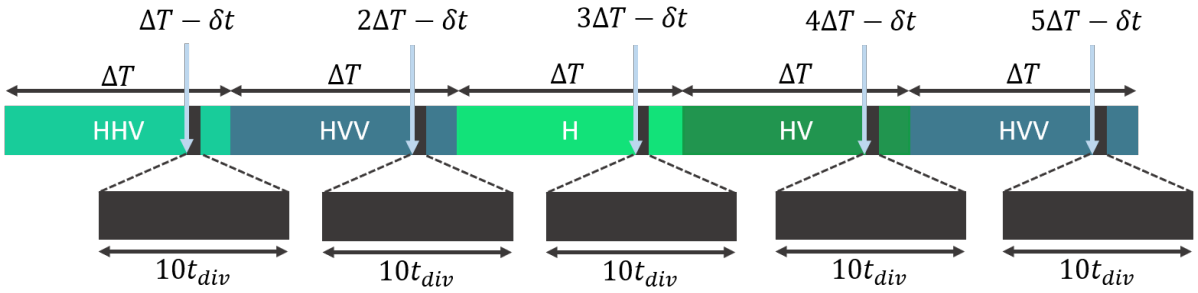


Figure 4.4: Schematics of sequence emission and power trace acquisition during the Randomize Method implemented for the FPGA

On the statistical relevance of this method, we note that the average time interval between the acquisition of power traces corresponding to the same sequence is $5 \cdot \Delta T$. If this interval is much larger than the symbol period, this is, $5 \cdot \Delta T \gg 10$ ns, and given that the sequence order is chosen randomly, different acquisitions of the same sequence can be considered independent.

4.4 Average Power Consumption

The main objective of this section was to study whether the average power consumption of the FPGA depends on the values of the symbols composing the key. If this dependence exists, it could leak information about the key to an eavesdropper who is monitoring the power consumption of the SoC. Nonetheless, we did not expect this to allow a symbol-to-symbol side-channel attack to the QKD transmitter due to the rise time of the system t_r, which was expected to be larger than the period associated with the emission of each symbol, i.e $t_{\text{r}} \gg 10$ ns. This means that, even if the power consumption of the FPGA would ideally increase when emitting one symbol value (i.e. H) and decrease when emitting another

(i.e. V), the system would not be able to respond to this variation in a 10 ns time span. Thus, if this DC difference existed, we expected it to manifest in longer time intervals, thus to be related to the relative percentages of symbols on the key. Although not allowing a symbol-to-symbol side-channel attack of the QKD transmitter at a qubit repetition frequency of 100 MHz, the existence of such a difference could leak information about the relative percentage of H and V symbols in the key. Additionally, it can leak symbol-to-symbol information at lower qubit repetition frequencies.

We started this analysis by measuring the rise time of the system, in order to support the initial claim $t_r \gg 10$ ns. Then, we showed a brief example on how the average power consumption contains information about operations of the SoC. Afterwards, we studied the average power consumption dependence on the symbol values composing the key and analysed until which qubit repetition frequency this dependence could be exploited by a symbol-to-symbol side-channel attack. Notably, all the measurement and subsequent results were obtained using the experimental setup for the average power consumption measurements, detailed in Section 4.1.

4.4.1 Rise Time

The rise time of a system is defined as the time it takes a signal to go from a certain low value to a high value. In our case, we are interested in the voltage difference at the terminals of the 1Ω resistance and we study the rise time between the FPGA's inactivity and the beginning of emission. Figure 4.5 shows the power trace acquired while the FPGA starts emitting an Only-H sequence. The acquisition was done

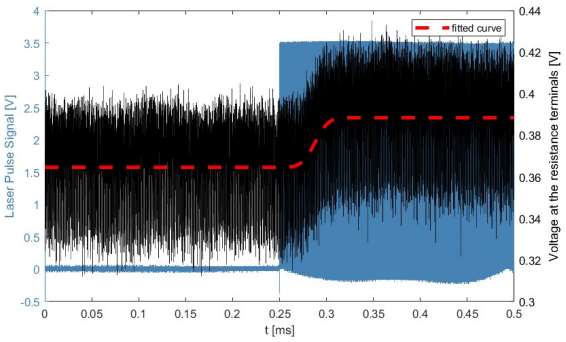


Figure 4.5: Trace acquired during the beginning of an emission

using a time scale of 50 μs/div. As mentioned in Section 4.1, the laser signal was used for triggering the acquisition. When the FPGA is inactive, the laser signal's average is 0 V and when it starts emitting, the signal rises sharply upon the emission of the first symbol. In Figure 4.5 one can see the laser signal rising at $t = 0.25$ ms, therefore symbolising the start of the emission. During emission, the average power consumption of the FPGA is higher than during inactivity. Nonetheless, the beginning of the emission at 0.25 ms does not correspond to an instantaneous rise in the power consumption, which is can be seen in Figure 4.5. To characterize this rise time, we modulate the system with a simple approach.

Let us consider $h(t)$ the impulse response of the system, this is, the system's output when presented by a brief input. For a linear time invariant system, the step response $a(t)$ can be obtained by the convolution

of $h(t)$ with the Heaviside function $H(t)$,

$$a(t) = (h * H)(t) = \int_{-\infty}^{+\infty} h(t)H(t-\tau)d\tau = \int_{-\infty}^{t} h(\tau)d\tau. \tag{4.7}$$

A system is said to have a Gaussian Response if its frequency response, $H(\omega)$, is given by

$$H(\omega) = e^{\frac{-\omega^2}{\sigma^2}}. \tag{4.8}$$

The impulse response $h(t)$ can be obtained by calculating the inverse Fourier Transform of the frequency response,

$$h(t) = \mathcal{F}^{-1}[H(\omega)](t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} H(\omega)e^{i\omega t} = \frac{\sigma}{2\pi}e^{-\frac{\sigma^2 t^2}{4}}\int_{-\infty}^{+\infty} e^{-u^2}du = \frac{\sigma}{2\sqrt{\pi}}e^{-\frac{\sigma^2 t^2}{4}}. \tag{4.9}$$

Finally, using Equation 4.7, we can find the step response function $a(t)$ for our system

$$a(t) = \int_{-\infty}^{t} h(\tau)d\tau = \frac{1}{2}\Big[1 + \text{erf}\Big(\frac{\sigma t}{2}\Big)\Big]. \tag{4.10}$$

Importantly, this response function characterizes a simplified system, which received the input at $t = 0$, has an initial voltage output of 0 V and a total voltage increase of 1 V. To account for all these parameters, we fit our voltage trace to $V(t)$,

$$V(t) = \frac{\Delta V}{2}\Big[1 + \text{erf}\Big(\frac{\sigma(t-t_0)}{2}\Big)\Big] + V_{t\to-\infty}. \tag{4.11}$$

By fitting the function in Equation 4.11 to the acquired power trace, we obtained the parameters for the fitted function shown in Figure 4.5 (Table 4.1). We define the rising time t_r as the time it takes for the system to go from 10% of rising, t_1, to 90%, t_2, such that

$$V(t_1) = V_{t\to-\infty} + 0.1\Delta V,\;\; V(t_2) = V_{t\to-\infty} + 0.9\Delta V,\;\; \Longrightarrow\;\; t_r = t_2 - t_1 = \frac{4}{\sigma}\text{erf}^{-1}(0.8). \tag{4.12}$$

Using the obtained σ, we calculate a rising time t_r of

$$t_r = (28.64 \pm 0.78)\ \mu\text{s}, \tag{4.13}$$

supporting the initial claim that $t_\text{r} \gg 10$ ns.

ΔV [mV]	σ [s^{-1}]	t_0 [ms]	$V_{t\to-\infty}$ [mV]
23.89 ± 0.08	$(127 \pm 3) \times 10^3$	0.288 ± 0.001	364.6 ± 0.2

Table 4.1: Parameters from the fitting of the rise time function

4.4.2 Central Processing Unit Operation

As detailed in section 4.1, the raw key data exchange between the CPU integrated in the SoC and the FPGA is done via the BRAM, a block RAM memory, of length $L_{\text{BRAM}} = 128$ kB [1]. When an external key is uploaded to the SoC, the BRAM is filled completely with key encoding bits. However, due to its finite size, a completely filled BRAM can only encompass the encoding of 524288 key symbols. If the length of the key uploaded by the external user to the CPU is larger that 524288 symbols, then, when the FPGA finishes reading the first half of the BRAM, the CPU begins writing the now empty half, while the FPGA reads the second half. This switching goes on until all the key has be written in the BRAM. During our analysis, we observed that the CPU's writing operation is visible through the average power consumption of the SoC.

Let us consider a user which uploads an external key encoding $1 \cdot 10^6$ symbols. The emission starts with the BRAM filled with the encoding of the first 524288 symbols. After half the symbols in the BRAM have been emitted by the FPGA, which at a qubit repetition frequency of 100 MHz, corresponds to a time interval of $\frac{1}{100 \cdot 10^6} \times \frac{524288}{2} = 2.621$ ms, the CPU starts writing the encoding of another 262144 symbols in the empty half of the BRAM. Afterwards, 786432 symbols have been encoded in the BRAM, meaning that there are still 213568 symbols remaining in the external key. Consequently, after the FPGA reads the BRAM's second half and moves to the now full first half, the CPU writes the encoding of the remaining symbols in the empty second half. This happens $2.621 \cdot 2 = 5.242$ ms after the emission begins.

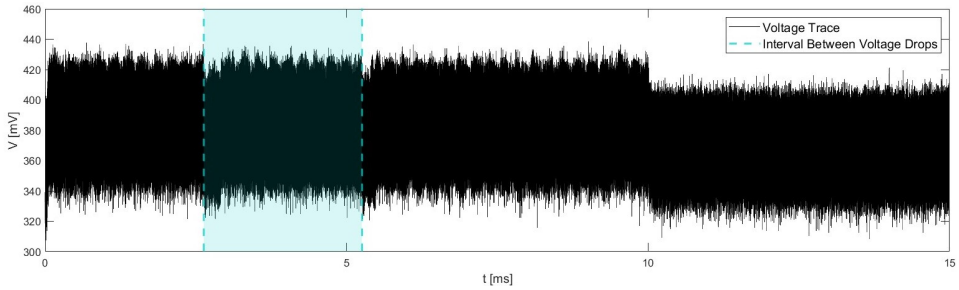


Figure 4.6: Trace acquisition corresponding to emission of 1 million random key symbols transferred externally to the SoC

Figure 4.6 shows the acquisition of a power trace during the emission of a 1 million symbol key, transferred to the SoC by uploading a file to the control interface. Since the key length was set to 1 million, at 10 ms we see the decrease in power consumption corresponding to the end of the emission. Noticeably, we saw two power consumption drops in the trace which appear to happen at times coincidental with the beginning of the CPU's writing operation. In order to calculate these times, we fit each power consumption decrease to the rise time function in Equation 4.11, where now $\sigma < 0$. Table 4.2 shows some of the parameters resulting from the fit, together with the falling time t_f, this is, the time it takes from the voltage to go from 90% of its initial value to 10%. We start by noting that the falling times are smaller than the rise time calculated in section 4.4.1. This can be justified by the smaller ΔV value, meaning that the power consumption drop associated with the CPU operation is smaller than the power increase upon the beginning of emission.

[1] The value for the length of the BRAM is not disclosed in [57] and it was obtained via a private conversation with the authors of the publication

	t_f [μs]	t_0 [ms]	ΔV [mV]
Voltage Drop 1	11.67 ± 1.7	2.637 ± 0.001	7.1 ± 0.1
Voltage Drop 2	16.04 ± 1.9	5.261 ± 0.001	7.4 ± 0.1

Table 4.2: Parameters obtained by fitting the rise response function to the power consumption drops in the trace corresponding to the emission of 1 million symbols

We can now calculate the times at which each power consumption drop was triggered, t_1 and t_2,

$$t_1 = t_{0,1} - \frac{t_{f,1}}{2} = (2.631 \pm 0.001) \text{ ms} \qquad t_2 = t_{0,2} - \frac{t_{f,2}}{2} = (5.253 \pm 0.001) \text{ ms} \tag{4.14}$$

The interval between the two voltage drops is $\delta t = t_2 - t_1 = (2.622 \pm 0.002)$ ms, meaning the results are consistent with a drop happening at approximately every δt. Figure 4.7 depicts the voltage trace acquired during the emission of a random 2 million symbol key, uploaded externally to the SoC. The vertical lines represent the integer multiples of the interval δt calculated previously. Visually we infer that this interval is also consistent with the power consumption drops in the 2 million symbol acquisition.

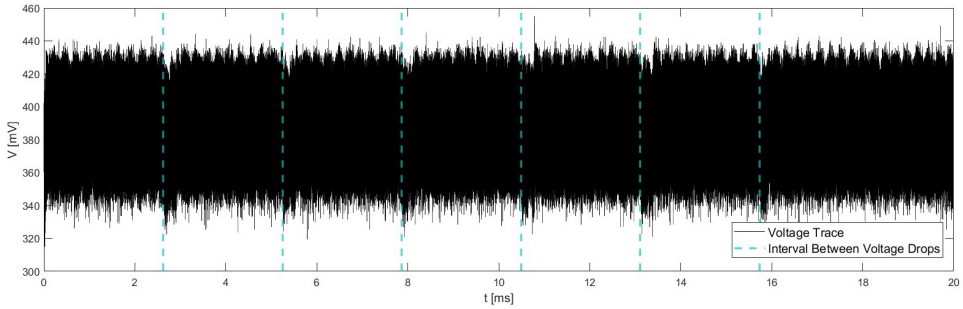


Figure 4.7: Trace acquisition corresponding to emission of 2 million random key symbols transferred externally to the SoC

In conclusion, we note that the calculated δt closely matches the time interval in which the FPGA emits the symbols encoded in half of the BRAM, thus allowing us to attribute the power consumption drops to the writing operation of the CPU.

4.4.3 Power Consumption Dependence on Symbol Value

After having characterized our system and seen that the average power consumption of the FPGA can encode information regarding the SoC's operation, we wanted to understand whether it can leak information regarding the symbols in the key. As previously debated, since $t_{resp} \gg 10$ ns, the information leakage does not manifest symbol-by-symbol. Thus, to perform this analysis, we started by considering the emission of fixed sequences with different percentage of H symbols in the key. Therefore, we used the Randomize Method, with the parameters displayed in Table 4.3.

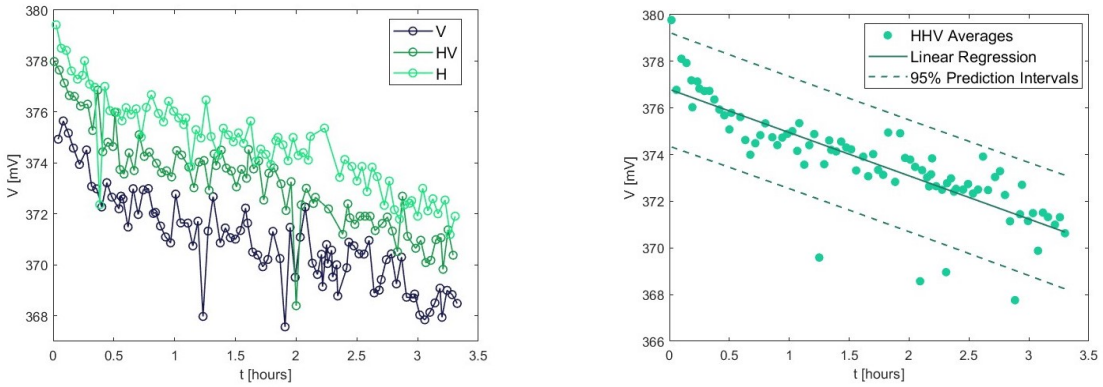
ΔT [s]	δt [s]	N_{seq}
30	10	400

Table 4.3: Randomize Method parameters for the analysis of the average power consumption of fixed-sequences with different percentage of H-symbols

With the chosen parameters, the entire acquisition takes approximately 3.33 hours, and the average time between the acquisition of two power traces corresponding to the same sequence is 2.5 minutes, assuring the independence condition in Section 4.3.1. Finally, we expect to acquire on average $400/5 = 80$ overall traces for each sequence.

Regarding the oscilloscope, the time scale was set to 500 μs/div, meaning that each power trace corresponds to the encoding of 500000 symbols. Given that we are only interested in the average power consumption value, which corresponds to the average of all the data points in the power trace, we can avoid transferring the entire waveform to the computer, thus saving time. After acquiring each power trace, the oscilloscope measures its average value and transfers it to the computer, which then stores it together with the acquisition time. We note that, during acquisition, some power traces showed an abnormally low average power consumption. This appeared to affect the traces randomly, such that we do not believe it can be attributed to an FPGA operation. In reality, we believe these power consumption drops are due to faults in the connection between the jack of the power cable and the ZedBoard's socket.

Figure 4.8 **(a)** displays the evolution in time of the average power consumption for Only-V, Only-HV and Only-H sequences. Each point in the graph represents an average over the encoding of 500000 symbols. The vertical scale in the oscilloscope was set to 15 mV/div, meaning that the uncertainty of each voltage data is 0.94 mV, thus having a negligible contribution to each average.



(a) **(b)**

Figure 4.8: **(a)** - Time evolution of the average power consumption of traces corresponding to different fixed-sequences; **(b)** - Linear regression for the data corresponding to the HHV sequence

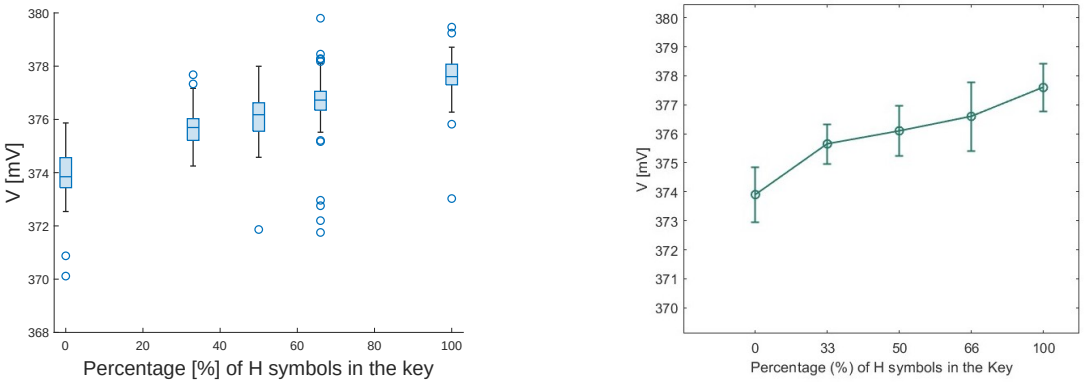
The average power consumption for the emissions of each fixed sequence showed a slow decrease over the acquisition time, which may be explained by a temperature dependence of the 1 Ω resistor.. Already it was apparent that Only-H sequences have a higher average power consumption than Only-HV ones, which in turn are higher than Only-V ones. Nonetheless, to quantify this difference, we had to mitigate the overall power decrease. To understand whether the power decrease affects all sequences equally, we performed linear regressions to the time evolution of the average power consumption for each fixed sequence. Figure 4.8 **(b)** shows the result of this fit to the data from the HHV sequence. Noticeably, there were 4 outliers which fall outside the 95% prediction interval, which corresponded to averages taken during the downwards power spikes. These spikes are ubiquitous to the voltage measurements, not only for the HHV sequence, and affected the traces randomly in time. They were also visible in Figure 4.8 **(a)**,

as expected. The results from the linear regressions $V = \beta_0 + \beta_1 t$ are displayed in Table 4.4. Generally, the β_1 parameters obtained for the different sequences are in agreement with a global power descent, independent of the sequence. We highlight the high value obtained for the HHV fit, which can be justified by a lower goodness-of-fit, as translated by the lower coefficient of determination R^2.

Sequence	β_1 [mV/h]	β_0 [mV]	R^2
Only-V	-1.61 ± 0.11	373.7 ± 0.21	0.73
Only-HVV	-1.79 ± 0.09	375.8 ± 0.17	0.86
Only-HV	-1.76 ± 0.10	376.2 ± 0.20	0.80
Only-HHV	-1.85 ± 0.14	376.8 ± 0.26	0.69
Only-H	-1.64 ± 0.10	377.4 ± 0.19	0.79

Table 4.4: Parameters from Linear Regression

To parameterize the power consumption descent as a global phenomenon, which approximately affected each sequence equally, we considered the average $\overline{\beta_1} = -1.73 \pm 0.1$ mV/h. In order to mitigate this effect, we shifted all data points (t, V) from all sequences to $(t, V - \overline{\beta_1} t)$, thus approximately centering the data for each sequence around its β_0 value.



(a) **(b)**

Figure 4.9: (a) - Scattering of average power consumption values for each H symbol percentage; **(b)** - Average power consumption value and associated standard deviation for each H symbol percentage

After shifting all the data points, we analysed the scattering of average power consumption values for each fixed-sequence, which is shown in the graph in Figure 4.9 **(a)**. The results showed a trend: the average power consumption increased with increasing percentage of H symbols in the key. This can also be seen in Figure 4.9 **(b)**, where the average power consumption for each fixed sequence is displayed, together with the associated standard deviation. The results showed a high distinguishability between Only-H and Only-V sequences. From the 76 average power consumption values collected for the Only-H sequence, only 2 were bellow the maximum value achieved by an Only-V sequence. Both these points can be attributed to the previously mentioned downwards power spikes. The difference between the average power consumption of an Only-H sequence, $\overline{PC}_{\text{Only-H}}$ and of an Only-V sequence, $\overline{PC}_{\text{Only-V}}$ is

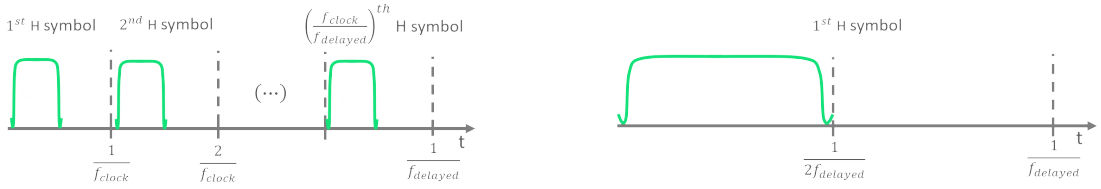
$$\overline{PC}_{\text{Only-H}} - \overline{PC}_{\text{Only-V}} = 3.7 \pm 1.8 \text{ mV}. \tag{4.15}$$

This means that, in the unrealistic scenario that the QKD transmitter is solely oscillating between the

emission of Only-H and Only-V, an eavesdropper could distinguish most of them by looking at the FPGA's power consumption. Regarding the Only-HVV, Only-HV and Only-HHV sequences, distinguishability cannot be assured, although their average power consumption increases respectively, obeying the trend. Additionally, the data from the HHV sequence (66% of H symbols) shows a higher number of outliers, justifying its higher standard deviation observed in Figure 4.9 **(b)**.

4.4.4 Simulating a QKD Transmitter with Lower Qubit Repetition Frequencies

As aforementioned, the average power consumption difference between the emission of an H symbol and a V symbol cannot manifest itself symbol by symbol at a qubit repetition frequency of 100 MHz, since the system's response is greater than the clock period of 10 ns. Nonetheless, it is possible to use the FPGA emitting at this rate to approximately simulate a smaller qubit repetition frequency. The FPGA outputs pulses with a 50% duty cycle: the phase modulator pulse has a high period in the first 5 ns upon the encoding of an H symbol, and in the second 5 ns upon encoding a V-symbol; the laser pulse has a 5 ns high period which is independent of the symbol value. For example, if we send 50000 H symbols at 100 MHz, the total emission time will be 0.5 ms, meaning that both laser and phase modulator pulses will have a non-zero voltage value during 0.25 ms. Now let us consider a QKD transmitter with a repetition frequency of 2 kHz, meaning that every 0.5 ms, Alice would send a qubit to Bob. Assuming the working principle of the FPGA is left unchanged, a smaller clock frequency would still correspond to a 50% duty cycle for the output pulses. Therefore, at 2 kHz, during the emission of a single H symbol, the output pulses have a non-zero voltage value during 0.25 ms, exactly as they did for the emission of 50000 H symbols at a 100 MHz clock rate. Generalizing, having the FPGA working at a qubit repetition frequency f_{clock}, one can imitate the high period of the output pulses emitted by a lower repetition frequency f_{delayed}, by emitting sequential sequences consisting of $n = f_{\text{clock}}/f_{\text{delayed}}$ same valued symbols. This is schematized in Figure 4.10, where the emission of 1 H symbol by an FPGA working at f_{delayed} is simulated by the emission of $n = f_{\text{clock}}/f_{\text{delayed}}$ H symbols by the FPGA at f_{clock}.



(a) **(b)**

Figure 4.10: (a) - Emission of $n = f_{\text{clock}}/f_{\text{delayed}}$ H-symbols by the FPGA working at f_{clock}; **(b)** - Emission of 1 H symbol by an FPGA working at f_{delayed}

Naturally, the simulation of lower qubit repetition frequencies is an approximation. We assume that the operation difference which causes the power consumption difference between H symbol and V symbol emission is still present at a lower clock frequency, in other words, we consider the working principle of the FPGA is left unchanged. Expectedly, at this lower clock frequency, the system can respond to the different power consumption requests from the FPGA in time, and the power traces will translate a symbol

to symbol power consumption difference between H and V emission. In this case, an eavesdropper would be able to retrieve the key simply by looking at the average power fluctuations in the traces.

In this section, we focused on quantifying the repetition frequencies for which the average power consumption of the FPGA contains symbol to symbol information about the key. To do so, we started by simulating a lower clock frequency of 2 kHz. At this frequency, the QKD transmitter sends symbols each 0.5 ms, meaning that, to simulate the emission of 1 H symbol, our FPGA working at 100 MHz must emit 50000 H symbols in a row. To understand whether at 2 kHz the power consumption difference is visible symbol by symbol, we looked at the power trace corresponding to the alternating emission of H and V by importing a key to the FPGA which simulates this. The key was encoded in a binary file and transferred to the FPGA through the control software. The file included the encoding of 450000 symbols, the first 50000 corresponding to H, the second 50000 corresponding to V, and so forth, oscillating between the two for the total emission of 9 symbols. For acquiring the power trace, we had to guarantee that it corresponded to and solely to the emission of the 9 symbols, this is, the oscilloscope acquisition started exactly at the beginning of the emission and ended exactly at the end of the emission. Therefore, before acquiring the power trace, we restarted the FPGA, ending emission of all output pulses, and left the oscilloscope in single acquisition mode. The rising edge of the laser pulses was used as a trigger, such that, when emission starts, the oscilloscope starts acquiring when the rising edge of the first laser pulse surpasses the 1 V limit. The time scale of the oscilloscope was set at 500 μs/div, and the delay set to -2 ms, thus allowing to retrieve the rising of the power consumption and the emission of the 9 symbols. An example of a power trace is shown in Figure 4.11, together with a schematic representation of the time intervals corresponding to each symbol.

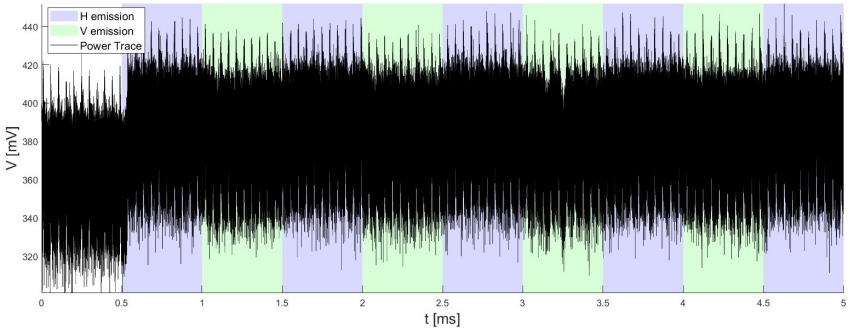
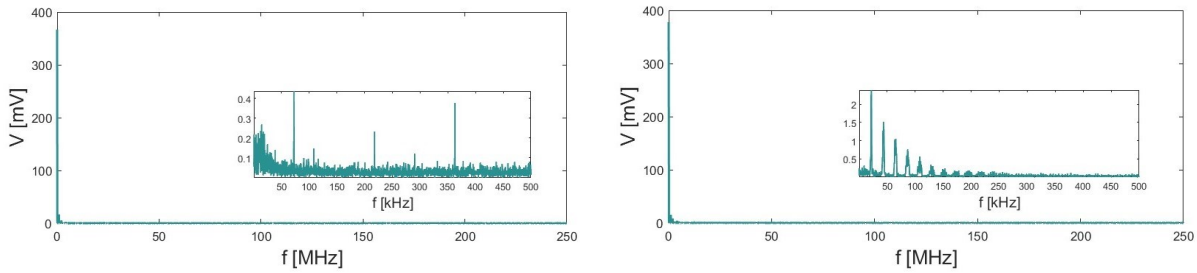


Figure 4.11: Power trace taken during the emission of a 2 KHz symbol key, alternating between H and V emission

The trace showed a high amount of noise, ubiquitous to power consumption measurements. Namely we note the peaks affecting the waveform at a frequency around 20 kHz, present before and after $t = 0.5$ ms, the time instant at which emission begins. Generally, the trace parts corresponding to the emission of H symbols (blue shaded area) show a higher average power consumption than the ones corresponding to the emission of V symbols (green shaded area). However the high amount of sampled data per period ($f_{\text{samp}} = 500$ MSa/s and $T = 0.5$ ms), together with the high amount of noise, made it difficult to see this difference.

Before moving forward with the characterization of the power consumption at lower clock frequencies,

we wanted to understand where the noise, namely the peaks affecting the waveform, were coming from. Since they were also present in the trace before the FPGA starts emitting, it was important to understand whether they translated an FPGA operation unrelated to emission, or they came from the AC power supply. Therefore, we collected two power traces while the FPGA was not emitting: one using a DC power supply and the other with the current setup so far, this is, the FPGA connected to the AC power supply of the building. The FFTs of both traces are displayed in Figure 4.12, **(a)** corresponding to the DC power supply. Both spectra show a strong DC component, as expected, and some noise in the kHz range, which is shown respectively in the amplified axis. Focusing on the right most spectrum, we see a first peak at 21.8 kHz and then an harmonic pattern, with following peaks at 43.6 kHz, 65.4 kHz, 87.2 kHz and so forth. The 21.8 kHz frequency bin translates the peaks affecting the power consumption measurements. Given that this peak, and its following harmonic pattern, is not present in the left most FFT, we conclude it is being introduced by the AC power supply. Since this 21.8 kHz noise posed a limit for the low qubit repetition frequency study, given that it could mitigate possible power consumption differences for a QKD transmitter emitting at frequency close to 20 kHz, we switched to a DC power supply for the following measurements.



(a) DC Power Supply

(b) AC Power Supply

Figure 4.12: FFTs of power traces taken during FPGA inactivity.

Having changed to a DC power supply, we acquired a new power trace for the 2 kHz clock frequency. To make the power consumption difference more visible, we started by filtering the power trace. Using MATLAB's filtering capabilities, we used a infinite impulse response filter with a lowpass bandwidth of 2 kHz, meaning that frequencies above the qubit repetition frequency we simulated are attenuated. Afterwards, we applied a running average to the data, where we considered data snippets of a fixed length and averaged them. For example, considering a snippet length of 1000 samples, the first 1000 samples in the power trace were averaged, and this averaged value corresponded to the first point in the final power trace, and so forth. This procedure can be perceived as reducing the sampling frequency of the trace. The initial power trace was acquired with a sampling frequency of 500 MSa/s. In the previous example, a data length of 1000 samples means the final trace after the running average will have a sample frequency of 500 kSa/s.

Figure 4.13 shows the power trace for the 2 kHz repetition frequency after data treatment. The power consumption between H and V emission was now more evident, appearing as a modulation of a 2 kHz wave in the trace. We note that the trace was plotted starting at the $t = 0.5$ ms, this is, at the start of FPGA emission. Nonetheless, there is a sharp increase around the first 80 μs of the treated power trace, which corresponds to the rising time of the power consumption upon starting emission. In Figure 4.11 we

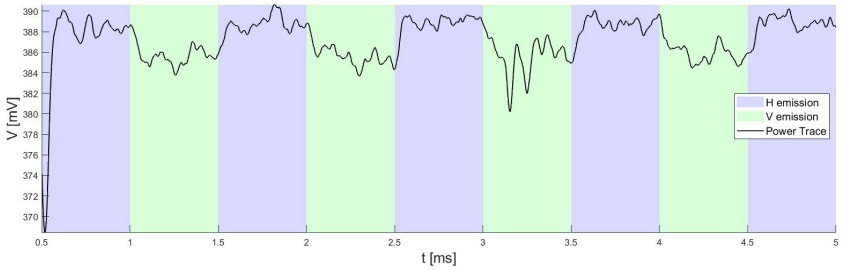


Figure 4.13: Power trace at 2 kHz after filtering and running average treatment

can see that, even though emission started at 0.5 ms, it takes some time for the power consumption to reach the higher value required by the FPGA, and this in the treated trace in Figure 4.13 corresponds to the initial rising peak. Additionally, we noted that at ≈ 3.12 ms, the power consumption showed an abrupt decrease and subsequent destabilization. This is consistent with a power consumption drop caused by the CPU's writing operation, which should happen $\delta t = 2.62$ ms after the emission start $t = 0.5$ms. The power consumption drop was also observed in the untreated trace depicted in Figure 4.11. Additionally we observed that, even after changing to a DC source and filtering, there is still noise in the power trace, else the treated trace would just correspond to a square wave with period 1 ms.

To understand whether the power consumption difference is always distinguishable at the 2 kHz qubit repetition frequency, we acquired more power traces corresponding to an HV sequence and evaluated whether all encoded symbols could be guessed correctly from the traces. To do so, we formulated a way to discern which part of the treated traces encoded a V and which encoded an H. A first approach would be to calculate the average power consumption of each 0.5 ms interval; if an average is smaller that the previous one, it corresponds to a V symbol and vice-versa. However, since the sample size for the running average was 1000 samples, which was smaller than the number of samples per 0.5 ms period, there were averages which contain H symbol and V symbol trace parts. Additionally, the power response in the H-V switching is not instantaneous, in order words, the average power consumption does not increase or decrease abruptly. Therefore, a better approach was to consider the power consumption value at the beginning and at the end of each 0.5 ms period. If the value decreased, this would mean the power consumption decreased, and thus the period corresponded to the emission of a V symbol, and the contrary for an H. Using the previously discussed acquisition settings and data treatment, 20 traces were obtained for the 2 kHz clock frequency. Using the previously debated prediction approach, each trace was analysed and used to predict 8 of the 9 symbols it encoded, where the first symbol is discarded due to the rising time effect on the trace. With this approach we obtain an 100% prediction accuracy, where we define prediction accuracy as

$$\text{Prediction Accuracy}(\%) = \frac{\text{Nº of Correctly Guessed Symbols}}{\text{Total Nº of Symbols in the Key}} \times 100. \tag{4.16}$$

The prediction accuracy with a random guessing method, this is, for each symbol in an infinite sized key, one randomly predicts between H and V, would be 50%. Therefore, a successful prediction accuracy lies

above the random guessing probability, where

$$\text{Random Guessing} = 50\% \tag{4.17}$$

The 100% prediction accuracies obtained at 2 kHz mean that, at this repetition frequency, the power consumption of an H and a V symbol is completely distinguishable. We then wished to see until which qubit repetition frequency this distinguishability was present. Supposedly, we are bounded by the bandwidth BW of our power system, meaning, the maximum qubit repetition frequency for which distinguishability should present corresponds approximately to

$$\text{BW} \approx \frac{0.35}{t_{\text{rise}}} = (12.22 \pm 0.95)\ \text{kHz}. \tag{4.18}$$

To test our hypothesis, we applied the procedure considered for the 2 kHz frequency to all the frequencies between 2 kHz and 46 kHz. For each repetition frequency, we uploaded a key to the SoC, working at a qubit repetition frequency of 100 MHz, which simulated the emission at the lower frequency of an H followed by a V and so forth. The oscilloscope settings were the same as the ones for the 2 kHz acquisition, meaning that each trace translated information regarding the first 4.5 ms of emission. The number of key symbols encoded in these 4.5 ms depends of the repetition frequency we wish to simulate. For the 2 kHz case, the trace collected information regarding $4.5 \cdot 10^{-3} \cdot 2 \cdot 10^{3} = 9$ symbols, but, for example, for the 7 kHz case we collect $7 \cdot 10^{-3} \cdot 2 \cdot 10^{3} = 31.5$ symbols. In this case, we neglect the information regarding the last 0.5 symbol, and calculate the prediction accuracy considering the total number of symbols in the key as 31. The same approach was followed for all the repetition frequencies which lead to power traces with a non integer number of symbols. The traces were treated and analysed using MATLAB. The code was developed such that each trace is filtered with a low pass filter, with bandwidth set to the qubit repetition frequency the trace corresponds to. For example, the power trace corresponding to a 23 kHz repetition rate is filtered for all frequencies above 23 kHz. The running average was also performed to all traces, and, to assure coherence between all the frequencies, the number of samples N_{samp} per average was set to,

$$N_{\text{samp}} = 0.1 \times \frac{f_{\text{samp}}}{f_{\text{delayed}}}, \tag{4.19}$$

Figure 4.14 shows two examples of traces acquired at different repetition rates, after the filtering and the running average.

In Figure 4.14, the power consumption drop from CPU's writing operation is observed on both power traces. Before, when only dealing with traces taken at a 2 kHz repetition frequency, this effect was negligible, since it happened in a time interval smaller than the period $T = 0.5ms$, and it always happened during the emission of a V symbol, as opposed to in between symbol emission. However, for different repetition frequencies, this power consumption drop and subsequent destabilization affects the traces in different ways. As the emission period decreases with an increasing repetition frequency, the effect encompasses more symbols, affecting the prediction accuracy in unpredictable ways. Therefore, to calculate the prediction accuracy, the symbols corresponding to the time interval associated with the CPU

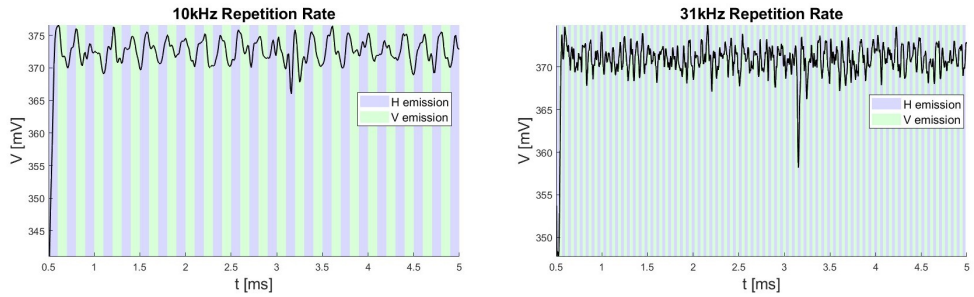


Figure 4.14: Power traces acquired at 10 kHz and 31 kHz repetition frequency, after filtering and running average treatment

writing operation were not considered. This means, for example, that the 6^{th} symbol in the 2 kHz voltage traces was not taken into account when calculating the prediction accuracy, whilst for the 10 kHz traces, symbols 27 to 29 were discarded.

Figure 4.15 depicts a box chart with the prediction accuracy distribution for keys taken at different repetition frequencies. For each frequency, 10 power traces were analysed, and for each trace, a prediction accuracy calculated. For qubit repetition frequencies until 11 kHz we obtained a prediction accuracy of 100% for all the power traces. From the 12 kHz repetition frequency forward, we saw a slow decrease in the prediction accuracy, which is in agreement with the expected behaviour for frequencies above the calculated bandwidth of the power system, where there is a cutoff in the distinguishability between H and V emission.

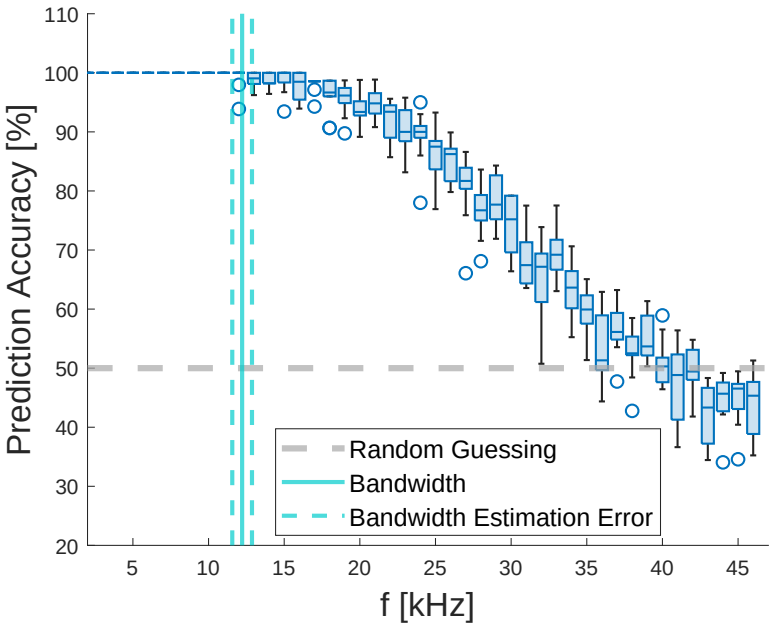


Figure 4.15: Prediction accuracy distribution for different power traces at different repetition frequencies. The calculated system's bandwidth is displayed together with its 95% confidence interval

From this result we can conclude that, for QKD setups working at low qubit repetition frequencies, if there is an electronic driver, e.g FPGA, with a power consumption which depends on the symbol value of each qubit, then power side-channel can reach high prediction accuracies with a simple symbol-to-symbol

analysis of the power consumption values.

4.5 Frequency Spectrum Analysis of the Power Consumption

The main objective of this section is to characterize the frequency spectrum of the power traces, identify possible information leakages and explore them for hacking the key. Given that we are focusing on the frequency domain, the bandwidth of our system is of crucial importance. Therefore, the power traces were acquired with the two oscilloscope probes with a 350 MHz bandwidth, using the setup schematized in Figure 4.2 **(b)** . Nonetheless, data taken with the same setup, but with two Tektronix P2220 voltage probes with a bandwidth of 200 MHz and 1 MΩ impedance will also be considered. Although having a lower bandwidth, these probes can be impedance matched to the oscilloscope, which can only switch between an input impedance of $50\ \Omega$ or 1 MΩ. This, together with the fact that these probes do not have an in-built signal attenuation, unlike the P2220 350 MHz probes, reduces the noise associated with the data acquisition. Data taken with these two probes will be analysed and compared. Unless stated otherwise, the measurements were taken with the 350 MHz probes.

4.5.1 Fixed Sequences

We started by analysing the frequency spectrum of power traces during the emission of fixed sequences. To do so, we set the FPGA to emit H symbols indefinitely and successively triggered the oscilloscope using the rising edge of the laser pulses, in order to acquire power traces. Having set the time scale of the oscilloscope to $200\ \mu$s/div, each power trace corresponded to the emission of 200000 H symbols. After the acquisition of 50 traces for the Only-H emission, the process was repeated for the emission of V symbols. We then calculated the FFT of each power trace, averaging all that corresponded to the emission of 200000 H symbols, and all that correspond to the emission of 200000 V symbols. The average spectrum for each emission is shown in Figure 4.16 between the frequencies of 50 MHz and 900 MHz. Each frequency bin displays the average magnitude, together with its standard deviation. For a

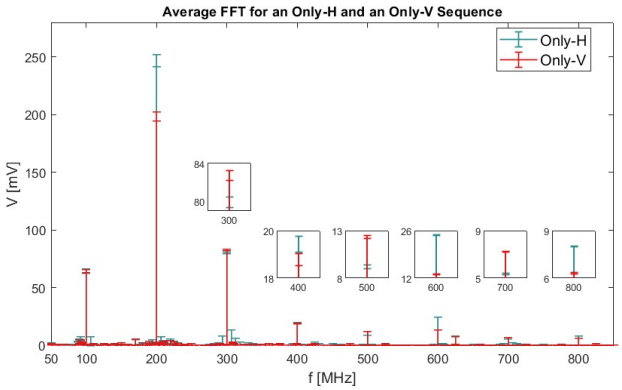


Figure 4.16: Average spectrum for the emission of 200000 H symbols (Only-H) and the emission of 200000 V symbols (Only-V) in the $[50, 900]$ MHz frequency range

given frequency bin, if the average magnitude differs depending on the sequence, and the respective

standard deviations do not overlap, then the magnitudes at that frequency are distinguishable. Therefore we note that the magnitudes for all the frequencies in this range which are integer multiples of 100 MHz show complete distinguishability between an Only-H and an Only-V sequence. Consequently, in these frequencies, we observe an information leakage in the magnitude of the frequency bins. When averaging all the FFTs corresponding to emissions of the same sequence, we smooth out the noise in the spectrum. To understand how this noise affected the distinguishability between spectra, we computed the correlation coefficients, at zero delay, between the different pairs of FFTs. The higher the correlation value is, the more similar the two FFTs are two one another. To analyse the results, we organized them in matrix form

$$\left[\begin{array}{ccc|ccc} h_1,h_1 & \cdots & h_1,h_{50} & h_1,v_1 & \cdots & h_1,v_{50} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{50},h_1 & \cdots & h_{50},h_{50} & h_{50},v_1 & \cdots & h_{50},v_{50} \\ \hline v_1,h_1 & \cdots & v_1,h_{50} & v_1,v_1 & \cdots & v_1,v_{50} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{50},h_1 & \cdots & v_{50},h_{50} & v_{50},v_1 & \cdots & v_{50},v_{50} \end{array}\right], \tag{4.20}$$

where x_i represents the FFT of the i^{th} power trace taken during an Only-X emission, $x \in \{\mathrm{H}, \mathrm{V}\}$, and x_i, x_j the correlation value between x_i and x_j.

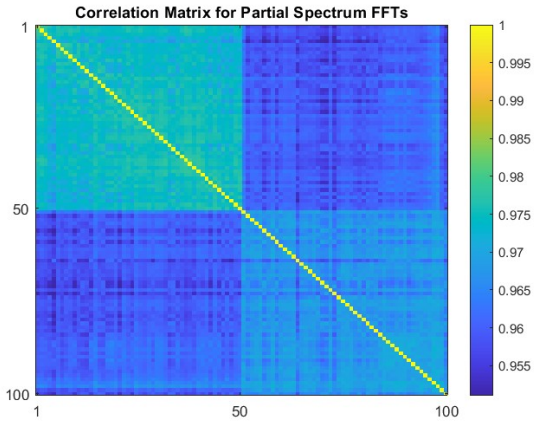
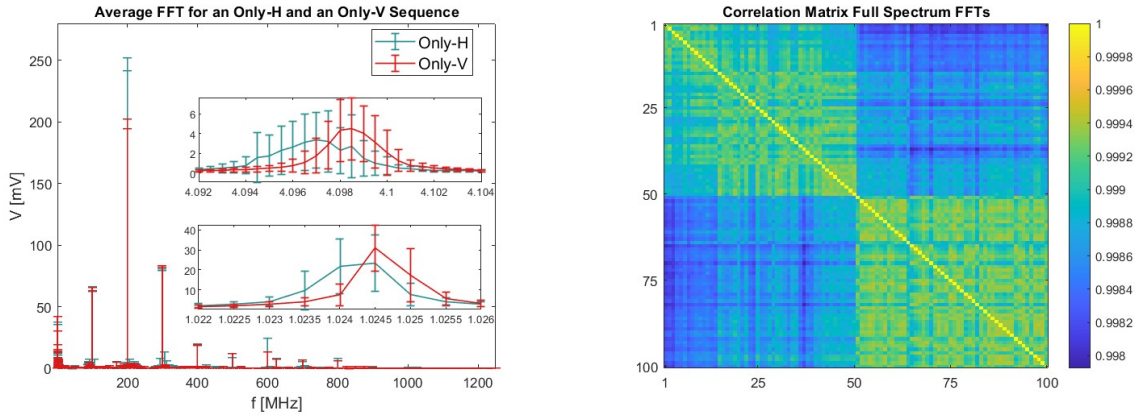


Figure 4.17: Correlation Matrix for the FFTs of Only-H and Only-V sequences in the $[50, 900]$ MHz range

Figure 4.17 shows the FFT correlation matrix, only considering the $[50, 900]$ MHz frequency range. Here we see that the top-left and bottom-right 50×50 sub matrices, which correspond to correlations between FFTs of the same sequence, showcase higher values. Therefore, it is possible to distinguish between the spectrum of Only-H and Only-V sequences without filtering the noise.

Until here we focused on the $[50, 900]$ MHz frequency range, where we attributed the main distinguishability between spectra to the amplitude of the frequency bins. Nonetheless, the same analysis can be expanded to the whole range. In Figure 4.18 **(a)**, we present the whole spectrum of the average FFT of an Only-H and an Only-V emission. As an example, we focus on the frequencies around the 4 MHz and the 1 MHz frequency bins. Here, we saw that the maximum amplitudes achieved are similar, however, the

frequencies which reach these amplitudes are slightly shifted. Regarding the correlation matrix in Figure 4.18 **(b)** we observe that, when considering the full spectrum, the similarity between same sequence FFTs increases, this is, the top-left and bottom-right 50×50 sub matrices show higher coefficients. However, the lowest values achieved for the full spectrum are around 0.998, whilst for the partial spectrum they were around 0.955, meaning that the similarity between Only-H and Only-V sequences increased as well. Therefore, we can conclude that the main information leakage lies in the $[50, 900]$ MHz spectral range, and as we increase the range, the information content added does not surpass the lower frequency noise which is also added to the FFTs.



(a) **(b)**

Figure 4.18: Full spectrum analysis. **(a)** - Average spectrum of Only-H and Only-V sequences in the full frequency range. Focus on the 4 MHz and 1 MHz frequency bins; **(b)** - Correlation Matrix for the full spectrum FFTs of Only-H and Only-V sequences

So far we had analysed FFTs of power consumption signals during the emission of 200000 symbols. We can define the FFT Resolution, also known as the spectral resolution, as the interval between two consecutive frequency bins. The higher the interval is, the less amount of frequency bins there will be in the FFT, and the less information one will have about the spectrum. Therefore, we associate high intervals to low FFT Resolutions. The FFT Resolution is related to the number of time domain samples, also known as the Record Length,

$$\text{FFT Resolution} = \frac{f_{\text{sample}}}{\text{Record Length}}. \tag{4.21}$$

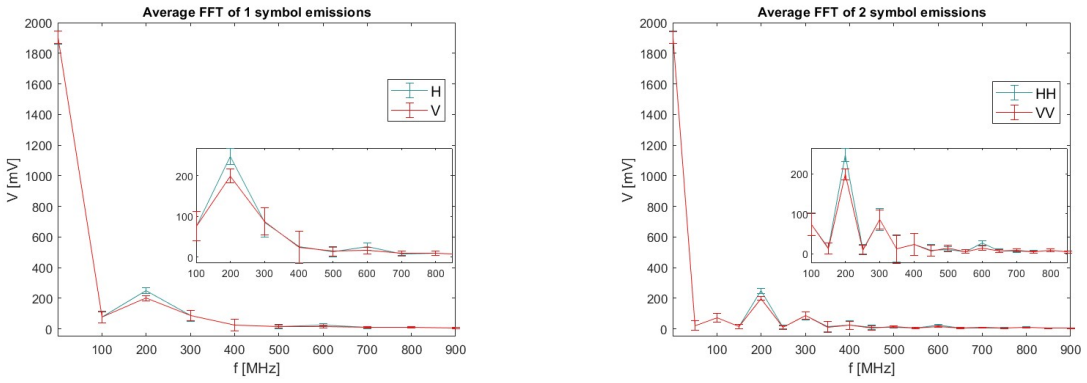
Therefore, as we increase the Record Length, the FFT Resolution also increases. Given that $f_{\text{sample}} = 2.5$ GSa/s and the qubit repetition frequency is 100 MHz, we have 25 samples per symbol emission, meaning that the FFT Resolution of the previous FFTs of 200000 symbols is

$$\text{FFT Resolution} = \frac{2.5 \times 10^9}{25 * 200000} = 500\text{Hz}. \tag{4.22}$$

Therefore, we had a frequency bin every 500 Hz. However, in a realistic symbol-to-symbol hacking scenario, we will want to consider the FFTs of smaller sized sequences. The frequency spectrum of the emission of 200000 symbols can only transmit information if all the symbols are either the same, as in the

previous case, or differ in a known way, for example the emission of an Only-HV sequence. In a realistic QKD scenario, the keys are emitted randomly, thus the frequency spectrum of the emission of 200000 symbols will not hold valuable information. Nonetheless, as we decrease the size of the sequence, the spectral resolution will also decrease. For example, the FFT of the emission of only one symbol will have a resolution of 100 MHz. The low spectral resolution of small sized sequences poses a big impediment for discerning possible distinguishability in spectra corresponding to different sequences. As the spectral resolution decreases, the information content decreases as well, and the noise in the signal becomes a stronger impediment for possible information leakages.

To understand how much the FFT resolution affects distinguishability, we must start with two spectra which, with high spectral resolution, are a priori distinguishable. Then we reduce the resolution and see whether it is possible to distinguish between the two spectra. Therefore, we used the previously obtained data retrieved for Only-H and Only-V sequences, given that we know their spectra is distinguishable when considering the emission of 200000 symbols. The 50 power traces of Only-H sequences were thus divided into snippets of n symbols. For example, if $n = 2$, the power traces were divided into snippets of 50 samples, each corresponding to the emission of 2 symbols, in the end producing $5 \cdot 10^6$ snippets corresponding to the emission of HH. The same process was followed for the 50 power traces of Only-V sequences. Then, the FFT of each snippet was individually calculated and finally all the FFTs corresponding to the emissions of the n same symbols were averaged. In this way, we were able to obtain the average frequency spectrum of n symbols during the emission of a fixed sequence. Figure 4.19 shows in **(a)** and **(b)** the average spectra for $n = 1$ and $n = 2$ respectively.



(a) $n = 1$ **(b)** $n = 2$

Figure 4.19: Spectrum of n sized sequences extracted from fixed sequence emission.

In both cases we are able to discern distinguishability between different symbol emissions, however it is now much smaller than the one observed in Figure 4.16, where the FFT resolution was much higher. Nonetheless, the fact that this distinguishability is present for such low spectral resolutions allows us to extend the study to the analysis of small sized sequences, which we will do in the following section. Nonetheless, we note that the average spectra displayed in Figure 4.19 corresponds to symbol emissions extracted from fixed sequences. Therefore, we cannot affirm, for example, that the average spectrum for an H emission in a realistic QKD scenario is the one displayed in Figure 4.19 **(a)**. The fact that the n symbol emissions were taken out of a fixed sequence emission makes it so that they are not independent

from each other, and their average spectra can show an increased distinguishability due to memory effects. To eliminate these constraints, and study a realistic QKD scenario, the next section will focus on the the emission of random keys.

4.5.2 Sequences in Random Key Emission

As aforementioned, the emission of fixed sequences is not a realistic QKD scenario. From the analysis in Section 4.5.1 we concluded that the spectra of different fixed sequence emissions is different. Now we wish to extend this study to the spectra of sequences during random key emission. If there is a difference between the spectra of different sequences emitted randomly, then this can be exploited by, for example, a correlation-based hack to extract information about the key at the qubit repetition frequency of 100 MHz.

To emit a random key, we encoded the key string in a binary file and transferred it to the FPGA, following the procedure detailed in Section 4.3. The key length was set to 200000 symbols, meaning that the file must consist of 50000 bytes. We note that the random key emission mentioned is, in reality, pseudo-random, given that the key string was created with Python's pseudo-random number generator. In a real QKD scenario, the string must be generated with a QRNG. Nonetheless, pseudo-randomness already provides the framework necessary for validating a possible hacking strategy, thus, from this point forward, we will neglect its difference from true-randomness.

To study the spectrum of different sequences during random emission, we first needed to acquire a data set consisting of randomly emitted symbols and the power consumption values during their emission. For this, we collected 5 power traces, each taken during the emission of an independent random key consisting of 200000 symbols. In order to map the symbols in the key to the data in the power traces, we had to guarantee that for each emitted key, the oscilloscope started acquiring at the beginning of the FPGA's emission and collected data during the time interval necessary for the FPGA to emit 200000 symbols. Therefore, we left the Oscilloscope in single acquisition mode and use the laser pulse exiting the FPGA to trigger the acquisition. Before emission, the laser pulse had an average voltage of 0 V. When the emission started, the pulse rose, triggering the power consumption acquisition. To guarantee that the acquisition covers the 2 ms necessary for the key emission, we set the time scale of the Oscilloscope to 200 μs/div and the time delay to -1 ms. By acquiring the power traces this way, and using the maximum sampling frequency allowed by the oscilloscope in a 3 channel configuration, 2.5 GSa/s, we were able to associate the first 25 samples of the trace to the first symbol in the key and so forth.

After data acquisition, we had a data set consisting of $1 \cdot 10^6$ randomly emitted symbols and their corresponding power consumption data. To study the spectrum of different sequences, we started again by analysing their average spectra. Before, each one of the five power traces were normalized, this is, for all the samples in a given trace, we subtracted the average voltage and divided by the standard deviation of the trace. With the normalized traces, we then studied the average FFTs of different sequences, for which we considered the following methodology:

1. Consider sequences S_L^i, where L is the number of symbols in the sequence and i the number of possible sequences of size L, $i \in \{1, \ldots, 2^L\}$

2. Divide the symbols in the data set into the 2^L possible sequences. Each sequence S_L^i occurs in the data set an average of

$$\frac{1 \cdot 10^6}{L \cdot 2^L} \tag{4.23}$$

times, meaning the higher the length, the less times the sequence appears in the data set.

3. Divide the power consumption data accordingly

4. Calculate the FFTs of all the occurrences of each sequence and average them according to the symbols they correspond to. After this step one has 2^L average FFTs, F_L^i, one per sequence S_L^i.

5. Compute the correlation matrix of the FFT averages

$$\begin{bmatrix} F_L^1, F_L^1 & \cdots & F_L^1, F_L^{2^L} \\ \vdots & \ddots & \vdots \\ F_L^{2^L}, F_L^1 & \cdots & F_L^{2^L}, F_L^{2^L} \end{bmatrix} \tag{4.24}$$

To further understand this, let us consider the example of $L = 2$, this is, sequences comprising of 2 symbols. In this case, there are 4 possible sequences, $S_2^i = \{$HH,HV,VH,VV$\}$, each occurring on average $1.25 \cdot 10^5$ times in the data set. Therefore, the average FFT of an HH sequence will correspond to an average over approximately $1.25 \cdot 10^5$ FFTs, and the same for the other three sequences. The correlation matrix for $L = 2$ will have the form

$$\begin{bmatrix} vv, vv & vv, hv & vv, vh & vv, hh \\ hv, vv & hv, hv & hv, vh & hv, hh \\ vh, vv & vh, hv & vh, vh & vh, hh \\ hh, vv & hh, hv & hh, vh & hh, hh \end{bmatrix}, \tag{4.25}$$

where, for example, $vv = F_2^1$ represents the average FFT of a VV sequence.

By averaging all the FFTs corresponding to the emission of the same sequence, two outcomes are achieved. Firstly, the signal noise will be smoothed out, increasing distinguishability. Secondly, we eliminate all memory effects from the spectrum, given that all occurrences of the sequence were preceded and succeeded by other completely random sequences. For example, retrieving the $L = 2$ case, of the $1.25 \cdot 10^5$ occurrences of an HH sequence, approximately one quarter of them were preceded by a VV sequence, another quarter by an VH sequence, and so forth. By averaging all the occurrences, we nullify the impact these previous sequences might had on the spectrum.

Figure 4.20 shows the correlation matrix for $L = 2$. The main diagonal in the matrix corresponds to the auto correlation coefficients, which are one. However, in addition to the diagonal, we can see that there are other correlation values extremely close to one. To address these values, we define a k-diagonal as the set of entries $A_{i,j}$ for which $j = i + k$. Therefore, we see that the higher correlation values lie in -2-diagonal and the 2-diagonal. Importantly, these diagonals represent correlations between sequences which share the first symbol and differ in the second symbol, telling us that the average FFT of a two symbol sequence depends on the value of the first symbol.

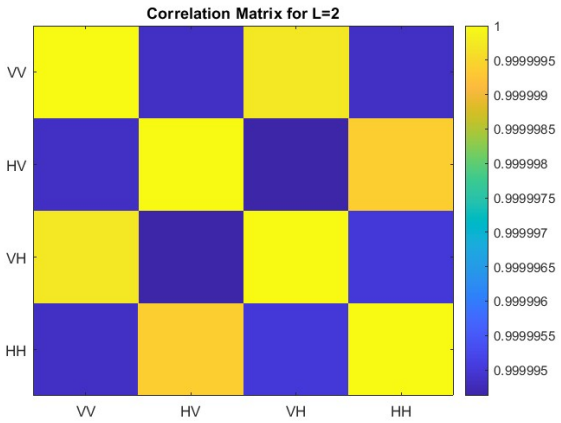


Figure 4.20: Correlation Matrix for the average FFTs of two symbol sequences, considering the full frequency range and the magnitude at each frequency.

Different components of the frequency spectrum can be studied with this methodology. Meaning that, after calculating the FFTs of all sequences, one can focus on the complex value at each frequency bin, its magnitude or its phase. If we focus, for example, on the magnitude at each frequency, the average FFT of a sequence will, in reality, correspond to the average magnitude distribution for the different frequencies in the FFT. Consequently, when computing the correlation matrix, we are studying the similarity between the different frequency contributions to signals corresponding to different sequences. This is exactly what was done to obtain the correlation matrix in Figure 4.20. In Figure 4.21 we can see the average spectra which led to this matrix. By visual inspection of the whole spectrum, no distinguishability can be found between the different sequences. Nonetheless, focusing on the first 8 integer multiple frequencies of 100 MHz, we can see small differences between the magnitudes of different sequences. These differences are very small, namely in the range of 1 to 2 mV, which justifies the very high correlation values between sequences which do not share the first symbol in the matrix of Figure 4.20. This being said, it is not the absolute value of these correlations that we can use to extract information, but their relative one. In other words, it is the fact that the amplitudes of HH and HV sequences are more similar to each another then to VH and VV, whilst the ladder are more similar to each other, that can be used to extract information.

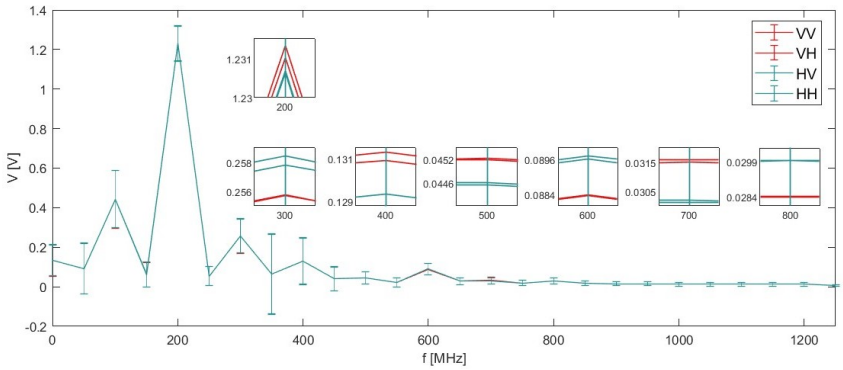
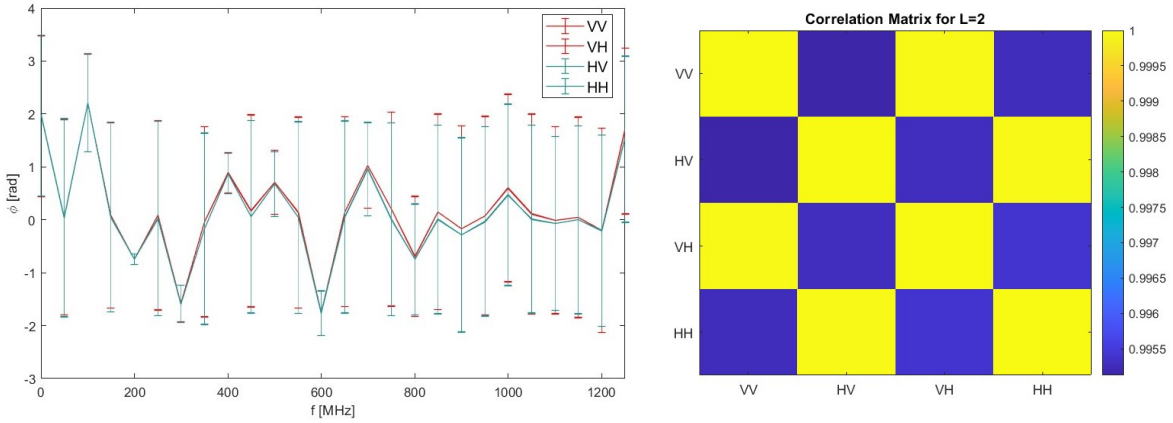


Figure 4.21: Average magnitudes for the full FFT spectrum of two symbol sequences emitted randomly.

Until now we have been considering the magnitude at each frequency bin, but, as aforementioned, we can also consider the phase. Figure 4.22 **(a)** shows the average phases of the frequency spectrum of 2-symbol sequences.



(a) Average phases of the FFT spectra

(b) Correlation Matrix for the average FFT phases

Figure 4.22: Phase analysis of the FFTs of 2-symbol sequences emitted randomly

Here we see an increased distinguishability between the sequences starting with an H symbol and the ones starting with a V. This increase is proved by the corresponding correlation matrix, displayed in Figure 4.22 **(b)**, where the lowest correlation value is 0.9951, between HV and VV, whilst for the correlation matrix between the FFT magnitudes, in Figure 4.20, the lowest value was achieved between the HV and VH sequences, and corresponded to 0.9999. Nonetheless, it is important to note that the standard deviations are high, meaning that, phases in the non averaged FFTs corresponding to the same sequence are not stable. This is mainly due to the noise in the signals, combined with the low frequency resolution in FFTs of sequences with 2 symbols. Even so, given that for the average magnitudes, the standard deviations were also high, we concluded that there is more information leakage in the phases of the frequencies composing the 2-symbol spectra than in the magnitudes. Therefore, from this point forward, we will focus on former.

In summary, we have reached two important conclusions from the study of 2 symbol sequences, $L = 2$. Firstly, we concluded that the frequency spectrum of the power consumption during the emission of 2 symbols, this is, during the time interval of 20 ns, depends on the first emitted symbol. In addition, we also concluded that this dependence is stronger for the phase of each frequency, in respect of the magnitude. With these conclusions in mind, we then moved to the study of larger sized sequences, namely $L = 4$, $L = 6$ and $L = 8$.

Figure 4.23 **(a)** displays the correlation matrix for $L = 4$, having considered the phases at each frequency of the FFTs. In this case, there are $2^4 = 16$ possible sequences of 4 symbols, meaning that each average FFT corresponds to an average over approximately 15625 FFTs. The correlation matrix is a 16×16 matrix, where again the main diagonal corresponds to the auto-correlation coefficient. Additionally, we distinguish some diagonals with correlation values close to 1. Among these, the $-12, -4, 4, 12$-diagonals correspond to the correlation coefficients between sequences which share the first two symbols. On the other hand, the $-8, 8$-diagonals represent the correlation values between sequences which only differ

in the last symbol. These higher correlation values follow the trend seen for $L = 2$ sequences, this is,

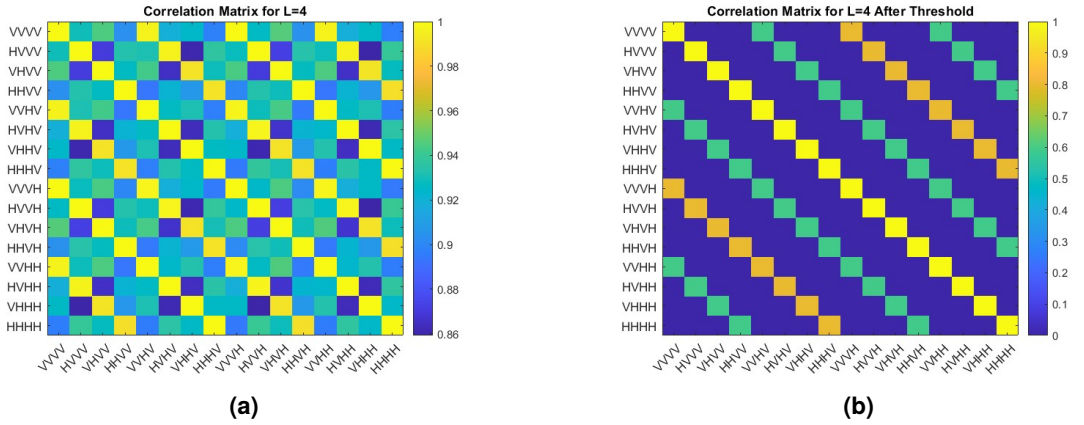


Figure 4.23: Correlation matrix for the FFTs of 4-symbol sequences without **(a)** and with **(b)** an applied threshold.

sequences which share the first symbols have higher correlated FFTs. To better understand possible differences between these high correlations, we apply a threshold to the matrix, retrieving the matrix **(b)** in Figure 4.23. The threshold is applied in the following way: all coefficients in the initial matrix which fall below 0.99 are set to 0, all between 0.99 and 0.9995 are set to their initial value minus 0.4, and all between 0.9995 and 1 are set to their initial value minus 0.2. This allows us to highlight differences between coefficients close to 1, which wasn't possible with the range of the initial matrix. Again we find a pattern in the matrix after applying a threshold, this is, the coefficients in the $-12, -4, 4, 12$-diagonals are lower than in the $-8, 8$-diagonals. In other words, four symbol sequences which share the first three symbols are more correlated than the ones which only share the first two symbols. This can be seen as an extension of the $L = 2$ case, whereas now, the correlation range is broader. The lowest correlation value achieved for $L = 4$ is 0.8592, between HVVV and VHHH, which is smaller than the lowest value achieved in the $L = 2$ case, 0.9951. Therefore, the distinguishability between sequences differing in the first symbols is increased in the $L = 4$ case. This can also be see in Figure 4.24, where the average phases of the FFT spectrum for different 4 symbol sequences are shown. By comparing the spectra in this figure with the ones from the $L = 2$ case, in Figure 4.22, we can visually see an increased distinguishability for the $L = 4$ case.

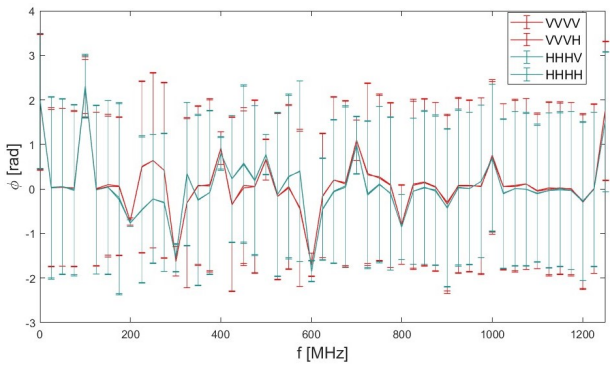
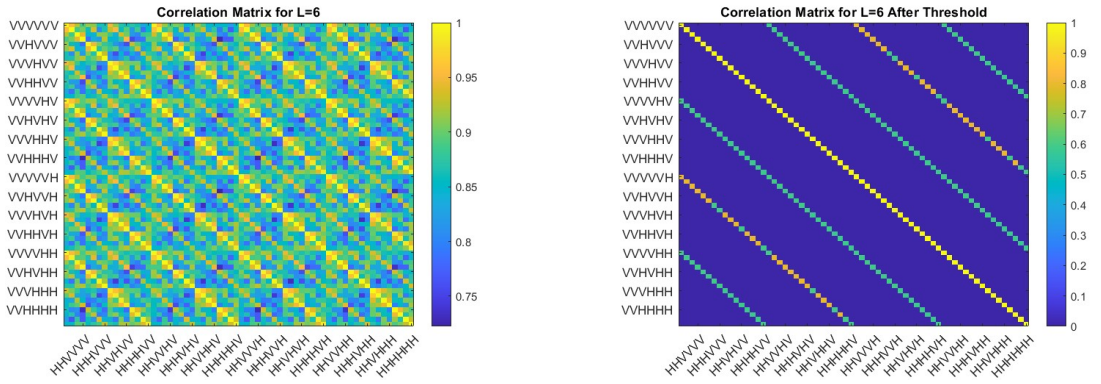


Figure 4.24: Average phases of the FFT spectrum of different 4 symbols sequences

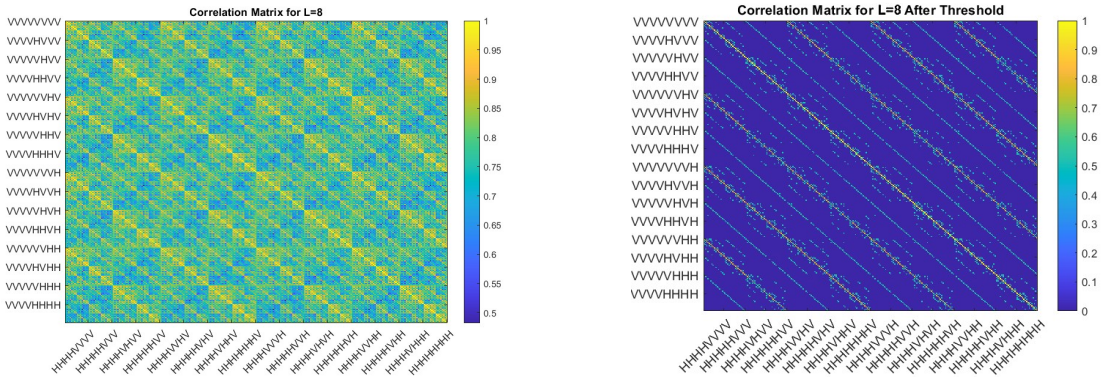
We identify two possible reasons for this increase, the first one being the internal operation of the FPGA, which we treat as a pseudo-black box, and do not propose to explain. The second reason is that as the sequence size increases, so does the resolution of the FFTs, thus increasing the information content of each frequency bin.



(a) (b)

Figure 4.25: Correlation matrix for the FFTs of 6-symbol sequences without **(a)** and with **(b)** an applied threshold.

Lastly, for completeness, we apply the same analysis for the $L = 6$ and the $L = 8$ case. Figure 4.25 shows the correlation matrix for the $L = 6$ case, where again we find the same correlation trend regarding the first 4 and 5 symbols in the sequence, highlighted by the application of a threshold. In this case, the threshold was applied such that coefficients below 0.972 were set to 0, ones between 0.972 and 0.99635 set to their initial value minus 0.4, and finally the ones between 0.99635 and 1 set to their initial value minus 0.2. These smaller threshold values, in comparison to the $L = 4$ case, showcase the higher distinguishability between sequences in the $L = 6$ case. This is also seen in the coefficient range of the initial correlation matrix, where the lowest achieved coefficient is now 0.7224, between the sequences VVHHHV and HHVVVH.



(a) (b)

Figure 4.26: Correlation matrix for the FFTs of 8-symbol sequences without **(a)** and with **(b)** an applied threshold.

Figure 4.26 shows the correlation matrix for the $L = 8$ case before and after applying a threshold. Again, in the initial matrix, we see 6 diagonals with very high correlation values, which are the $-192, -64, 64, 192$-

diagonals and the $-128, 128$-diagonals, corresponding to coefficients between sequences which share the first 6 and 7 symbols respectively. However now, the coefficients in the $-128, 128$-diagonals are not, in general, higher than the ones in the $-192, -64, 64, 192$-diagonals. In fact, when applying a threshold to the initial matrix, there is no value which allows us to distinguish between these diagonals. This can be seen by analysing the matrix after the threshold, where the values chosen were the following: coefficients below 0.9 were set to 0, ones between 0.9 and 0.973 set to their initial value minus 0.4, and ones between 0.973 and 1 set to their initial value minus 0.2. Here, we see that the $-128, 128$-diagonals display similar correlation coefficients to the ones in the $-192, -64, 64, 192$-diagonals. Nonetheless, we see other highlighted diagonals with smaller coefficients than the former, namely the $32, 96, 160, 224$-diagonals, which correspond to sequences sharing the first 5 symbols. In other words, for the $L = 8$ case, the sequences which share the first 5 symbols are highly correlated, but less than the ones which share the first 6 and 7 symbols. Finally, we note that the distinguishability between sequences which do not share initial symbols decreased compared to the previous considered cases. This can be seen in the coefficient range of the initial matrix, where the lowest achieved value is now 0.4821, between HHVHVVVH and VVHVHHHH.

In conclusion, the power consumption of the FPGA during the emission of random symbols can leak information about the key. This information leakage is present in the frequency spectrum of the power consumption, and can be explored for hacking the key, as we will see in the next section.

4.5.3 Hacking the key at 100 MHz repetition frequency

In the previous section we have seen that the similarity between the average FFTs of different sized sequences increases with the amount of shared initial symbols. Nonetheless, the correlation values between sequences which do not share initial symbols is, in general, high. Given so, the hacking strategy much rely on extracting information in the relative correlation values, not the absolute ones. Therefore we propose a template attack based on the 'FFT Fingerprints' of the system. Here, we assume that the eavesdropper has initial access and control of the QKD transmitter, or of one equal to it. During this initial stage, the eavesdropper uses this control to store power consumption data and the keys the data corresponds to. These keys are not to be used for any type of data encryption, and are just to provide the eavesdropper with information about the system. With the power consumption data, the eavesdropper computes the FFT averages of L sized sequences, i.e., for $L = 4$, Eve computes the average FFT during the emission of HH, the average FFT during the emission of VV, and so forth. These four FFT averages are the 'FFT Fingerprints' for $L = 4$. For an arbitrary L, we define FFT Fingerprint as the average FFT of the power consumption during the emission of a certain sequence with L number of symbols. Note that, for our hack, we considered the average phase at each frequency bin, as opposed to the magnitude.

In the second phase of the hack, Eve loses control over the QKD transmitter, but keeps monitoring the power consumption of the FPGA. It is in this phase that Alice and Bob use the QKD system to share secret keys, not knowing that Eve has had previous access and is still monitoring the power consumption of the FPGA. When the key sharing starts, Eve uses the power consumption data and accordingly computes the FFTs of snippets of data corresponding to the emission of L symbols. To guess these symbols, she

uses the FFT Fingerprints, and calculates which is more likely to correspond to the FFT of the symbols she doesn't know. She then uses this probability to guess part of the symbols, and performs this over the power consumption data taken during the emission of the key she wants to guess.

In the proposed hacking strategy there are two parameters which Eve must choose: L, the length of the sequences for which the FFT Fingerprints are calculated, and δN, the number of symbols she guesses after finding the best match for an FFT, such that $0 < N \leq L$. To better understand this, let us consider the example of a hacking strategy with $L = 4$ and $\delta N = 1$, schematized in Figure 4.27.

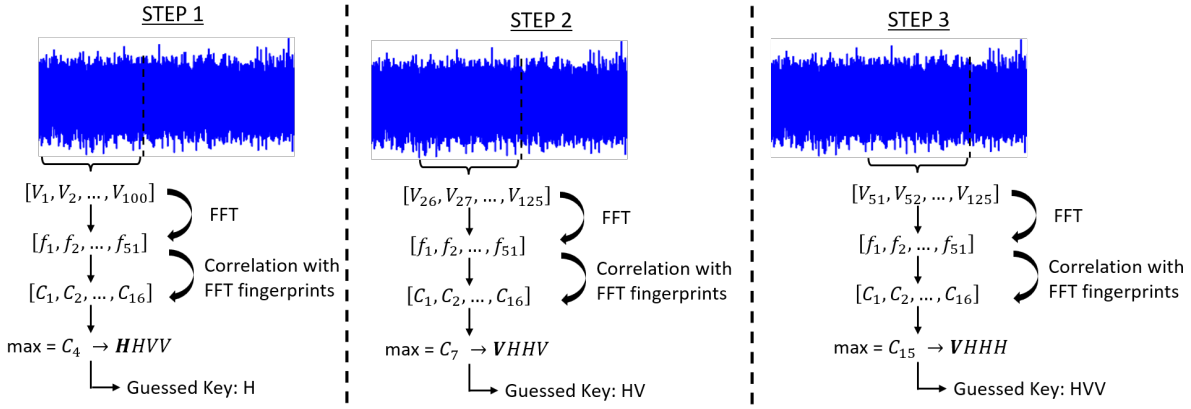


Figure 4.27: Schematics of the first three steps of hacking strategy with $L = 4$ and $\delta N = 1$.

Here, Eve uses the data she collected in the first part of the attack to compute the FFT Fingerprints of sequences with 4 symbols. Then, in the second phase, she retrieves the power consumption trace of the key she wants to guess. Knowing the sampling frequency of her data, which in this case we consider 2.5 GSa/s, and the qubit repetition frequency of the FPGA, which in this case is 100 MHz, she starts by analysing the first 100 samples, knowing they correspond to the first 4 emitted symbols. She computes the FFT of this data snippet, and then calculates the correlation coefficients of this FFT with the 16 FFT Fingerprints she has at her disposal, each corresponding to one of the 16 possible sequences with 4 symbols. After this, she will be left with 16 correlation coefficients, and she uses them to find which FFT Fingerprint corresponds to the maximum correlation. Finally, knowing which sequence this FFT Fingerprint corresponds to, given that $\delta N = 1$, she uses the value of its first symbol to guess the first symbol in the key. Then, she collects another snippet of 100 samples, but this time considering the second to fifth symbol emission. Repeating the same procedure, she guesses the second symbol in the key. In this case, by repeatedly applying this procedure, she will be able to guess all the symbols in the key except for the last three.

In general, for an arbitrary L and $\delta N = 1$, Eve is not able to guess the last $L - 1$ symbols in the key. However, this effect is negligible at the qubit repetition frequency of 100 MHz when using oscilloscopes with a record length in the order of a milion samples. A power trace with $1 \cdot 10^6$ samples, at $f_{samp} = 2.5$ GSa/s, encodes 40000 symbols. Therefore, for any L value which falls below 100, the non-guessed symbols at the end would compose less than 0.25% of the key encoded in the power trace, thus being negligible.

To prove the feasibility of this hacking strategy, we hacked 15 independent random keys. For the first

phase of the hacking strategy, we used a previously obtained set of $1 \cdot 10^6$ random symbol emissions and their corresponding power consumption data, which acquisition was detailed in Section 4.5.2. Following the methodology detailed in Section 4.5.2, we used the data set to calculate the FFT Fingerprints of L sized sequences. For the acquisition of the power consumption data during the emission of each one of the 15 random keys, we used again the same acquisition process detailed in Section 4.5.2. Each power trace, corresponding to the emission of one key encoding 200000 symbols, was stored by the computer, together with the encoded key, which was used to quantify the performance of the hack.

Two hacking trials were performed, one with a pair of oscilloscope probes with a bandwidth of 350 MHz, and another with a pair with 200 MHz bandwidth. For each trial, a different set of 15 independent random keys was considered. The FFT Fingerprints were calculated using a data set acquired with the according probes, meaning that, for example, in the 200 MHz acquisitions, the FFT Fingerprints were calculated considering a previous data set of $1 \cdot 10^6$ random symbols, whose power consumption was acquired also with 200 MHz probes. The two hacking trials were performed with a 1 month time interval between them and the performance of both will be shown. In general, as explained in Section 4.5, power consumption data acquired with the 200 MHz probes is less prone to noise than the one acquired with the 350 MHz ones.

To quantify the hacking performance we calculate the prediction accuracy, defined in Equation 4.16, for each of the 15 keys in each hacking trial. Given that each key consists of 200000 symbols, each symbol assuming a value between H or V, we consider the hacking of a key successful [16] if it exceeds random guessing, as defined in Equation 4.17, by 3 standard deviations of the binomial distribution, this is,

$$\text{Prediction Accuracy}(\%) > \left(0.5 + 3\frac{\sqrt{200000 \times 0.5 \times 0.5}}{200000}\right) \times 100 = 50.34\%. \tag{4.26}$$

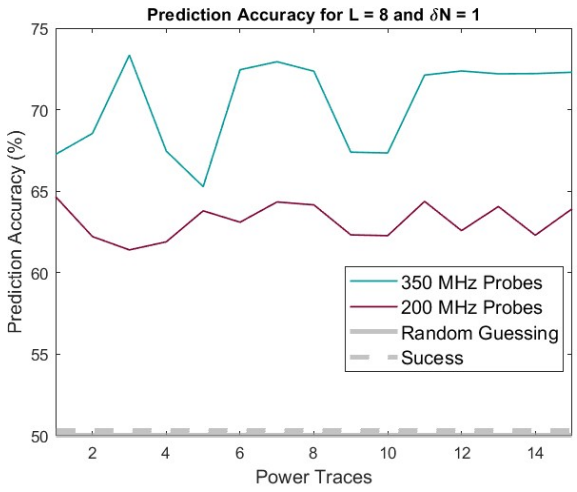
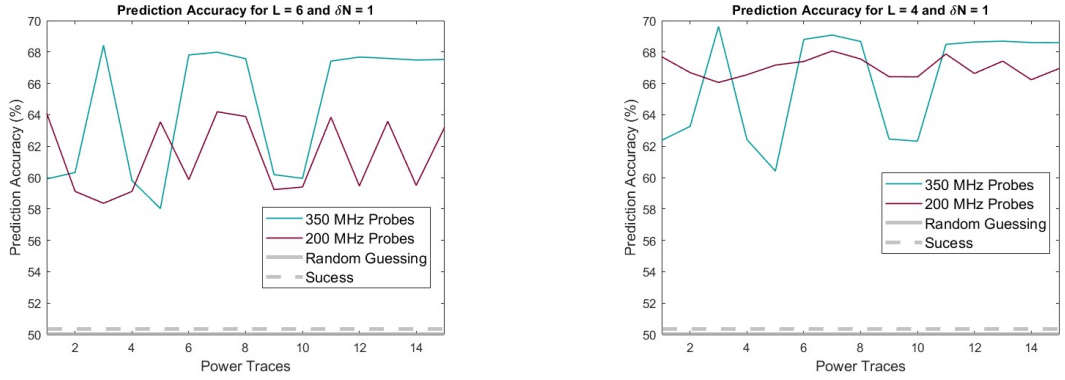


Figure 4.28: Prediction Accuracy for a $L = 8$ and $\delta N = 1$ hacking strategy

With a hacking strategy with $L = 8$ and $\delta N = 1$, we were able to achieve a maximum prediction accuracy of 73.35%, which lies above the success probability by a large amount. The prediction accuracy achieved for each one of the 15 power traces in the two hacking trials is shown in Figure 4.28. The performance of the hacking trial with the 350 MHz probes surpasses the one with the 200 MHz ones.

This tells us that the bandwidth of the acquisition system is more important to the hacking performance then the amount of noise associated with the measurements. Although this is true for the $L = 8$ and $\delta N = 1$ case, it is not necessarily true for other parameters. As L decreases, the resolution of the FFTs decreases as well, making noise a crucial impediment for the prediction of the key. To understand whether the experimental data sustain this claim, and also to understand how the L and δN parameters affect the hacking performance, we repeated the trials for $\delta N = 1$ and smaller values of L. In Figure 4.29 we can see the Prediction Accuracy results for $L = 6$, in **(a)**, and $L = 4$, in **(b)**. In both cases, the prediction



(a) **(b)**

Figure 4.29: Prediction Accuracy for a hacking strategy with $L = 6$ and $\delta N = 1$ **(a)** and with $L = 8$ and $\delta N = 1$ **(b)**.

accuracy decreased for all the power traces in comparison with the $L = 8$ case. For $L = 6$, the trial with the 350 MHz probes still performed generally better than the one with the 200 MHz ones. However, this is not true for the $L = 4$ case, where even though some power traces show higher prediction accuracy values in the 350 MHz trial, the values for the 200 MHz trial are higher on average. Generally, the $L = 4$ hacking strategy performed better than the $L = 6$ one, which can tell us that the increase in the FFTs resolution in this case does not translate in more information leakage. The spectral resolution for $L = 6$ is approximately 16.67 MHz, meaning that the FFTs computed in this case do not have frequency bins located at the integer multiple frequencies of the 100 MHz frequency, which, as we have seen before, are one of the main sources of information leakage. For the $L = 8$ and $L = 4$ case, the spectral resolution is 12.5 MHz and 25 MHz respectively, meaning that the FFTs in both cases have frequency bins in the integer multiple frequencies of 100 MHz. Finally, a strategy with $L = 2$ and $\delta N = 1$ was also applied, and the results are shown in Figure 4.30. In this case we see a strong decrease in the prediction accuracy. Nonetheless, all hacking attempts in both trials performed above the success probability, meaning that even with sequences with 2 symbols, and thus FFTs with 50 MHz of spectral resolution, it is still possible to extract information about the key. As seen in Figure 4.30, the hacking trial with the 200 MHz probes performs better for all the attempts, sustaining the initial claim that as the spectral resolution decreases, noise becomes a larger impediment for guessing the key and the bandwidth of the system a less important requisite.

Until now we have studied the performance of the L parameter, having kept δN constant and equal to one. To understand the impact of the former, we hacked the 15 keys keeping L constant and changing

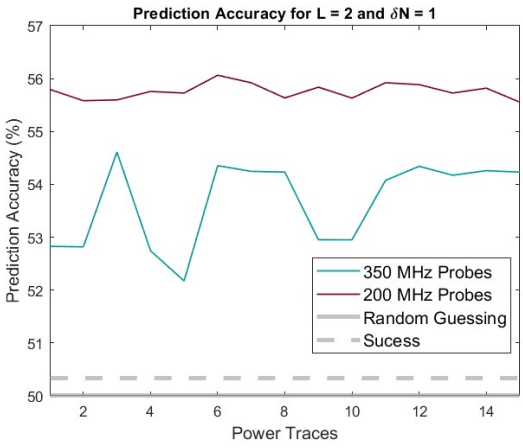
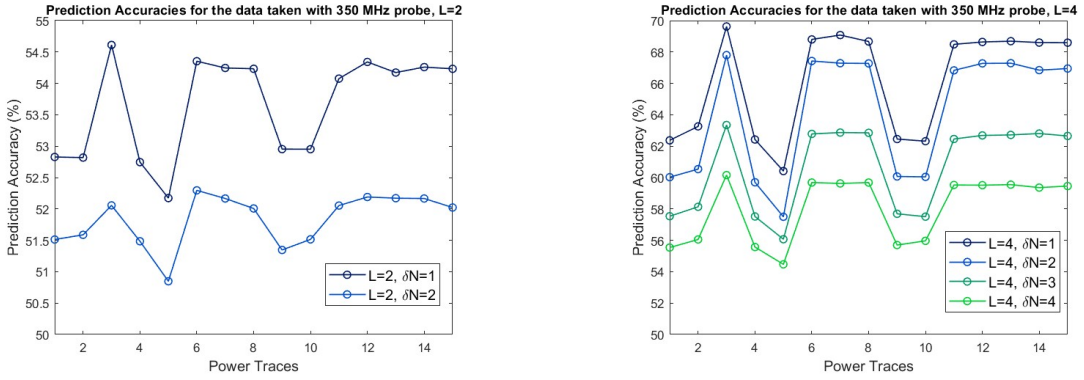


Figure 4.30: Prediction Accuracy for a $L = 2$ and $\delta N = 1$ hacking strategy

δN. The results for $L = 2$ and $L = 4$ are shown in Figure 4.31, in **(a)** and **(b)** respectively, where for simplicity, only the trial with the 350 MHz probes was considered. In both we see a decrease in the hacking performance as δN increases. This behaviour follows the conclusions from the correlation study in Section 4.5.2, where in general, sequences showed higher correlations with other sequences which shared the most amount of initial symbols. This, in our hacking strategy, means that after calculating the FFT of a given snippet and finding the FFT Fingerprint with the higher correlation coefficient, we have a high degree of certainty on the first symbol in the snippet. This certainty decreases for the second symbol, and so forth for $L > 2$. Therefore we see that the hacking strategy with lowest performance is the one that combines the lowest FFT resolution with the highest δN value allowed, this is, $L = 2$ and $\delta N = 2$. Nonetheless, even in this case, all hacking attempts were successful.



(a) **(b)**

Figure 4.31: Prediction Accuracies for hacking strategies with different δN and constant $L = 2$ in **(a)** and $L = 4$ in **(b)**.

Finally, it is worth mentioning why higher values of L were not considered, since we have seen that, in general, increasing the sequence length by powers of 2 increases the performance of the hack. As L increases, the number of times each sequence occurs in the initial data set decreases, as seen in Equation 4.23. The occurrences of a given sequence in the initial data set are used for calculating its FFT Fingerprint, thus, if there is not a representative amount of them, their average will not produce a

valid fingerprint. For example, if $L = 16$, which would correspond to increasing $L = 8$ by 2, there are approximately 0.95 occurrences of each sequence in our initial data set, according to Equation 4.23. In fact, on average, there will be sequences which do not occur at all, thus being impossible to calculate their FFT Fingerprint. Nonetheless, even considering L values which are not powers of 2 but are higher than 8, for example $L = 10$, the number of occurrences of each sequence in the initial data set are not enough for a valid FFT Fingerprints calculation. A possible solution for this problem is to increase the size of the initial data set, which we leave for possible next steps.

In conclusion, we were able to reach a Prediction Accuracy of 73.35% with a template based hack, considering correlations between the phases of frequencies in the FFTs of sequences with 8 symbols. This result shows a high level of information leakage in the spectral content of the power consumption during the emission of keys at a 100 MHz qubit repetition frequency. Since this is the first time a power consumption side-channel attack has been performed to a QKD framework, the achieved prediction accuracy represents an important milestone in this novel field.

Chapter 5

Conclusions

In this thesis, we have made the first known attempt to perform power side-channel analysis to a QKD system. This eavesdropping approach breaches the assumption made by security proofs of QKD protocols that the laboratories of Alice and Bob are secure locations where no unwanted information is leaked. In this work, we propose an attack on Alice's laboratory, namely, on the power consumption of the FPGA controlling the electro-optical components of the transmitter.

During this analysis, we have found that the average power consumption of the FPGA depends on the symbol value of the emitted key. Nonetheless, a variation in the average power consumption upon the emission of different symbols cannot be seen at high qubit repetition frequencies, for example 100 MHz, since the system is bounded by its bandwidth. Therefore, we have proven that until qubit repetition frequencies close to the system's bandwidth, we are able to distinguish between the emission of an H and the one of a V by analyzing the difference in the average power consumption of the FPGA. We found that, for frequencies until 11 kHz, this method achieves an 100% accuracy for distinguishing between H and V emission.

To hack the key at the qubit repetition frequency of 100 MHz, a typical frequency for mature QKD distribution systems, we analyzed the frequency spectrum of the power consumption. We have found that, during the emission of finite sized sequences, the spectrum of the power consumption, mainly the phase of each frequency, depends on the values of the first symbols in the sequence. By exploiting this dependence, we were able to hack the key in a realistic QKD scenario, where the transmitter emits a key composed of a random string of symbols at a qubit repetition frequency of 100 MHz. Via the implementation of a template attack, using a correlation-based approach, we were able to achieve a maximum accuracy of 73.35% for the prediction of a random key. We believe this result, together with the fact that all hacking attempts in the two hacking trials performed above the success threshold, proves the feasibility of this novel hacking approach to QKD systems. Looking into the future, there are some improvements that can be implemented to possibly increase the prediction accuracy, namely:

- Increase the sampling frequency and bandwidth of our acquiring system. Currently, our oscilloscope has a maximum sampling frequency of 2.5 GSa/s. By using an oscilloscope with a higher sampling frequency, we would be able to increase the number of samples per symbol and thus increase the

resolution of the FFTs. Additionally, using oscilloscope probes with a bandwidth higher than 350 MHz would reduce the attenuation of frequencies in the $[400, 900]$ range, which were proven to be a source of information leakage.

- Perform the template attack with an initial data set of bigger size, in order to implement a hacking strategy with higher values of the L parameter, namely, $L = 16$ and $L = 32$.
- Employ a neural network for signal recognition

This being said, the results achieved until now prove that the power consumption of electronic controllers in QKD setups can be a source of information leakage, thus opening an unexplored path for cryptanalysis in QKD. Therefore, this work has opened new questions, namely:

- What counter measurements can be employed in order to mitigate the found information leakage?
- Given that the power consumption of the FPGA depends on its programming, what is the behavior of the power consumption in FPGAs of other QKD transmitters?

In conclusion, in this work we were able to exploit the power consumption of the FPGA controlling the electro-optical components of the QKD transmitter in the University of Padua to guess 73.35% of the emitted key. This is the first implementation of a power side-channel attack to a QKD system, and the positive results have unlocked new research questions and problems which will be approached in future work.

Bibliography

[1] R L Rivest, A Shamir and L Adleman. 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems'. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://doi.org/10.1145/359340.359342.

[2] P W Shor. 'Algorithms for quantum computation: discrete logarithms and factoring'. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[3] Frank Arute et al. 'Quantum supremacy using a programmable superconducting processor'. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510. ISSN: 14764687. DOI: 10.1038/s41586-019-1666-5.

[4] Han-Sen Zhong et al. 'Quantum computational advantage using photons'. In: *Science* 370.6523 (2020), pp. 1460–1463. DOI: 10.1126/science.abe8770. URL: https://www.science.org/doi/abs/10.1126/science.abe8770.

[5] Lars S. Madsen et al. 'Quantum computational advantage with a programmable photonic processor'. In: *Nature* 606.7912 (June 2022), pp. 75–81. ISSN: 14764687. DOI: 10.1038/s41586-022-04725-x.

[6] Michele Mosca. *Cybersecurity in an era with quantum computers: will we be ready?* Cryptology ePrint Archive, Paper 2015/1075. 2015. URL: https://eprint.iacr.org/2015/1075.

[7] Yang Liu et al. 'Experimental Twin-Field Quantum Key Distribution over 1000 km Fiber Distance'. In: *Physical Review Letters* 130.21 (May 2023). ISSN: 1079-7114. DOI: 10.1103/physrevlett.130.210801. URL: http://dx.doi.org/10.1103/PhysRevLett.130.210801.

[8] Wei Li et al. 'High-rate quantum key distribution exceeding 110 Mb s–1'. In: *Nature Photonics* 17.5 (2023), pp. 416–421. ISSN: 1749-4893. DOI: 10.1038/s41566-023-01166-4. URL: https://doi.org/10.1038/s41566-023-01166-4.

[9] Yu-Ao Chen et al. 'An integrated space-to-ground quantum communication network over 4,600 kilometres'. In: *Nature* 589.7841 (2021), pp. 214–219. ISSN: 1476-4687. DOI: 10.1038/s41586-020-03093-8. URL: https://doi.org/10.1038/s41586-020-03093-8.

[10] M Sasaki et al. 'Field test of quantum key distribution in the Tokyo QKD Network'. In: *Optics Express* 19.11 (May 2011), p. 10387. ISSN: 1094-4087. DOI: 10.1364/oe.19.010387. URL: http://dx.doi.org/10.1364/OE.19.010387.

[11] D Lopez et al. ‘Madrid Quantum Communication Infrastructure: a testbed for assessing QKD technologies into real production networks’. In: *2021 Optical Fiber Communications Conference and Exhibition (OFC)*. 2021, pp. 1–4.

[12] Vadim Makarov * and Dag R Hjelme. ‘Faked states attack on quantum cryptosystems’. In: *Journal of Modern Optics* 52.5 (2005), pp. 691–705. DOI: 10.1080/09500340410001730986. URL: https://doi.org/10.1080/09500340410001730986.

[13] Artem Vakhitov, Vadim Makarov and Dag R Hjelme. ‘Large pulse attack as a method of conventional optical eavesdropping in quantum cryptography’. In: *Journal of Modern Optics* 48.13 (Nov. 2001), pp. 2023–2038. ISSN: 0950-0340. DOI: 10.1080/09500340108240904. URL: https://www.tandfonline.com/doi/abs/10.1080/09500340108240904.

[14] Paulo Vinicius Pereira Pinheiro et al. ‘Eavesdropping and countermeasures for backflash side channel in quantum cryptography’. In: *Opt. Express* 26.16 (Aug. 2018), pp. 21020–21032. DOI: 10.1364/OE.26.021020. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-26-16-21020.

[15] Jing-Zheng Huang et al. ‘Quantum hacking of a continuous-variable quantum-key-distribution system using a wavelength attack’. In: *Phys. Rev. A* 87.6 (June 2013), p. 62329. DOI: 10.1103/PhysRevA.87.062329. URL: https://link.aps.org/doi/10.1103/PhysRevA.87.062329.

[16] Adomas Baliuka et al. ‘Deep-learning-based radio-frequency side-channel attack on quantum key distribution’. In: *Physical Review Applied* 20.5 (Nov. 2023), p. 54040. DOI: 10.1103/PhysRevApplied.20.054040. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.20.054040.

[17] Kadir Durak, Naser C Jam and Saeid Karamzadeh. ‘Attack to Quantum Cryptosystems Through RF Fingerprints From Photon Detectors’. In: *IEEE Journal of Selected Topics in Quantum Electronics* 28.2: Optical Detectors (2022), pp. 1–7. DOI: 10.1109/JSTQE.2021.3089638.

[18] Alfred Menezes, Paul van Oorschot and Scott Vanstone. *Handbook of Applied Cryptography*. 1st ed. CRC Press, Oct. 1996.

[19] Auguste Kerckhoffs. ‘La cryptographie militaire’. In: *Journal des sciences militaires* IX (Jan. 1883), pp. 5–38.

[20] Hans Delfs and Helmut Knebl. *Introduction to Cryptography*. 2nd ed. Springer Berlin, 2007.

[21] Stephen Wiesner. ‘Conjugate Coding’. In: *SIGACT News* 15.1 (Jan. 1983), pp. 78–88. ISSN: 0163-5700. DOI: 10.1145/1008908.1008920. URL: https://doi.org/10.1145/1008908.1008920.

[22] Charles H Bennett and Gilles Brassard. ‘Quantum cryptography: Public key distribution and coin tossing’. In: *Theoretical Computer Science* 560 (Dec. 2014), pp. 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025. URL: http://dx.doi.org/10.1016/j.tcs.2014.05.025.

[23] Artur K Ekert. ‘Quantum cryptography based on Bell’s theorem’. In: *Physical Review Letters* 67.6 (Aug. 1991), pp. 661–663. DOI: 10.1103/PhysRevLett.67.661. URL: https://link.aps.org/doi/10.1103/PhysRevLett.67.661.

[24] Charles H Bennett. 'Quantum cryptography using any two nonorthogonal states'. In: *Physical Review Letters* 68.21 (May 1992), pp. 3121–3124. DOI: 10.1103/PhysRevLett.68.3121. URL: https://link.aps.org/doi/10.1103/PhysRevLett.68.3121.

[25] Damien Stucki et al. 'Fast and simple one-way quantum key distribution'. In: *Applied Physics Letters* 87.19 (Nov. 2005), p. 194108. ISSN: 0003-6951. DOI: 10.1063/1.2126792. URL: https://doi.org/10.1063/1.2126792.

[26] Fadri Grünenfelder et al. 'Simple and high-speed polarization-based QKD'. In: *Applied Physics Letters* 112.5 (Jan. 2018), p. 051108. ISSN: 0003-6951. DOI: 10.1063/1.5016931. URL: https://doi.org/10.1063/1.5016931.

[27] M Lucamarini et al. 'Overcoming the rate–distance limit of quantum key distribution without quantum repeaters'. In: *Nature* 557.7705 (2018), pp. 400–403. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0066-6. URL: https://doi.org/10.1038/s41586-018-0066-6.

[28] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, June 2012. ISBN: 9781107002173. DOI: 10.1017/CBO9780511976667.

[29] Andreas Klappenecker and Martin Rötteler. 'Constructions of Mutually Unbiased Bases'. In: *Finite Fields and Applications*. Ed. by Gary L Mullen, Alain Poli and Henning Stichtenoth. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 137–144. ISBN: 978-3-540-24633-6.

[30] W K Wootters and W H Zurek. 'A single quantum cannot be cloned'. In: *Nature* 299.5886 (1982), pp. 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: https://doi.org/10.1038/299802a0.

[31] Ramona Wolf. *Quantum Key Distribution*. Vol. 988. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-73990-4. DOI: 10.1007/978-3-030-73991-1.

[32] M D Eisaman et al. 'Invited Review Article: Single-photon sources and detectors'. In: *Review of Scientific Instruments* 82.7 (July 2011), p. 071101. ISSN: 0034-6748. DOI: 10.1063/1.3610677. URL: https://doi.org/10.1063/1.3610677.

[33] Sylvain Fasel et al. 'High-quality asynchronous heralded single-photon source at telecom wavelength'. In: *New Journal of Physics* 6.1 (Nov. 2004), p. 163. DOI: 10.1088/1367-2630/6/1/163. URL: https://dx.doi.org/10.1088/1367-2630/6/1/163.

[34] Tian Zhong et al. 'High-quality fiber-optic polarization entanglement distribution at 1.3 μm telecom wavelength'. In: *Opt. Lett.* 35.9 (May 2010), pp. 1392–1394. DOI: 10.1364/OL.35.001392. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-35-9-1392.

[35] Shellee D Dyer et al. 'High-efficiency, ultra low-noise all-fiber photon-pair source'. In: *Opt. Express* 16.13 (June 2008), pp. 9966–9977. DOI: 10.1364/OE.16.009966. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-16-13-9966.

[36] Markus Hijlkema et al. 'A single-photon server with just one atom'. In: *Nature Physics* 3.4 (2007), pp. 253–255. ISSN: 1745-2481. DOI: 10.1038/nphys569. URL: https://doi.org/10.1038/nphys569.

[37] Satoshi Kako et al. 'A gallium nitride single-photon source operating at 200 K'. In: *Nature Materials* 5.11 (2006), pp. 887–892. ISSN: 1476-4660. DOI: 10.1038/nmat1763. URL: https://doi.org/10.1038/nmat1763.

[38] V Fock. 'Konfigurationsraum und zweite Quantelung'. In: *Zeitschrift für Physik* 75.9 (1932), pp. 622–647. ISSN: 0044-3328. DOI: 10.1007/BF01344458. URL: https://doi.org/10.1007/BF01344458.

[39] Christopher Gerry and Peter Knight. *Introductory Quantum Optics*. Cambridge: Cambridge University Press, 2004. ISBN: 9780521527354. DOI: DOI:10.1017/CBO9780511791239. URL: https://www.cambridge.org/core/product/B9866F1F40C45936A81D03AF7617CF44.

[40] Stefano Pirandola et al. 'Fundamental limits of repeaterless quantum communications'. In: *Nature Communications* 8.1 (2017), p. 15043. ISSN: 2041-1723. DOI: 10.1038/ncomms15043. URL: https://doi.org/10.1038/ncomms15043.

[41] Chandra M Natarajan, Michael G Tanner and Robert H Hadfield. 'Superconducting nanowire single-photon detectors: physics and applications'. In: *Superconductor Science and Technology* 25.6 (2012), p. 063001. ISSN: 0953-2048. DOI: 10.1088/0953-2048/25/6/063001. URL: https://dx.doi.org/10.1088/0953-2048/25/6/063001.

[42] Jun Zhang et al. 'Advances in InGaAs/InP single-photon detector systems for quantum communication'. In: *Light: Science & Applications* 4.5 (2015), e286–e286. ISSN: 2047-7538. DOI: 10.1038/lsa.2015.59. URL: https://doi.org/10.1038/lsa.2015.59.

[43] Yu-Qiang Fang et al. 'InGaAs/InP single-photon detectors with 60% detection efficiency at 1550 nm'. In: (July 2020). DOI: 10.1063/5.0014123. URL: http://arxiv.org/abs/2007.06792%20http://dx.doi.org/10.1063/5.0014123.

[44] C H Bennett and G Brassard. 'Experimental Quantum Cryptography: The Dawn of a New Era for Quantum Cryptography: The Experimental Prototype is Working]'. In: *SIGACT News* 20.4 (Nov. 1989), pp. 78–80. ISSN: 0163-5700. DOI: 10.1145/74074.74087. URL: https://doi.org/10.1145/74074.74087.

[45] Christopher Portmann and Renato Renner. 'Security in quantum cryptography'. In: *Reviews of Modern Physics* 94.2 (June 2022), p. 25008. DOI: 10.1103/RevModPhys.94.025008. URL: https://link.aps.org/doi/10.1103/RevModPhys.94.025008.

[46] C E Shannon. 'A mathematical theory of communication'. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[47] Christopher A Fuchs et al. 'Optimal eavesdropping in quantum cryptography. I. Information bound and optimal strategy'. In: *Physical Review A* 56.2 (Aug. 1997), pp. 1163–1172. DOI: 10.1103/PhysRevA.56.1163. URL: https://link.aps.org/doi/10.1103/PhysRevA.56.1163.

[48] Dagmar Bruß. 'Optimal Eavesdropping in Quantum Cryptography with Six States'. In: *Physical Review Letters* 81.14 (Oct. 1998), pp. 3018–3021. DOI: 10.1103/PhysRevLett.81.3018. URL: https://link.aps.org/doi/10.1103/PhysRevLett.81.3018.

[49] S Pirandola et al. ‘Advances in quantum cryptography’. In: *Advances in Optics and Photonics* 12.4 (2020), pp. 1012–1236. DOI: 10.1364/AOP.361502. URL: https://opg.optica.org/aop/abstract.cfm?URI=aop-12-4-1012.

[50] Won-Young Hwang. ‘Quantum Key Distribution with High Loss: Toward Global Secure Communication’. In: *Physical Review Letters* 91.5 (Aug. 2003), p. 57901. DOI: 10.1103/PhysRevLett.91.057901. URL: https://link.aps.org/doi/10.1103/PhysRevLett.91.057901.

[51] Anqi Huang et al. ‘Quantum key distribution with distinguishable decoy states’. In: *Physical Review A* 98.1 (July 2018), p. 12330. DOI: 10.1103/PhysRevA.98.012330. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.012330.

[52] Vadim Makarov, Andrey Anisimov and Johannes Skaar. ‘Effects of detector efficiency mismatch on security of quantum cryptosystems’. In: *Physical Review A* 74.2 (Aug. 2006), p. 22313. DOI: 10.1103/PhysRevA.74.022313. URL: https://link.aps.org/doi/10.1103/PhysRevA.74.022313.

[53] Roger Newman. ‘Visible Light from a Silicon pn Junction’. In: *Physical Review* 100.2 (Oct. 1955), pp. 700–703. DOI: 10.1103/PhysRev.100.700. URL: https://link.aps.org/doi/10.1103/PhysRev.100.700.

[54] Christian Kurtsiefer et al. ‘The breakdown flash of silicon avalanche photodiodes-back door for eavesdropper attacks?’ In: *Journal of Modern Optics* 48.13 (Nov. 2001), pp. 2039–2047. ISSN: 0950-0340. DOI: 10.1080/09500340108240905. URL: https://doi.org/10.1080/09500340108240905.

[55] Víctor Zapatero et al. ‘Advances in device-independent quantum key distribution’. In: *npj Quantum Information* 9.1 (2023), p. 10. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00684-x. URL: https://doi.org/10.1038/s41534-023-00684-x.

[56] Paul Kocher, Joshua Jaffe and Benjamin Jun. ‘Differential Power Analysis’. In: *Advances in Cryptology — CRYPTO’ 99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. ISBN: 978-3-540-48405-9.

[57] Andrea Stanco et al. ‘Versatile and concurrent FPGA-based architecture for practical quantum communication systems’. In: (July 2021). DOI: 10.1109/TQE.2022.3143997. URL: http://arxiv.org/abs/2107.01857%20http://dx.doi.org/10.1109/TQE.2022.3143997.

[58] G L Roberts et al. ‘Patterning-effect mitigating intensity modulator for secure decoy-state quantum key distribution’. In: *Optics Letters* 43.20 (2018), pp. 5110–5113. DOI: 10.1364/OL.43.005110. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-43-20-5110.

[59] Costantino Agnesi et al. ‘All-fiber self-compensating polarization encoder for quantum key distribution’. In: *Optics Letters* 44.10 (2019), pp. 2398–2401. DOI: 10.1364/OL.44.002398. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-44-10-2398.

[60] Marco Avesani et al. ‘Deployment-Ready Quantum Key Distribution over a Classical Network Infrastructure in Padua’. In: *Journal of Lightwave Technology* 40.6 (Mar. 2022), pp. 1658–1663. ISSN: 15582213. DOI: 10.1109/JLT.2021.3130447.

[61] Chi-Hang Fred Fung and Hoi-Kwong Lo. 'Security proof of a three-state quantum-key-distribution protocol without rotational symmetry'. In: *Physical Review A* 74.4 (Oct. 2006), p. 42342. DOI: 10.1103/PhysRevA.74.042342. URL: https://link.aps.org/doi/10.1103/PhysRevA.74.042342.

[62] Y Ren et al. 'Key recovery against 3DES in CPU smart card based on improved correlation power analysis'. In: *Tsinghua Science and Technology* 21.2 (2016), pp. 210–220. ISSN: 1007-0214. DOI: 10.1109/TST.2016.7442503.

[63] C O'Flynn and Z David Chen. 'Side channel power analysis of an AES-256 bootloader'. In: *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2015, pp. 750–755. ISBN: 0840-7789. DOI: 10.1109/CCECE.2015.7129369.

[64] Avnet. *ZedBoard Hardware User Guide*. 2014. URL: https://www.avnet.com/wps/wcm/connect/onesite/922900e3-3d57-4cc7-883f-a8b9fbea0cd0/ZedBoard_HW_UG_v2_2.pdf?MOD=AJPERES&CACHEID=ROOTWORKSPACE.Z18_NA5A1I41L0ICD0ABNDMDDG0000-922900e3-3d57-4cc7-883f-a8b9fbea0cd0-nxyWMFS.

[65] Keysight Technologies. *N2790A 100 MHz, N2791A 25 MHz and N2891A 70 MHz High-Voltage Differential Probes*. Sept. 2021.

[66] 'IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters'. In: *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)* (2011), pp. 1–139. DOI: 10.1109/IEEESTD.2011.5692956.

[67] Andrew Schaefer. *The Effective Number of Bits (ENOB) of my R&S Digital Oscilloscope*. Tech. rep. 2011.