# MASTER'S THESIS

## POST-QUANTUM CRYPTHOGRAPHY FOR HEALTH DATA

ausgeführt am

CAMPUS GRAZ 02

FACHHOCHSCHULE DER WIRTSCHAFT

Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: David Pöttler

Personenkennzeichen: 1810320011

Graz, am 10. Juli 2020

.............................................................

Unterschrift

# EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.............................................................

Unterschrift

# DANKSAGUNG

Ich möchte meiner Familie danken, welche mich in den zeitweise schwierigen Phasen während dieser Arbeit immer wieder aufgebaut hat oder manchmal auch einfach nur ein offenes Ohr für mich hatte. Ein besonderes Danke an meine Schwester welche diese Arbeit Korrektur las und auch einige stilistische Schnitzer verbesserte. Ebenfalls möchte ich meinem Arbeitgeber, der Landwirtschaftskammer Steiermark danken. Ohne die Flexibilität, die mir ermöglicht wurde, wäre die Erstellung der Arbeit um einiges schwieriger geworden. Abschließend möchte ich mich noch bei Herrn FH-Hon.Prof. Ing. Dipl.-Ing. (FH) Dipl.-Ing. Dr. techn. Michael Georg Grasser MBA MPA CMC für die Betreuung dieser Masterarbeit und seine fachliche und organisatorische Unterstützung sowie für die schnelle Reaktionszeit bei Anfragen bedanken.

# KURZFASSUNG

Das Ziel dieser Masterarbeit ist es das Thema post-quantum Kryptografie zu beleuchten. Hierfür wird einleitend das Thema Quantencomputer allgemein erläutert. Dies soll zeigen, wie Quantencomputer im Vergleich zu traditionellen Computern arbeiten. Dies beinhaltet welche Realisierungsformen es gibt, welche Potentiale diese besitzen aber auch welchen Einfluss dies auf die Informationssicherheit hat. Klassische, häufig verwendete Algorithmen der Gegenwart werden hierfür genauer beschrieben und wie anfällig das diese gegen Quantencomputer sind. Doch es gibt alternativen die langsam an Bedeutung gewinnen. Das Feld der post-quanten Kryptografie beschäftigt sich mit neuen Verfahren welche gegen herkömmliche Computer als auch gegen Quantencomputer sicher sind. In dieser Arbeit wird dieses noch recht junge Thema mithilfe von Experten erarbeitet und soll ein Bild er derzeitigen Lage liefern, und aufzeigen wo Handlungsbedarf besteht. Die erzielten Ergebnisse werden abschließend auf ein ausgewähltes System im Gesundheitswesen angewendet und soll zeigen wie gut dieses gegen solche Angriffe aufgestellt wäre.

# ABSTRACT

This master's thesis is about post-quantum cryptography. Therefore, it shows how quantum computers differ from traditional computers, what kind of quantum computers do exist, and what strengths they have. But the focus lies on the threat to information security. For this purpose, the paper shows how traditional cryptographic algorithms work and how vulnerable they are against quantum computers. The paper also shows the possible alternatives. This field is called post-quantum cryptography, and these algorithms are resistant against traditional computers and also against quantum computers. In this paper, expert interviews illustrated an outline of the threat and where it is mandatory to react. At least the results are applied to an IT system in the health sector, and it is analyzed how vulnerable it would be against quantum computers.

# CONTENTS

# 1 INTRODUCTION

*"If computers that you build are quantum,*  
*then spies of all factions will want 'em.*  
*Our codes will all fail,*  
*and they'll read our email,*  
*till we've crypto that's quantum, and daunt 'em. (Shor & Shor)"*

In 2013 the surveillance activities of the NSA and other intelligence agencies were made public. They now have access to the infrastructure of internet carriers and also to servers from Microsoft, Google and many others. (Clement)

The usage of HTTPS in the Web has increased significantly. While in 2014 the amount of HTTPS traffic was between 30 and 40 percent, now in 2019 it has increased to 70 to 90 percent, depending on the platform. (Google, 2019)

Today's cryptosystems are quite developed for the computing environment of today. Quantum computers, however, are on the rise and in 2019, Google achieved quantum supremacy. This means they conducted a computation on a quantum computer in a few minutes while a classic computer would need thousands of years. But it is not a threat to current cryptosystems currently. (Artemenko, 2019) Some of today's most used cryptosystems like RSA, Diffie-Hellman or elliptic curve cryptography are broken by a quantum algorithm. Hence, RSA can be broken by the Shor Algorithm in $\ln n^2$ operations. Because of quantum computers, most of today's user asymmetric cryptosystems will become breakable. (Bernstein, Buchmann, & Dahmén, 2009) According to Mosca (2018), RSA-2048 could be broken with an 50 percent chance of success by the year 2031. Organizations like the NSA have started to migrate to quantum-safe algorithms.

## 1.1 Research question, objectives, methodology

The primary goal in this paper is to answer the following research question: How to secure encrypted connections and data again quantum computers?

To answer this question, a set of methods were used. A fundamental part was the literature study to get an insight into how quantum computers work and answer the questions:

- How does today's cryptography work?
- What impact can quantum computers have?
- What can be done against quantum computers in cryptography?

Based on this current state of knowledge, a set of recommendations were developed on the basis of this literature research. In detail, these are based on multiple theses defined in chapter six and are checked with expert interviews.

The aimed results are a set of recommendations to increase the information security level against quantum computers. These recommendations are grouped into short, intermediate and long term recommendations. At last, the results are applied in a case example which analyzes a healthcare system and shows how to improve it’s security level.

## 1.2 **Structure**

The second chapter processes the topic of quantum computers in general. It covers the basic principles of their functionality. This leads to the physical kind of implementations and handles the way quantum computers can be built. The last two topics cover the fundamental algorithms related to cryptography and the actual size of current quantum computers.

An overview of classic cryptography is given in chapter three. It covers the goals of cryptography as well as the extensive sub-areas of cryptography. These are the hash functions, symmetric cryptography and asymmetric or public-key cryptography.

Chapter four shows the impact of quantum computers, including the possibilities of chances for improvements in various problems nowadays. Additionally, it highlights the weaknesses in current cryptography. The threats for each sub-area and their gravitas are showcased.

The possible new approaches for cryptography in a post-quantum world are described in chapter five. Beginning with the challenges post-quantum cryptography has to face as well as the difference to quantum cryptography. It gives an overview of new approaches and the basic principle of how they function. Lastly, the chapter covers several post-quantum algorithms.

Chapter six covers the main part as it answers the research question. Hypotheses are derived from the theoretical part of the paper. These hypotheses are checked with expert interviews. The answer of the research question is based on the results.

In the last two chapters, the results are theoretically applied to a case example, and a short conclusion is given.

# 2 QUANTUM COMPUTERS

This chapter covers the functions of quantum computers as well as the functions of classic computers. It handles the differences, benefits, and disadvantages of each technique. The section concerning quantum computers will handle the potentials in detail, and which types of computers exist.

## 2.1 Classic Computers

Before starting with quantum computers, the focus will lie on the fundamental operations of a classic computer. This paper will only cover the Boolean algebra and the resulting gates. This is the basis for every classic computer and consist of the values zero and one. The difference of how they work is explained in section 2.2.

Integrated circuits in a computer only have two values, zero and one. A zero is most represented by a voltage between zero and one volt and a one by a voltage between two and five volts. With gates, it is possible to perform calculations with these values. These gates are made out of transistors, which work as a fast switch. The transistor has three connections a collector, a basis, and an emitter. If in $V_{in}$ is below a defined threshold of voltage then the transistor is off, and $V_{out}$ has a value near $V_{cc,}$ which is typically fife volt. When $V_{in}$ oversteps this threshold the transistor starts to work, and $V_{out}$ goes to ground which will be interpreted as a zero. If $V_{in}$ is low, then $V_{out}$ is high and vice versa, this is an inverter and is displayed in Figure 2-1 (a)

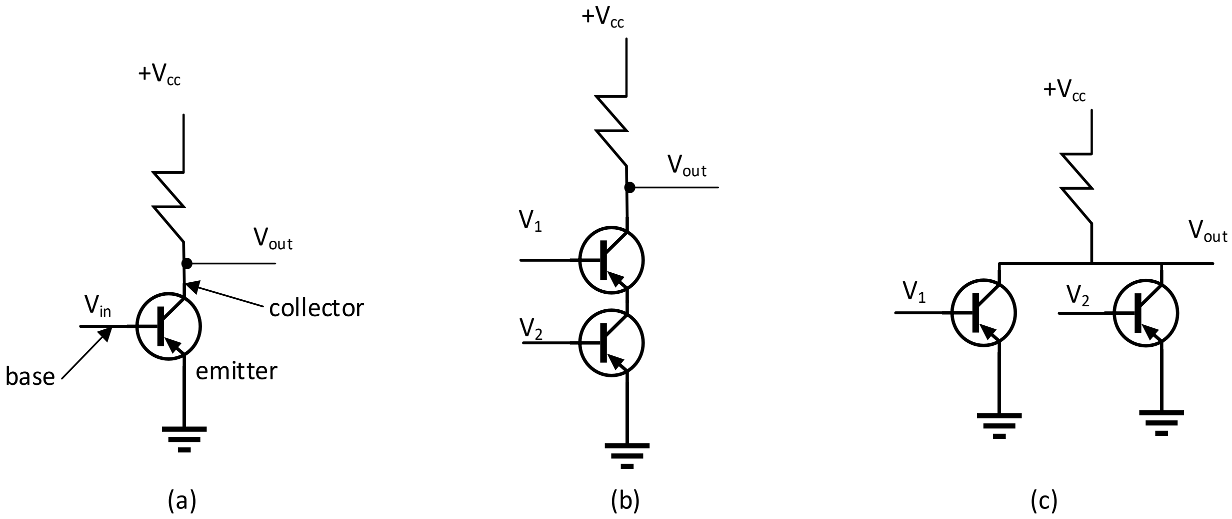
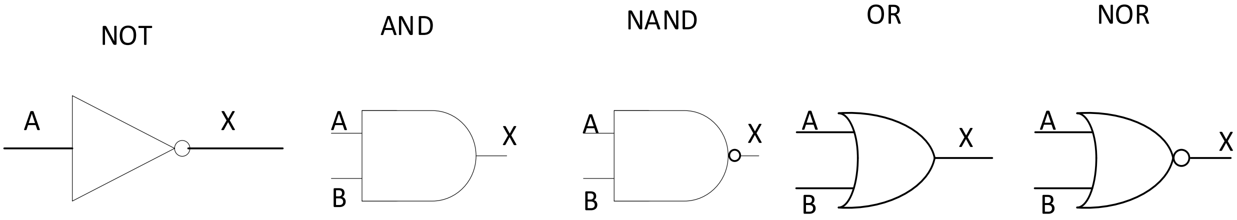


*Figure 2-1: (a) Transistor-inverter; (b) NAND-Gate; (c) NOR-Gate (cf. Tanenbaum & Goodman, 2004)*

Figure 2-1 (b) represents a NOT And (NAND) gate. It works as a serial connection. If none or one of the two inputs is low then the output is high. Only if both inputs are high, the outputs switch to low. Figure 2-1 (c) is a parallel connection. This is called a NOT OR (NOR) gate. For this gate, it is enough if only one of the two inputs is high to switch the output to a low. These are the three most simple gates. It is possible to build any other gate out of these three. There are also the

AND and OR gate, which are inverted versions of the NAND and NOR gates. (Tanenbaum & Goodman, 2004) In Figure 2-2 all five gates are represented with their truth tables.



| A | X |
|---|---|
| 0 | 1 |
| 1 | 0 |

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

*Figure 2-2: Symbols and truth tables of the fife basic gates (cf. Tanenbaum & Goodman, 2004)*

## 2.2 **Quantum Computers**

In a classical computer, bits are represented with different levels of voltage. They can even handle some disruption in these signals and get correct values. They can be either zero or one. In quantum computers, information is represented by quantum bits (qubit). A quantum bit can be a one, a zero or both at the same time. This behavior is called superposition. At every single moment, a qubit can represent all possible values at the same time. Unlike classic computers, noise is a big problem for quantum computers. It can falsify the computation and may affect the result. (National Academies of Sciences, 2019) The following sections will explain the principals of quantum computers more detailed as well as how computations are executed and which types of quantum computers exist. Lastly, it covers which potentials these computers have and if they can replace a classic computer.

### 2.2.1 Fundamentals

One of the fundamentals of quantum mechanics is explained in the thought experiment of Schrodinger's cat; a cat is sitting in a box. Then, something happens and with a probability of 50 percent, the cat could now be dead or alive. As long you don't actually look into the box the cat is dead and alive. This is called superposition in the world of quantums. It's only possible to determine the status of the quantum when you measure it. This measurement destroys the superposition. In Schrodinger's cat, you open the box to see if the cat is alive or dead. The quantum has an angle of $\alpha$ in one and an angle of $\beta$ in the other direction at the same time. Depending on $\alpha$ and $\beta$ the quantum goes in one of the both directions with a certain probability $p$. In quantum computers, this is the process where a qubit gets to a one or zero. (Homeister, 2018)

A single qubit's state is represented by

$$|\psi\rangle \; = \; a|0\rangle \; + \; b|1\rangle$$

with $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. That means that a single qubit will be measured with a given probability of a zero or a one. a and b are complex numbers and the square of the modulus needs to be one. With these boundaries a gets $e^{i\alpha}\left(\cos\frac{\theta}{2}\right)$ and b gets $e^{i(\alpha+\varphi)}\left(\sin\frac{\theta}{2}\right)$. The numbers φ and θ defining a point in an three-dimensional room for the state of a qubit, which can be illustrated in a Bloch sphere. (National Academies of Sciences, 2019) Such a position of a qubit is represented in Figure 2-3.

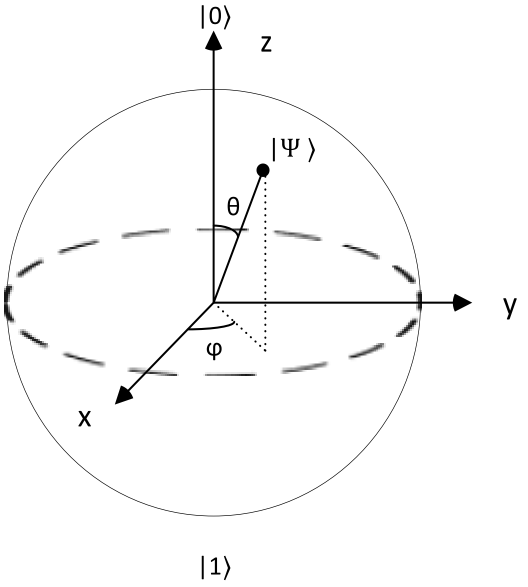


*Figure 2-3: Representation of a qubit in a Bloch sphere. (cf. Nielsen & Chuang, 2010)*

> *It turns out that the global phase α has no physical significance whatsoever, and a single-qubit state can be fully described by two real numbers $0 \leq \theta < \pi$ and $0 \leq \varphi < 2\pi$. (National Academies of Sciences, 2019, p. 44)*

The state of a qubit can be modeled in a Hilbert space. Because there are only the states zero and one the Hilbert space has only two dimensions.

$$\mathrm{H} = \mathbb{C} \oplus \mathbb{C}$$

If we don't have only a single qubit, wen needs n registers so the Hilbert space needs to expand.

$$\mathrm{H}_n = \mathrm{H}_1 \oplus \mathrm{H}_2 \oplus \ldots \oplus \mathrm{H}_n$$

The result of a computation which can be measured is:

$$v \; = \sum_{I \in I_n} \alpha_I |I\rangle$$

For $I_2$ the result vectors can be

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

and any of them can be measured with a certain probability of

$$P_v(I) = |\alpha_I|^2 / \sum_{J \in I_n} |\alpha_J|^2 .$$

The amplitude of the desired vector should be large compared to the other vectors. If it is not possible to reduce the other vectors to zero, then there is a certain amount of uncertainty in the result. When we look at the field of cryptography, this uncertainness is not a real problem because we can check if the calculation was correct or not. (Bernstein et al., 2009) This composing of qubits also works for calculations. For this, we use tensor products. They are the products of two vector rooms. This is represented by

$$V_1 \otimes V_2 \, ,$$

as an (m*n) room with a basis of $e_0 \ldots e_{n-1}$ for $V_1$ and $f_0 \ldots f_{n-1}$ for $V_2$ In this room the tensor product is defined as

$$V_1 \otimes V_2 = \left( \sum_{i=0}^{m-1} \alpha_i e_i \right) \otimes \left( \sum_{j=0}^{n-1} \beta_j f_j \right).$$

As an example, in a matrix with $n = m = 2$ the tensor product looks like the following:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}.$$

A m bit long register can be described by the m times product of the single bit. (Homeister, 2018)

### 2.2.2 Unitary transformations

Any transformation applied on a qubit has to be unitary. This is achieved by a unitary matrix. This unitary transformation is reversible. For n input qubits a $2^n \times 2^n$ matrix is needed for n output qubits. A matrix is unitary if $A^{-1} = A^T$. The inverse matrix of A is called $A^{-1}$ . This is true if $A \times A^{-1} = I$. I is the identity matrix and is for a 2 x 2 Matrix defined as

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} .$$

$A^T$ is defined as

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} .$$

Furthermore, the unitary transformation has the following properties:

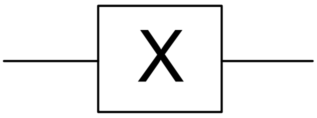
1. Every allowed state of a qubit must be in an allowed state after the transformation. Transformations preserve their length.
2. Unitary transformations do not change the scalar product of two vectors. They preserve their angle.
3. Unitary transformations must be reversible.

4. Unitary transformations are linear because they can be described by matrices. (Homeister, 2018)

### 2.2.3 Gates

In this section, the paper will cover the most important quantum gates which you can see as a quantum equivalent of the classic gates described in section 2.1.

To start with the one qubit gates, the simplest quantum gate is the Pauli-X-Gate. It acts like a NOT gate. Figure 2-4 shows the symbol of the gate and also the matrix.



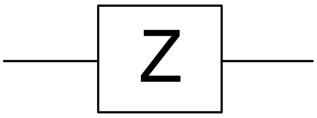
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

*Figure 2-4: Pauli-X-Gate (cf.Scherer, 2016)*

This gate negates the input value. With a given input vector of α + β, the NOT gate acts like below.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

The next gate is the Pauli-Z-Gate. It turns the $|1\rangle$ to -$|1\rangle$ and leave the $|0\rangle$ as it is, this is shown in Figure 2-1.



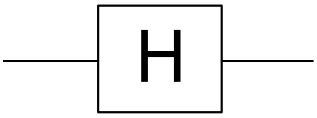
$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

*Figure 2-5: Pauli-Z-Gate (cf.Scherer, 2016)*

This will flip the direction of β. When you look at Figure 2-3, this would mean the vector would turn 180 degrees around the Z-axis, this is described as

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} * \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}.$$

One of the most used gates is the Hadamard-Gate shown in Figure 2-6.



$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

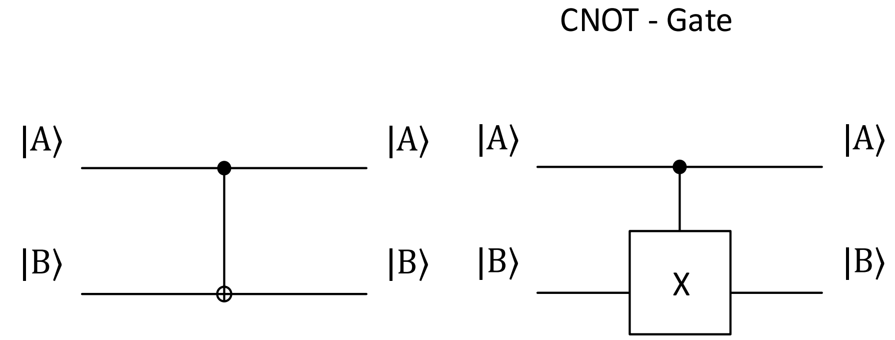
*Figure 2-6: Hadamard gate (cf.Scherer, 2016)*

The input $|0\rangle$ turns into $(|0\rangle + \ |1\rangle)/\sqrt{2}$ and the $|1\rangle$ transforms to $(|0\rangle - \ |1\rangle)/\sqrt{2}$. It will rotate the vector 90 degrees around the y axis and 180 degrees around the x-axis. (Nielsen & Chuang, 2010; Scherer, 2016) This gate is used for the entanglement of qubits as you will see later.

There are also gates for multiple qubits. The most used is the Controlled Not (CNOT) gate. It has a control bit and a target bit. If the control qubit is zero then the target bit won't be changed. But is the control bit a one the target qubit will be flipped. This means the following:

$$|00\rangle \rightarrow |00\rangle;\ |01\rangle \rightarrow |01\rangle;\ |10\rangle \rightarrow |11\rangle;\ |11\rangle \rightarrow |10\rangle$$

It acts as a exclusive or (XOR) gate. It represents an addition of the two input vectors module two. There are two common symbols as shown in Figure 2-1

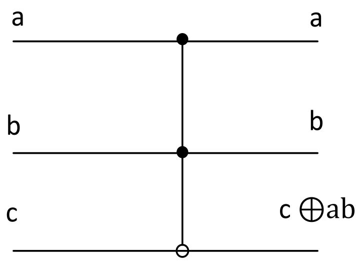


$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

*Figure 2-7: CNOT gate with both icons calculation matrix (cf. Scherer, 2016)*

From this gate and the single-qubit gates the following and any other multi-qubit gate can be constructed out of these. (Nielsen & Chuang, 2010)

It is possible to simulate classic gates in quantum computers. Classic gates are irreversible but it is possible to build classic circuits only from reversible quantum gates. A special gate, called Toffoli gate, is used for this process. The Toffoli gate can act as a NAND gate. From the NAND gate, it is possible to build any other gate and circuit. It has three inputs. Two of them are control bits which will not be affected by the operation. The third one is a data bit. Only if both of the control bits are one, the data bit will flip its state. The representation and truth table for the Toffoli gate is illustrated in Figure 2-8.



| Inputs | | | Outputs | | |
|---|---|---|---|---|---|
| a | b | c | a' | b' | c' |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

*Figure 2-8: Toffoli gate representation and truth table (cf. Nielsen & Chuang, 2010)*

This gate is responsible for a quantum computer to perform deterministic operations such as a classic computer. In non-deterministic operations the quantum computer faces no problems whatsoever. With a simple Hadamard gate and an input vector of one, it produces random numbers with a 50/50 probability of one and zero. With this classic computations, the quantum computer has no advantage against a classic PC. (Nielsen & Chuang, 2010) In section 2.3, quantum algorithms which can use the advantage of quantum effects will be discussed.

### 2.2.4 Entanglement

As described before, in a quantum computer a bit can have the state zero and one at the same time. This behavior is called entanglement. Entanglement is the reason why a quantum computer is superior to a classic computer in various calculations. For the example of a two-qubit system the possibilities are

$$\Psi = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle\,.$$

An example for an entangled state is the Einstein-Podolski-Rosen state

$$|\Psi_{EPR}\rangle = \frac{(|01\rangle + |10\rangle)}{\sqrt{2}}\quad.$$

This state is called entangled when it is not possible to create from the product of two individual qubits. Such a state can be achieved by applying several simple transformations to a two-qubit vector. The already described Hadamard transformation and a CNOT transformation is needed for this. First, the Hadamard Transformation is applied to the first qubit and the CNOT transformation between the first and the second qubit. This can be written as

$$|01\rangle \rightarrow (|0\rangle + |1\rangle)|1\rangle \rightarrow |01\rangle + |10\rangle\,.$$

The benefits of entanglement can be simply illustrated by Deutsch's Problem in section 2.4.3. (Vedral & Plenio, 1998)

### 2.2.5 Error Correction

Errors in data is a problem in all systems that process data as it corrupts the result and can lead to false decisions. In a classic two-bit system, there is only one kind of error to detect, namely the bit flip. In a two-qubit quantum computer, there are four possible kinds of errors represented by the Pauli operators

$$\mathbb{I} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \sigma_y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}, \sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}.$$

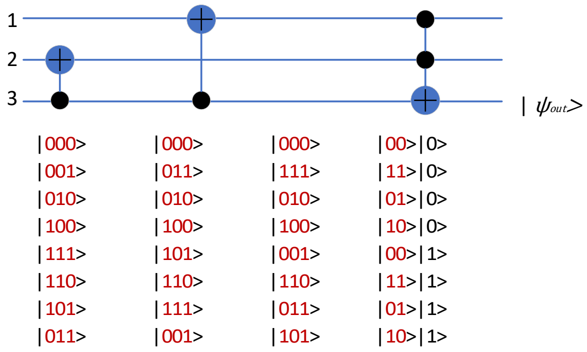
The probability for each of these errors is 25 percent.

$$|\psi\rangle 12 \rightarrow \begin{cases}\mathbb{I}|\psi\rangle 12 & Prob\ .25\\ \sigma_x|\psi\rangle 12 & Prob\ .25\\ \sigma_y|\psi\rangle 12 & Prob\ .25\\ \sigma_z|\psi\rangle 12 & Prob\ .25\end{cases}$$

There are various possible types of error detection and correction model for quantum computers. This paper will only handle the most basic one, which is the quantum repetition code. For these two, additional bits are implemented for each qubit. For a qubit in a superposition, it looks like

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle\,.$$

A network to decode a qubit with repetition code is illustrated in Figure 2-9

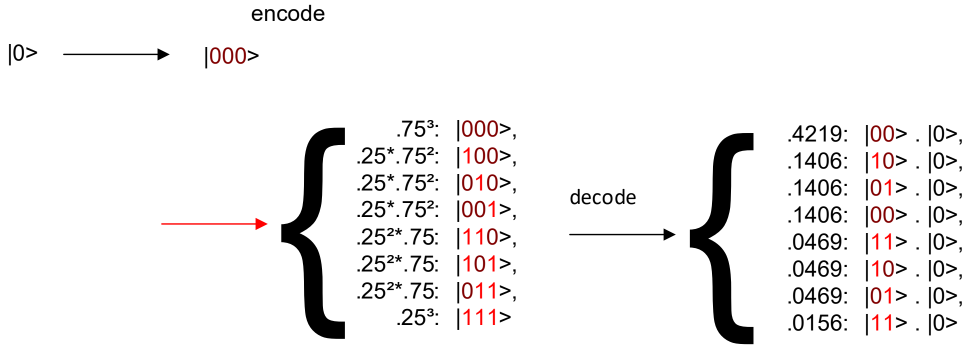


| | | | |
|---|---|---|---|
| \|000> | \|000> | \|000> | \|00>\|0> |
| \|001> | \|011> | \|111> | \|11>\|0> |
| \|010> | \|010> | \|010> | \|01>\|0> |
| \|100> | \|100> | \|100> | \|10>\|0> |
| \|111> | \|101> | \|001> | \|00>\|1> |
| \|110> | \|110> | \|110> | \|11>\|1> |
| \|101> | \|111> | \|011> | \|01>\|1> |
| \|011> | \|001> | \|101> | \|10>\|1> |

Quantum network for decoding the output qubit 3. The steps show the effect of the network to the basis states. The decoded qubit is shown in the last column.

*Figure 2-9: Repetition code network for one qubit and the corresponding transformations. (cf. Knill et al., 2002)*

This encoding will detect and correct an error if only one of the three bits is changed. The majority of bits represent the interpreted value of the qubit. The error probability for a change of a single bit is 25 percent. Figure 2-10 shows the probabilities for every possible state that can occur in this encoding. There are four correct interpreted states and four false interpreted states. The total probability of the false states is about 16 percent which is an improvement compared to the 25 percent probability of any kind of error without error correction. (Knill et al., 2002)



| | | | |
|---|---|---|---|
| $.75^3$: | \|000>, | .4219: | \|00> . \|0>, |
| $.25^*.75^2$: | \|100>, | .1406: | \|10> . \|0>, |
| $.25^*.75^2$: | \|010>, | .1406: | \|01> . \|0>, |
| $.25^*.75^2$: | \|001>, | .1406: | \|00> . \|0>, |
| $.25^{2*}.75$: | \|110>, | .0469: | \|11> . \|0>, |
| $.25^{2*}.75$: | \|101>, | .0469: | \|10> . \|0>, |
| $.25^{2*}.75$: | \|011>, | .0469: | \|01> . \|0>, |
| $.25^3$: | \|111> | .0156: | \|11> . \|0> |

*Figure 2-10: Error probability for an encoded zero. (cf. Knill et al., 2002)*

## 2.3 Physical realization

There are a few basic requirements for a potential physical realization of a quantum computer must match. These are:

*1. Robustly represent quantum information*

*2. Perform a universal family of unitary transformations*

*3. Prepare a fiducial initial state*

*4. Measure the output result (Nielsen & Chuang, 2010, p. 279).*

The first and the second requirement are the most difficult to achieve. For the first one, the state of a qubit has to develop unitary. Because of the decoherence, it possible for the qubit to interact with its environment. If this happens, information gets lost. But it is not possible to perfectly isolate the qubit from its environment. There is a short time until this interaction happens. In this time the computations and measurements must be complete. The second means that it must be possible to use two-qubit gates with any two qubits. For the third point, it must be possible to create a constant stating state. The fourth point states that measurements must be able with every subset of bits. (Homeister, 2018)

According to Nielsen (2010), Table 2-1 shows the decoherence time of some physical implementations as the time needed for an operation and the maximum operation until the coherence time is reached.

| **System** | **$T_Q$** | **$T_{op}$** | **$n_{op}$** |
|---|---|---|---|
| **Nuclear spin** | $10^{-2} – 10^{8}$ | $10^{-3} – 10^{-6}$ | $10^{5}$ - $10^{14}$ |
| **Electron spin** | $10^{-3}$ | $10^{-7}$ | $10^{4}$ |
| **Ion trap** | $10^{-1}$ | $10^{-14}$ | $10^{13}$ |
| **Electron – Au** | $10^{-8}$ | $10^{-14}$ | $10^{6}$ |
| **Electron – GaAs** | $10^{-10}$ | $10^{-13}$ | $10^{3}$ |
| **Quantum dot** | $10^{-6}$ | $10^{-9}$ | $10^{3}$ |
| **Optical cavity** | $10^{-5}$ | $10^{-14}$ | $10^{9}$ |
| **Microwave cavity** | $10^{0}$ | $10^{-4}$ | $10^{4}$ |

*Table 2-1: Decoherence time TQ, operation times Top and maximum operations nop for some physical implementations of quantum bits (Nielsen, 2010)*

The following sections will cover several of the physical implementations of table 2-1. These explanations will only cover the fundamental way of function and will not be a detailed workup.

### 2.3.1 Photons

Optical photons fit quite well as a physical implementation of a quibt. Because they are chargeless, they do not interact strongly with each other or even with most materials, they can be transported over a long distance using optical fibers and can be manipulated with simple tools like a phase shifter, a mirror or a beam splitter. Mirrors are used to change the propagation of the photon. A phase shifter can be considered as a transparent plate with different refraction than the free space around. A beam splitter generally is a construction of two prisms which will only reflect a fraction of the input. (Nielsen & Chuang, 2010) The two qubit states are represented by two different polarisations of the photons. Theses are up-down and left-right. The source of the photons oftentimes is a laser. (National Academies of Sciences, 2019)

There is, however, a problem with photon based qubits when it comes to realizing the universal CNOT gates. This problem is illustrated in Figure 2-11.

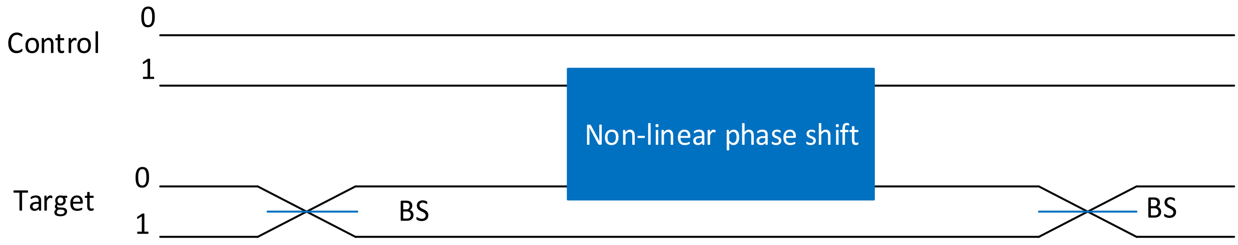


*Figure 2-11: Possible realization of an optical CNOT gate. (cf. O'Brien, 2007)*

The two beam splitters perform a Hadamard Operation. The second beam splitter will undo the first one if the phase shift is not applied. As for CNOT, this phase shift must only be done if the value is one. There is no known material which would offer the required nonlinearity to perform this operation. In 2001, it was possible to do that with single-photon sources an detectors. However, there is still a lot of improvement necessary for efficient, scalable devices. (O'Brien, 2007)

### 2.3.2 Ion trap

The two states of a qubit can be reached with the two internal states of an ion. The ground state represents a zero and the excited state a one. In an n bit register each ion can be manipulated seperatly with a different laser beam. Therefore, the CNOT gate between the ions in the trap can be implemented by exciting them with a laser. The decoherence time is exceptionally long compared with many other implementations. The readout of the computation can be done by using quantum jumps with high efficiency. The ions must be cooled near the absolute zero, which is done with lasers. (Cirac & Zoller, 1995) The iron trap electromagnetic field holds the cooled ions in position and needs to be in a vacuum. By exiting the ion it starts to move. This can be perceived as a kind of data-bus out of exited and ground ions. (Homeister, 2018) This process is illustrated in Figure 2-12.

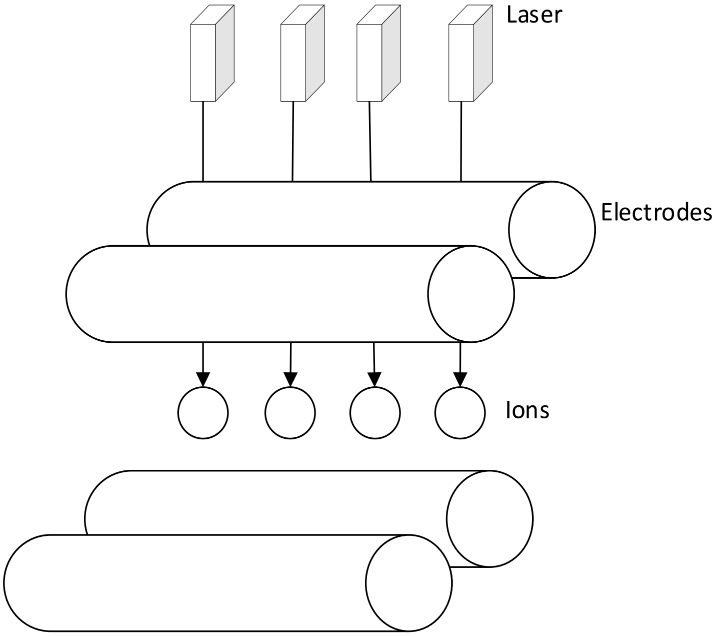


*Figure 2-12: Ions In an ion trap, read and transformed by lasers. (cf. Homeister, 2018)*

### 2.3.3 Nuclear magnetic resonance

Nuclear magnetic resonance (NMR) is based on the spin of atomic nuclei. The nuclei spin acts as a dipole which is aligned by a magnetic field. The atomic nuclei must have an odd number of protons or neutrons. Nmerous atoms meet this requirement such as Carbon or Hydrogen. On the example of Hydrogen, the nuclei can have multiple states. The dipole can be oriented in the same direction as the magnetic field. In this case, the nuclei are in a lower energy state. If the dipole oriented against the direction of the magnetic field, it is in a higher energy state. The desired state is the lower energy state. In this case, if the temperature is higher than the absolute zero, the states will be flipped. One state will have more nucleus and those will be able to be measured. (Mlynárik, 2017) The initialization requires a stable ground state. In most variants, this is reached by colling, however, this is not practical for NMR. Since 1996 it has been established that a pseudo-pure state is sufficiant. In this mixed state, the result will only be a probability and needs to be repeated. Single state transitions are not possible to detect. There are about $10^{17}$ molecules with $10^{13}$ nuclei the lower energy state used. To manipulate the nuclei radio frequency is used. This frequency depends on the molecule and is called Larmor frequency which also depends on the strength of the magnetic field. (Jones, 2000) This behavior is illustrated in Figure 2-13. It shows the influence of the radio frequency on the spin.

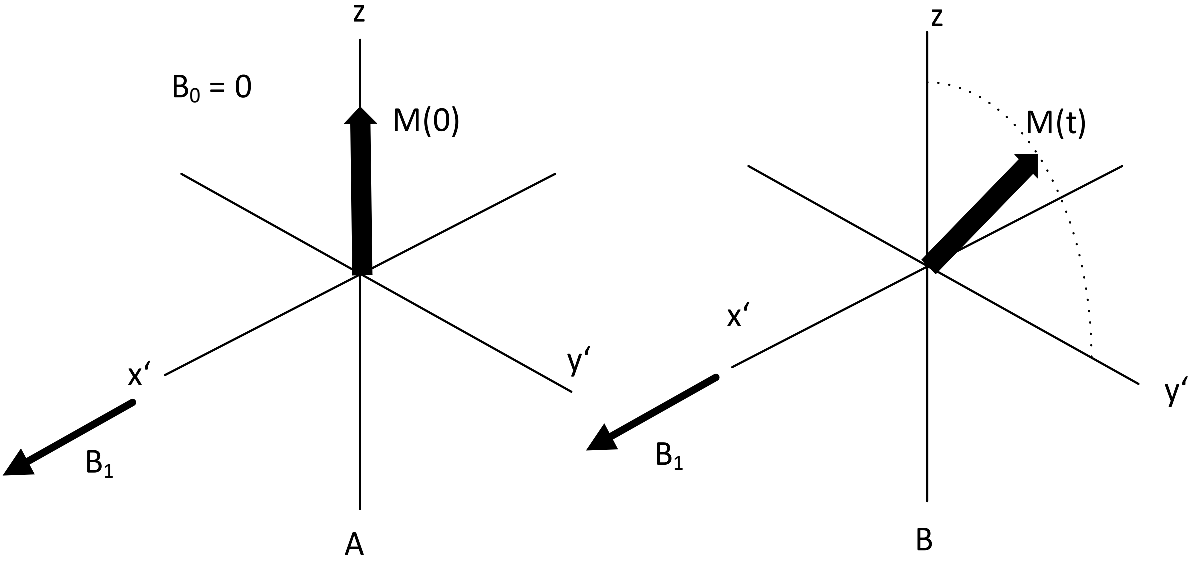


*Figure 2-13: The static nuclei when $B_0 = 0$ (A) and the spin caused by $B_1$ depending on time and the amplitude of the radio frequency field $B_1$. (cf. Mlynárik, 2017)*

### 2.3.4 Superconducting

As shown in Savage (2018) many modern quantum implementations use the technique of superconducting circuits. While most implementations use microscopic entities such as atoms or photons, superconducting circuits use an oscillator. Furthermore, they use an aluminum atom which is used in wires and plates. The basic circuit of a superconducting qubit is illustrated in Figure 2-14. To achieve the quantum effects for the computation there are two effects used. Firstly, the superconductivity. Superconductivity is the effect of the flow of electical fluid without any friction. Cochran and Mapother (1958) found out that this can be achieved with aluminum with a temperature of around 1 Kelvin which is around -272 degree Celsius. Secondly, the Josephson effect. This effect allows the circuit to be nonlinear without the need for dissipation or dephasing. The motion is described as the flux Φ and is threading the inductor which works as the center of mass in a mass-spring oscillator. With a Josephson Tunnel Junction, the circuit works like an artificial atom. This allows the selective transition from the ground state to the excited state. Because of this behavior it can be used as a qubit. (Devoret & Schoelkopf, 2013)

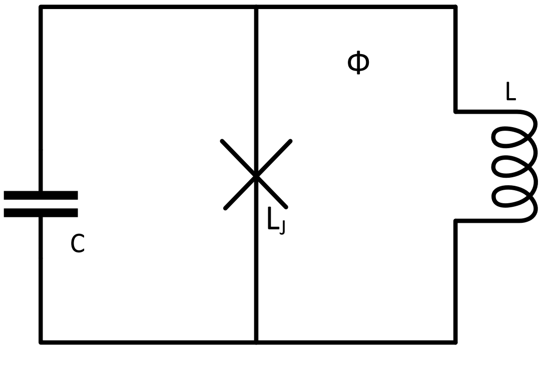


*Figure 2-14: Superconducting qubits consist of a capacitance C, the Josephson tunnel $L_J$ and the inductance L. The flux Φ is threading the loop between both inductances. (cf. Devoret & Schoelkopf, 2013)*

## 2.4 Algorithms

This section covers the three most important and most famous quantum algorithms known today. Starting with the most basic one, the Deutsch Algorithm answers the question if a function is balanced or not. The Groover Algorithm is used for searches in unsorted data. At least the Shor Algorithm is described which is famous for the fast factoring big numbers.

### 2.4.1 Deutsch's Algorithm

To illustrate the advantages of quantum computing in a type of computations, the Deutsch Algorithm is the most basic one to illustrate that. The problem is the following function

$$f\colon \{0,1\} \rightarrow \{0,1\}.$$

The function $\mathrm{f}(0)$ can be zero or one and the function $\mathrm{f}(1)$ can be zero or one as well. The values of the function are not necessarily of interest but it is important if the function is constant, $f(0) = f(1)$ or if the function is balanced, $f(0) \neq f(1)$. The following task is to determine the value of $\mathrm{f}$ but only computing it once. In a classic computer $f(0)$ and $f(1)$ would be calculated separately. In the quantum computer, we need these two qubits to be calculated in one iteration. The first one is the input and the second one is the internal hardware part. This is shown in Figure 2-15.

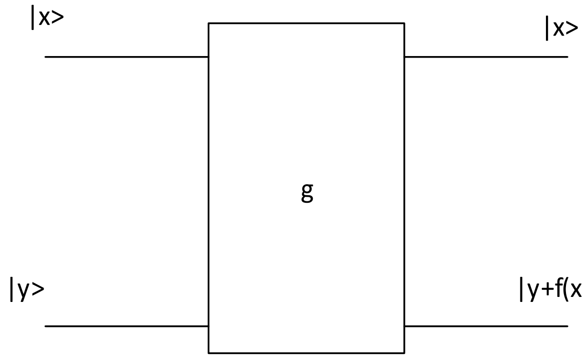


*Figure 2-15: Deutsch algorithm circuit. |x> is the input and |y> is the hardware part. The function g will only interact with the second qubit with $y \oplus f(x)$ (cf.Vedral & Plenio, 1998)*

On the inputs $|x\rangle$ and $|y\rangle$ a Hadamard Transformation needs to be applied to them establish the entangled state $|x\rangle|y\rangle = (|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$. With this input we have a superposition state of all four input states $|\Psi_{in}\rangle = |00\rangle - |01\rangle + |10\rangle - |11\rangle$. When applying the transformation of g in Figure 2-15 the output will be in the following state

$$|\Psi_{out}\rangle = |0f(0)\rangle - |0\overline{f(0)} + |1f(1)\rangle - |1\overline{f(1)}\rangle$$

$$= |0\rangle(|f(0)\rangle - |\overline{f(0)}\rangle) + |1\rangle(|f(1)\rangle - |\overline{f(1)}\rangle) .$$

Each of the four superpositioned input values from $|\Psi_{in}\rangle$ got transformed by $y \oplus f(x)$ in only one computation. The function is constant when $|\Psi_{out}\rangle = (|0\rangle + |1\rangle)(|f(0) - |\overline{f(0)}\rangle)$ and balanced when $|\Psi_{out}\rangle = (|0\rangle - |1\rangle)(|f(0) - |\overline{f(0)}\rangle)$. When a Hadamard Transformation is applied to |x⟩ at the ouput ist possible to say if the function $|\mathrm{x}\rangle$ is $|0\rangle$ is balanced otherwise not. (Vedral & Plenio, 1998)

### 2.4.2 Grover algorithm

The Grover Algorithm can detect the desired value in an unsorted data source tremendously faster than a classical computer. For $\mathrm{N}$ values a conventional computer needs $O = (N)$ steps. A quantum computer can do this because of the superposition states in $O = (\sqrt{N})$. To achieve this, there needs to be a function that returns one if $\mathrm{x}$ matches the desired value or zero. With the Hadamard Transformation, every value in the system has the same amplitude $\frac{1}{\sqrt{N}}$. This needs $O(\log N)$ steps. Then the function, which is a unitary transformation, is applied and the desired state does a phase shift. This transformation is described by

$$U|x\rangle|y\rangle = (-1)^{f(x)}|x\rangle|y\rangle \; .$$

If the function is positive a phase flip will be done. The $\mathrm{x}$ register consists all possible values and the $\mathrm{y}$ register consists of only one. The $\mathrm{y}$ register gets to a one with a NOT operation, and a Hadamard is applied to both. When the $\mathrm{U}$ transformation is applied to this registers. Because of the phase shift of the desired value the mean of the amplitudes has changed. Figure 2-1 shows this phase shift.

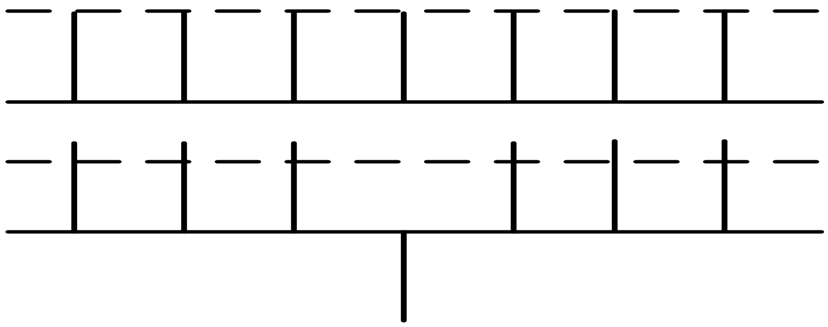


*Figure 2-16: Phase shift and change of amplitude mean. ( cf. Brands, 2011)*

Because the phase shift cannot be measured, an inverting $\mathrm{D}$ is applied with the mean of the amplitudes. The desired values will undo the phase shift and its amplitude grows to the amount of the mean, the other amplitudes shrink to the size of the mean. The difference is now $\approx \frac{2}{\sqrt{N}}$. This will be repeated $\sqrt{N}$ times to get the desired value with a high probability. (Grover, 1997)

### 2.4.3 Shor Algorithm

Shor's factoring algorithm is known for factoring big numbers out of prime numbers. On a classical computer there is no known way to do this in polynomial time. Shor used for the Quantum Fourier Transformation (QFT). $\omega_N$ is defined as the n root of unity $e^{2\pi i/N}$.

$$QFT_N|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{j*k}|k\rangle$$

The QFT multiplies the integer numbers j and k. It operates in the set of {0,….,N-1} with modulo N. $|j\rangle$ and $|k\rangle$ are integer values represented in a qubit. The $\mathrm{QFT_N}$ makes it possible to implement with $O(n^2)$ gates to be realized with $\mathrm{N=2^n}$.

The goal of Shor's algorithm is to factorize the integer n with a < n with the period of the function

$$f(x) = a^x \; mod \; n \; .$$

Therefor are two quantum registers with the length of log n bits required. The basic operation of Shor's algorithm is:

1. $R = |a\rangle|b\rangle \leftarrow |0 \dots 0\rangle|0 \dots 0\rangle$, initialize the two quibts with the length log n with zeros.
2. Apply the Hadamard transformation on qubit a: $R \leftarrow \frac{1}{\sqrt{N}} \sum_{X=0,\dots,N-1} |x\rangle|0 \dots 0\rangle$
3. Apply the oracle: $R \leftarrow U_f R = \frac{1}{\sqrt{N}} \sum_{X=0,\dots,N-1} |x\rangle|f(x)\rangle$
4. Measurement of register $|b\rangle$
5. Apply the QFT to $|a\rangle$
6. Measurement and output of $|a\rangle$

This will return the period p of the function $\mathrm{f(x)}$. For a large m are $\frac{1}{\log \log m}$ integers coprime to m. It is likely that the result is correct. It can be easily checked by trying to factorize n with p. The complexity to determine a real coprime of n is $O = ((\log n)^4)$. (Homeister, 2018)

## 2.5 **Large Quantum Computers**

Nowadays, the largest quantum computers are built with trapped ions or with superconducting chips. The main advantage of superconduction chips is that they can build like classic computer chips. They are also quite fast and can do operations on a billionth of a second. Their coherence time, however, is also only a few milliseconds. IonQ is functioning on an iron trap based quantum computer with 32-qubit. IBM made a five qubit processor available on their cloud platform in 2016 and upgraded it to a 20-qubit processor. They also developed a 50-qubit processor which might be build into commercial systems. Google has announced its 72-qubit processor. Both the IBM and Google developments are based on superconducting. With around 50-qubit systems it is possible to reach an equivalent of ten quadrillions of a conventional computer. With this it is possible to perform computations that are not possible with the computers which are available today and can induce quantum supremacy. (Savage, 2018)

# 3 CRYPTOGRAPHY

This chapter examines cryptography in general. It contains the goals of cryptography and why it is used. After that, a short overview of hash functions, what they are and which ones are often used follows. Subsequent to this the topics of symmetryic and asymmetric crypthography are discussed. Each one consists of a short description of the basics of each method. It will include a more detailed illustration of popular representatives of each of these methods and what the security recommendation for each one can be.

## 3.1 Goals

Depending on the used method, cryptography is used to achieve one or more of these four goals:

1. Confidentiality
2. Integrity
3. Authenticity
4. Liability

Confidentiality is needed when two parties want to communicate, but a third party must not be able to attend this communication. To achieve this, encryption is used. Both parties need the right keys to decrypt the message. There are symmetric methods where both parties need the same key and asymmetric methods, where a publicly available key is used to encrypt, and the recipient uses his private key to decrypt the message.

Integrity ensures that the message was not modified between the sender and the recipient. To achieve this, Hash algorithms are used. These algorithms compute a checksum of the message. If a single bit is changed the checksum will be completely different. The recipient can also compute his checksum and compare them.

Authenticity ensures that the message was really sent from the specified sender. For this, a Message Authentication Code (MAC) can be used. This works similar to a Hash but a secret code is involved. Both parties must know the secret code to create the secured thumbnail of the message.

A MAC cannot be used for liability purposes as both, the sender and the recipient, know the secret code, so both could create secured message. But when it is required to ensure the creator of a message, a digital signature is needed. This uses asymmetric keys and the sender uses his private key to secure the message. Therefore, only the sender can know the private key and only the sender was able to create the signature of this message. (Spitz, Pramateftakis, & Swoboda, 2011)

The methods used for these goals will be explained in more detail in the following sections.

## 3.2 Hash Functions

This section covers the most popular hash functions. It includes Message Digest (MD) 5 which can still be found in Transport Layer Security (TLS) 1.2. (Dierks & Rescorla, 2008) Furthermore, it looks at the whole family of Secure Hash Algorithm (SHA).

A hash function maps a set of any number of strings on a string with a defined length

$$h\colon \textstyle\sum^{*} \rightarrow \sum^{n},\ n \in \mathbb{N}\ .$$

It is referred to as a compression function when it maps a string with a fixed length on an another shorter fixed-length string

$$h\colon \textstyle\sum^{m} \rightarrow \sum^{n},\ m,n \in \mathbb{N}, \qquad m > n\,.$$

A cryptographic secure hash function needs to fulfill several requirements. It needs to be a one-way function. This means that it must be easy to calculate the hash function out of the input but it must be impossible to restore the input out of the hash. (Buchmann, 2008) It has also to be collision-resistant. There are two ways of collision-resistance, a weak collision-resistance and a strong collision-resistance. For the weak collision-resistance it must be impossible to find another message to the given hash in less then $2^n$ attempts. The strong collsion-resistance protects against finding any two messages with the same hash value. These messages can be chosen by the attacker. For a strong collision-resistance, the attack must try

$$j \approx \sqrt{2} * 2^{\frac{n}{2}}$$

times to find two matching messages. (Spitz et al., 2011)

### 3.2.1 Message-Digest Algorithm 5

MD5 comes from the family of MDx. Additionally, there is also MD2 and MD4. MD2 was too slow, and MD4 was not secure enough. MD5 was presented in 1992 by R.Rivest and was a milestone in hash development at that time. As a result, it was a widley used 128 Bit hash function and was implemented by many applications and protocols. (Sobti & Ganesan, 2012)

MD5 works with blocks of a size of 512 bits. The last block is padded to 448 mod 512 and a 64-bit value is added which contains the length of the message. Each of these 512-bit blocks is divided into 16 words with a length of 32 bit. A single Block will be complete after 64 rounds. The four 32-bit registers are initialized with fixed values. Each of the four words will be processed four times. Figure 3-1 shows that in the first round of each word the function F is applied which provides a 32-Bit value, and an addition modulo $2^{32}$. $M_i$ is a word of the message which is also aggregated with mod $2^{32}$. $K_i$ is a constant depending on the round I and is aggregated with mod $2^{32}$. Afterwards, a left shift is applied and at least word B is aggregated with mod $2^{32}$. A will then be the input value for B, B for C, C for D and D for A. (Rivest, 1992; Spitz et al., 2011)

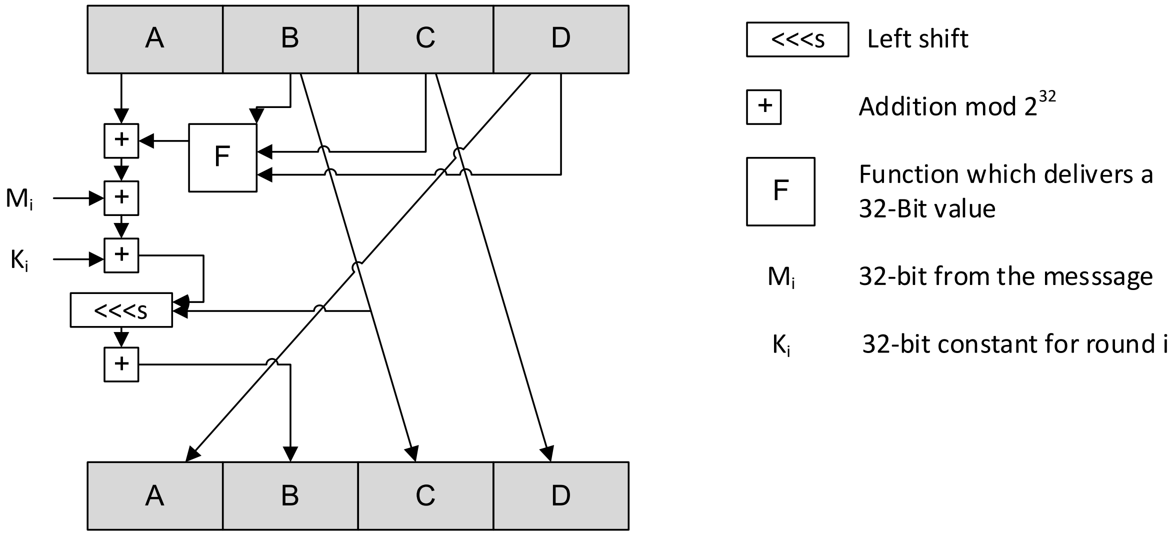

A
B
C
D
+
F
Mi
+
Ki
+
<<<s
+
A
B
C
D
<<<s Left shift
+ Addition mod 2^32
F Function which delivers a 32-Bit value
Mi 32-bit from the messsage
Ki 32-bit constant for round i

*Figure 3-1: Circuit for MD5 hash. ( cf. Spitz et al., 2011)*

Stevens (2006) shows that is possible find a collision with MD5 that can be archived in minutes. This was back in 2006, with today's computers it will be faster an MD5 should not be used anymore. Xie et al. (2013) has illustrated that the complexity of an collision with 2 block input diefferences is reduced to $2^{19}$ bit and a single input difference with a complexcity of $2^{46}$.

### 3.2.2 Secure Hash Algorithm 1

SHA-1 was released by the National Institute of Standards and Technology (NIST) in 1995. SHA-1 works with 512-bit blocks and padding in the last block like MD5. Each block consists of 16 words with a length of 32-bit. Each block will be processed in 80 rounds. The intern will be 80 words processed with:

$$w_i = shift(w_{i-1}\ XOR\ w_{i-8}\ XOR\ w_{i-14}\ \ XOR\ w_{i-1}),\ \ if\ i < 15.$$

This shift was implemented to improve security compared to SHA-0. SHA-1 works with five input values, each with a length of 32-bit wich will result in a 160-bit hash value. The five input values have a fixed initial value. Figure 3-2 shows a single round of the SHA-1 algorithm.

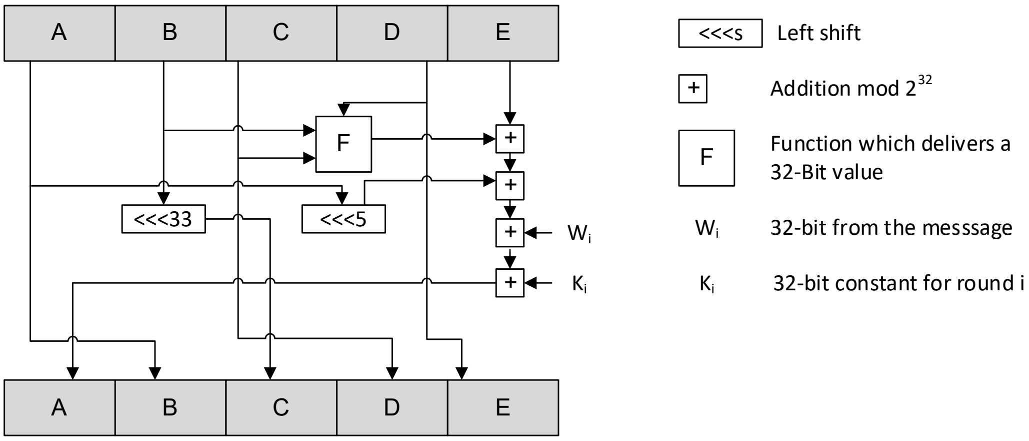

A
B
C
D
E
F
+
+
<<<33
<<<5
+
Wi
+
Ki
A
B
C
D
E
<<<s Left shift
+ Addition mod 2^32
F Function which delivers a 32-Bit value
Wi 32-bit from the messsage
Ki 32-bit constant for round i

*Figure 3-2: SHA-1 circuit. (cf. Spitz et al., 2011)*

Most of the manipulation happens in the last input word E. First, a function will be aggregated with mod $2^{32}$ to input E. This function varies depending on the round. Second, the shifted input A will be aggregated mod $2^{32}$ to E. At last, a input word and a constant which, depends on the round, will be aggregated from mod $2^{32}$ to E. The only other manipulation happens in input B with a shift. (Eastlake & Jones, 2001; Spitz et al., 2011) Stevens et al. (2017) demonstrated the fist collision with a full SHA-1. The could find a collision of an SHA-1 hash with two pdf documents. This can be done with a complexity of $2^{63}$. SHA-1 cannot be considered secure anymore.

### 3.2.3 Secure Hash Algorithm 2

SHA-2 is divided into two groups. SHA-256/224 and SHA-512/384. Only SHA-256 are handled in detail this paper, but the other two are fairly similar, the differences will be explained.

SHA-256 works with 64 32-bit words and eight working variables with a length of 32 bit each. The last block will be padded. The result is a hash value out of eight 32-bit words and has a final length of 256-bit. Each block will be processed in 64 rounds. For the first round each of the eight working variables will have a fixed initial value. The first 16 words represent the block and other 48 words are calculated with

$$w_i = \sigma_1^{256}(w_{i-2}) + w_{i-7} + \sigma_0^{256}\,(w_{i-15}) + w_{i-16},$$

$$if\ i < 15, + = addition\ mod\ 2^{32}, \qquad \sigma = rigth\ rotation$$

Figure 3-3 shows a full SHA-2 round.

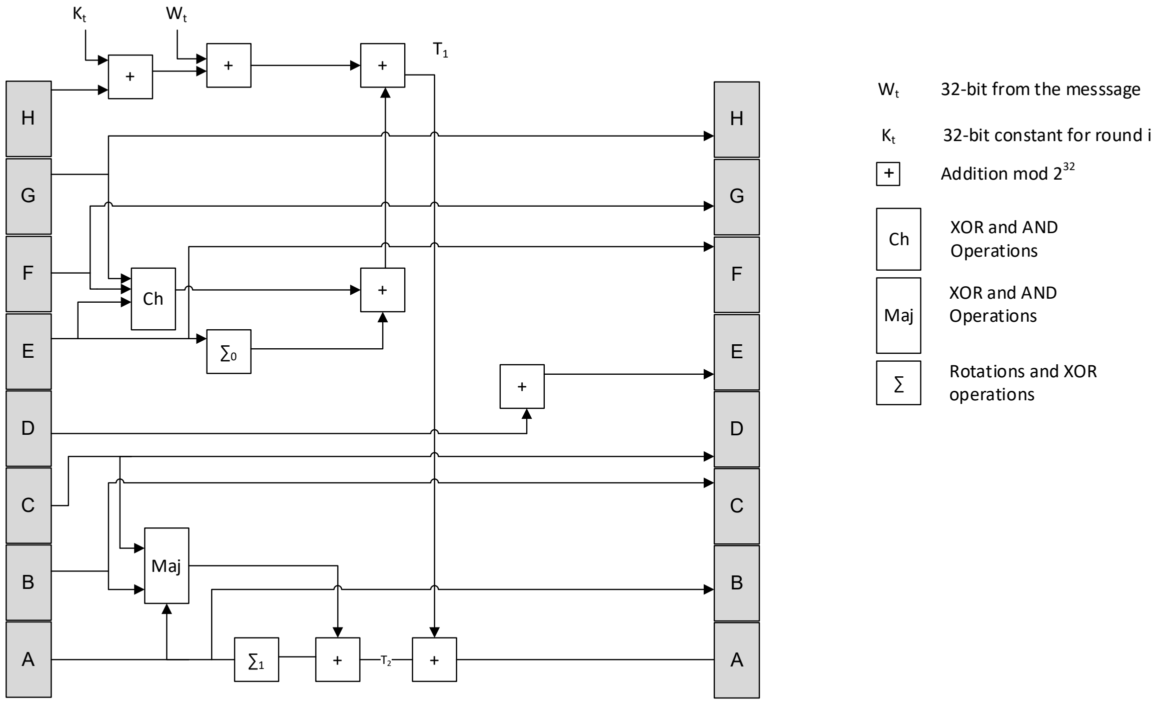

K$_t$
W$_t$
T$_1$
+
H
G
F
E
D
C
B
A
Ch
Σ$_0$
Maj
Σ$_1$
T$_2$
W$_t$ 32-bit from the messsage
K$_t$ 32-bit constant for round i
+ Addition mod $2^{32}$
Ch XOR and AND Operations
Maj XOR and AND Operations
Σ Rotations and XOR operations

*Figure 3-3: SHA-256 round. (cf. Chaves, Kuzmanov, Sousa, & Vassiliadis, 2006)*

$K_i$ is a round depending constant which will be aggregated with mod $2^{32}$. Maj and Ch are XOR operations with three input values. SHA-224 works in the same way but uses only seven of the eight 32-bit values as an output. (FIPSPUB180-4, 2015).

SHA512 uses 64-bit words instead of 32-bit and needs 80 rounds to process a single block. The output will be a 512-bit hash value. SHA-384 works the same but only uses six of the eight 64-bit values as an output.

SHA-2 can still be considered as secure today. There are some attacks such as Aoki et al. (2009) or Sanadhya and Sarkar (2008) which are all based on a reduced SHA-2 implementation. For the preimage the complexity is still around $2^{250}$ for SHA-256 and $2^{500}$ for SHA-512. For the a collision the complexity is lower with around $2^{30}$ but only for a 24 step SHA-2. Both do not threat the security for a whole implementation of SHA-2.

### 3.2.4 Secure Hash Algorithm 3

SHA-3 is based on the Keccak algorithm and is specified with 224, 256, 384 and 512-bit lengths. It uses a sponge construction which works completely different compared to SHA-1 and SHA-2. It has two phases, the absorbing and the squeezing phase. During the absorbing phase, the round function is applied to all blocks. SHA-3 defines 24 rounds. In the squeezing phase, the hash value will be dumped. Figure 3-4 shows this sponge construction.

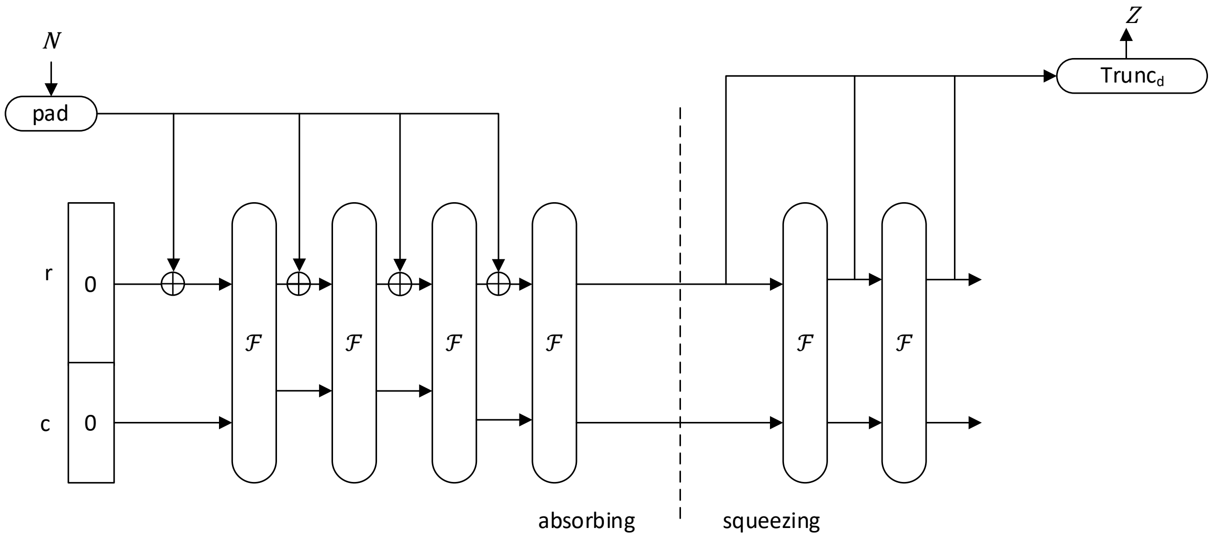


*Figure 3-4: The SHA-3 sponge construction. (cf. FIPSPUB202, 2015)*

The width of the function f is called b and is always 1600-bit wide in SHA-3. This b is divided into r and c and is defined as $r = b - 2n$ and $c = b - r$, where n is the length of the hash. Each block will be XOR linked with r. The message is padded to the length r. The first n bits from the last f function will be the hash value of the input. The 1600-bit state is stored in a three-dimensional array. The x and y index is five bit long and the z index 64-bit. The f function consists of five steps:

1. **Theta.** The result of theta is that every bit is XOR linked with two columns of the array. The two columns are the *[x-1,z]* and the *[x+1,z-1]* based on the current bit *[x,y,z].*

2. **Rho.** Each lane is rotated by a fixed-length depending on the $x$ and $y$ coordinate. Each bit in the lane is shifted by the value modulo and the lane size, which is 64-bit in SHA-3.
3. **Pi.** Changes the position of the 25 lanes. This rearrangement is defined as $A'[x,y] = A[(x+3y)\ mod\ 5, x]$.
4. **Chi.** Each bit in a row is XOR linked with two other AND linked bits in this row. This is defined as $A'[x,y] = A[x,y] \oplus ((A[x+1\ \ mod\ 5, y]\ \oplus 1) \wedge (A[x+2\ mod\ 5, y]\ ))$.
5. **Yota.** The last step adds a constant value depending on the round to the first lane. All other lanes are not affected. This 64-bit constant is XOR-linked to the lane.

All of these steps are applied 24 times for each block that will be computed. (FIPSPUB202, 2015)

## 3.3 Symmetric

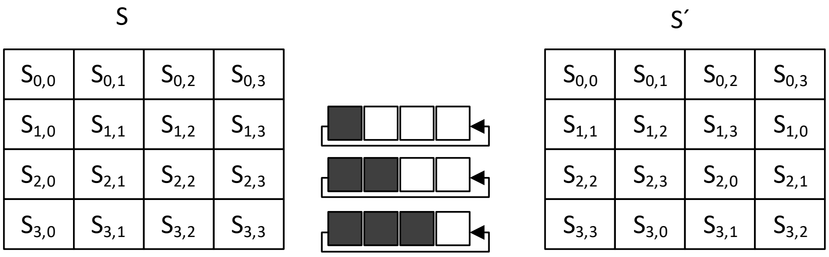
In this cryptosystem, both parties have the same key. For each set of communication pairs, a separate key is needed. The main advantage of symmetric cryptosystems is the speed. Symmetric cryptosystems can be implemented as a block or stream cipher. A block cipher maps an unencrypted input block to an encrypted output block with the same size. Stream ciphers encrypt every single bit. The key is used as an initial vector for the pseudorandom generator. (Buchmann, 2008) The most popular symmetric cryptosystems are Advanced Encryption Standard (AES) and its predecessor Data Encryption Standard (DES), which still can be found as Tripple DES today in TLS 1.2 (Dierks & Rescorla, 2008).

### 3.3.1 Advanced Encryption Standard

AES is based on the Rijndael algorithm and was specified as the AES standard back in 2001. It was the winner of a contest hosted by the NIST. AES is a symmetric block cipher. It is defined by a block size of 128-bits and can use keys with the length of 128, 192 or 256 bit. AES works with bytes. The 128-bit input is divided into 16 bytes and is organized as a 4x4 matrix. This will be divided into four 32-bit words which are a column in the matrix. AES uses a round function for encryption and decryption. This function consists of 4 steps und will be applied multiple times. AES-128 has 10 rounds, AES-192 has 12 rounds and AES-256 has 14 rounds. The four steps are:

1. **SubBytes.** The first step is called SubBytes. It uses a substitution of the input values with an S-Box. This uses a Galois field and adds nonlinearity to the algorithm. The S-Box has 256 values and is a 16x16 matrix. For in input value of 20 it will use the value in the third row und first column.

2. **ShiftRows.** A left-shift is applied. The first row stays as is. On the second rows is one left shit applied, on the third row 2 left-shifts and on the last row three left shifts. This is illustrated in Figure 3-5.



S

| | | | |
|---|---|---|---|
| $S_{0,0}$ | $S_{0,1}$ | $S_{0,2}$ | $S_{0,3}$ |
| $S_{1,0}$ | $S_{1,1}$ | $S_{1,2}$ | $S_{1,3}$ |
| $S_{2,0}$ | $S_{2,1}$ | $S_{2,2}$ | $S_{2,3}$ |
| $S_{3,0}$ | $S_{3,1}$ | $S_{3,2}$ | $S_{3,3}$ |

S´

| | | | |
|---|---|---|---|
| $S_{0,0}$ | $S_{0,1}$ | $S_{0,2}$ | $S_{0,3}$ |
| $S_{1,1}$ | $S_{1,2}$ | $S_{1,3}$ | $S_{1,0}$ |
| $S_{2,2}$ | $S_{2,3}$ | $S_{2,0}$ | $S_{2,1}$ |
| $S_{3,3}$ | $S_{3,0}$ | $S_{3,1}$ | $S_{3,2}$ |

*Figure 3-5: ShiftRows step. (cf. FIPSPUB197, 2001)*

3. **MixColumns:** This is a simple multiplication of 2 matrices. Each Column of the data will be multiplied with a fixed 4x4 matrix which increases the diffusion because a simple bit change changes the result of this operation. This is defined as

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}.$$

4. **AddRoundKey:** For each round a round key will be generated. Then the round key will be bit XOR linked to the block. For this the amount of round + 1 keys are needed. The cipher key is splitted in four 32-bit words. This occurs in the fist round. For all of the following data blocks the key is calculated as

$$w_i = w_{j-1} \oplus w_{i-4}, \qquad i \; mod \; 4 \neq 0$$

$$w_i = w_{j-1} \oplus (SubWord(RotWord((w_{i-4}) \oplus Rcon), \qquad i \; mod \; 4 = 0$$

SubWord is a S-Box substation and RotWord a rotation as described in the steps before. Rcon is a round depending constant.

The decryption works the other way around and uses inversed matrices. The steps are a bit AES-192 and AES-256 but it follows the same scheme. As mentioned before the first round is does only the AddRoundKey and the last round works without the MixColumns step. (FIPSPUB197, 2001)

### 3.3.2 Data Encryption Standard

The DES algorithm is the predecessor of the AES algorithm. There is also a version called triple DES which applies the algorithm three times with different keys. In this version it is still a common algorithm. It works with 64-bit blocks and a 64-bit key. The key is divided into eight bytes and the last bit of every byte is used for error correction. So the real key has a length of 56-bit. The algorithm is applied in rounds and uses 16 rounds.

The first step is a fixed initial permutation which is independent of the key. Then, the block is divided into the two 32-bit blocks $L_0$ and $R_0$. $R_0$ is now the input for $L_1$ and an $R_1$ is computed by

$R_1 = L_0 \oplus f(R_0, K_1)$. This will be repeated 15 times. In the last round the computation will be applied on $R_{16}$ and not an $L_{16}$. This is illustrated in Figure 3-6.

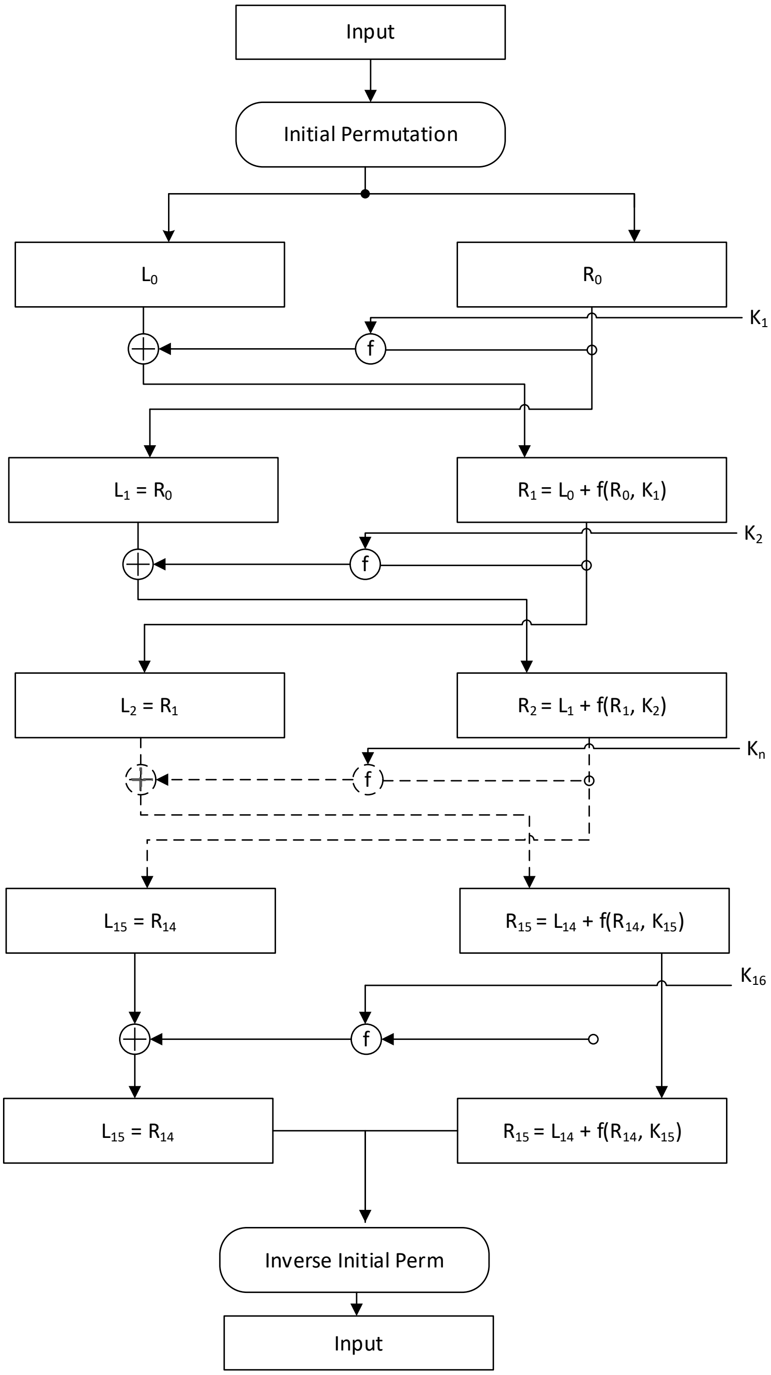


*Figure 3-6: DES algorithm encryption. (cf. FIPSPUB46-3, 1999)*

For the computation of the round, key R is used and expanded to 48 bits. This will be XOR computed with 48 bits of the key. The result is applied on eight S-Boxes with six bits for each box. The result of the S-Boxes are four bits which results in a 32-bit value. At least a permutation in applied. This will give the round key. Triple DES uses $O = E_{K3}\left(D_{K2}\left(E_{K1}(I)\right)\right)$. With this combination of encryption and decryption, a security level of 112 bit can be achieved. (FIPSPUB46-3, 1999) Triple DES must not be replaced in existing systems and is still secure enough nowadays, however, new systems should aim for AES. (BSI, 2019)

## 3.4 **Asymmetric**

This section covers two popular asymmetric cryptosystems. These systems use a public/private key mechanism. The public key is distributed and can even be stored in a public key store. Each member needs a set of a private and public key. The private key must be securely stored and no other but the owner of the key is allowed to know about it. Messages are encrypted with the public key of the receiver. Only the receiver can decrypt it with his private key. (Buchmann, 2008) The Rivest–Shamir–Adleman (RSA) algorithm from 1977 which is used for encryption and digital signatures. The other system is the Diffie-Hellman algorithm which is primarily used for key exchange. The Elliptic-Curve-Cryptography (ECC) which is not a cryptosystem but a method which decreases the key length with the same security and is often used today.

### 3.4.1 Rivest-Shamir-Adleman

RSA can be used for signatures and encryptions. The security of RSA is based on the difficulty of factoring two prime numbers.  The basic algorithm is

$$C \equiv E(M) \equiv\ M^e\ (mod\, n), \qquad for\ the\ message\ M$$
$$D(C) \equiv\ C^d\ (mod\, n), \qquad for\ the\ ciphertext\ C$$

The values $\mathrm{e}$ and $\mathrm{n}$ are a positive integer and the message $\mathrm{M}$ must have a length between zero and $\mathrm{n}$-1. The encryption key consists of $(\mathrm{e}, \mathrm{n})$ and the decryption key of $(\mathrm{d}, \mathrm{n})$. Both have their own set of keys. The steps to compute this keyset are:

1. Computing n out of two primes $\mathrm{p}$ and $\mathrm{q}$:

$$n = p * q.$$

2. Choose a large random number which is coprime to $(\mathrm{p}$-1) * (q-1) :

$$\gcd\big(d, (p-1) * (q-1)\big) = 1\ .$$

3. Lastly, $\mathrm{e}$ can be computed as the multiplicative inverse of $\mathrm{d}\ \mathrm{mod}\ (\mathrm{p}$-1) * (q-1):

$$e * d\ \equiv 1\ \big(mod\ (p-1) * (q-1)\big).$$

To choose $\mathrm{d}$, any prime number which is larger than $\mathrm{p}$ or $\mathrm{q}$ will be suitable. (Rivest, Shamir, & Adleman, 1978) But it is common to choose $\mathrm{e}$ first and use a fixed value of $e = 2^{16} + 1 = 65537$. Nowadays, there is no known algorithm for a classical computer to break this factoring problem. But with the increasing computing power, shorter RSA encryptions can be broken. (Spitz et al., 2011) The Bundesamt für Sicherheit in der Informationstechnik (BSI) considers a length for $\mathrm{p}$ of 2000 bit to be secure until 2022. Then $\mathrm{p}$ should be 3000-bit minimum. (BSI, 2019)

### 3.4.2 Diffie-Hellman

The Diffie-Hellmann algorithm is not a public key algorithm but is widely used for key exchange. It uses the problem of the discrete logarithm. For this, no algorithm is known to solve this efficiently. The basic operation is

$$A \equiv g^a \, mod \, p, \; a \in \{0,1,\dots,p-2\}.$$

The exponent a is the discrete logarithm of $\mathrm{A\ basis\ g}$. In this example the two parties are named Alice and Bob which are common in cryptography. For the key exchange Alice and Bob exchange $\mathrm{p}$ and a primitive root $\mathrm{g\ mod\ p}$. This can be sent over an unsecured connection. Now Alice computes $A \equiv g^a \, mod \, p, \; a \in \{0,1,\dots,p-2\}$ and sends $\mathrm{A}$ to bob. Bob now computes $B \equiv g^b \, mod \, p, \; b \in \{0,1,\dots,p-2\}$ and sends $\mathrm{B}$ to Alice. For Alice it is now possible to get the key from $B^a \, mod \, p$ and Bob from $A^b \, mod \, p$. The key is $K = g^{ab} \, mod \, p$. The attacker needs to figure out $g^{ab}$, which is not possible in a reasonable time. The only thing he can do is a man-in-the-middle attack. (Buchmann, 2008) The prime $\mathrm{p}$ should have a minimum length of 2000-bit today and over 3000-bit after 2022. For the value $\mathrm{g}$ every value between $1$ and $\mathrm{p\text{-}1}$ should be worth considering. This can be achieved when $\mathrm{g}$ is a prime root of $\mathrm{p}$. (BSI, 2019)

### 3.4.3 Elliptic Curve Cryptography

ECC is not a cryptosystem. It is a tool which helps to develop or adopt asymmetric cryptosystems. The advantage of ECC is the high level of security with short keys. A 160-bit ECC key can be compared with 1024-bit RSA. The basic equation for ECC implementations is

$$y^2 = x^3 + ax + b, \; \{y, a, b, c \in \mathbb{R}\}$$

and it operates with real numbers. It must not have multiple zeros and $4a^3 + 27b^2$ is not allowed to be zero. But for cryptographic purposes real numbers are not suitable. Here a finite field is used. One possible solution is to work with the modulo of prime number in a finite field. There is no curve but there are many points. To get the point $\mathrm{R}$ of two arbitrary points $\mathrm{P}$ and $\mathrm{Q}$ the following equations are used

$$\lambda = (y_2 - y_1)(x_2 - x_1)^{-1} \, mod \, p$$

$$x_R = (\lambda^2 - x_1 - x_2) \, mod \, p$$

$$y_R = (x_1 - x_3)\lambda - y_1 \, mod \, p.$$

Because the parameters $\mathrm{a}$ and $\mathrm{b}$ cannot take arbitrary values due the limitations of the zeros and the amount of possible points on the curve the NIST has defined some curves for each key length for various fields. The security of ECC is based on the Elliptic Curve Discrete Logarithm Problem and can be compared with the discrete logarithm problem. The performance seems to be more intricate because of the complex computation but this is balanced out by the short keys. (Spitz et al., 2011) Table 3-1 shows the security equivalent of ECC compared to DH.

| EC key length | Diffie-Hellman key length |
|---|---|
| 160 | 1024 |
| 224 | 2048 |
| 256 | 3072 |
| 384 | 7680 |

*Table 3-1: Comparison of the security of key lengths. (Spitz et al., 2011)*

### 3.4.4 McEliece

The McEliece Cryptosystem was introduced in 1978 in McEliece (1978) and was the first code based cryptosystem. It uses error correction codes for encryption purposes on basis of Goppa codes. It is still unbroken and is even used again in quantum computers. Therefore, it is in the category of post used quantum cryptography algorithms. A Goppa code is an error correction code. The basic procedure is (Löndahl, 2015; McEliece, 1978):

1. A linear code $\mathrm{C}$ is code generated from $(\mathrm{n,k})$ with the probability to correct $\mathrm{t}$ errors. $\mathrm{G}$ is the generator matrix for $\mathrm{C}$ and consists of $n * k$. $\mathrm{S}$ is a random $k * k$ matrix and $\mathrm{P}$ a random $n * n$ matrix. The private key is $(\mathrm{S,G,P})$, the public key is $(\hat{G}, t), \hat{G} = SGP$.
2. The message gets divided into a length of $\mathrm{k}$ und transformed to a vector by $\grave{c} = m\hat{G}$. A random vector $\mathrm{z}$ is generated with a length $\mathrm{n}$ and a weight $\mathrm{t}$. The ciphertexts is $c = \grave{c} + z$.
3. For the decryption $\hat{c}$ is computed by $\hat{c} = cP^{-1}$. With the decoing algorithm for $\mathrm{C}$ it is possible to decode $\hat{c} \rightarrow \widehat{m}$. Now the message can be computed by $m = \widehat{m}S^{-1}$

Code based cryptography has not been used frequently due to the resulting key sizes. Depending on the parameters the key can reach between 100 kilobytes and several megabytes. (Bernstein et al., 2009)

# 4 IMPACT OF QUANTUM COMUTERS

The following chapter discusses the impact of quantum computers in computation. This encompasses the possible improvements for solving problems in a much faster way than it is possible today. Another aspect is the impact on security as cryptographic systems use highly complicated mathematics problems.

## 4.1 Improvements

The classification of the severity of a problem the computational complexity theory is often used. This theory has two primary categories. First, time difficulty and second, space difficulty in solving problems. Space difficulty means the amount of memory that is needed to solve the problem, time difficulty classifies the problems by the amount of time which is required to solve the problem. In this paper, the focus lies on time complexity. (Hromkovič, 2014) Figure 4-1 shows the most important complexity classes for P.

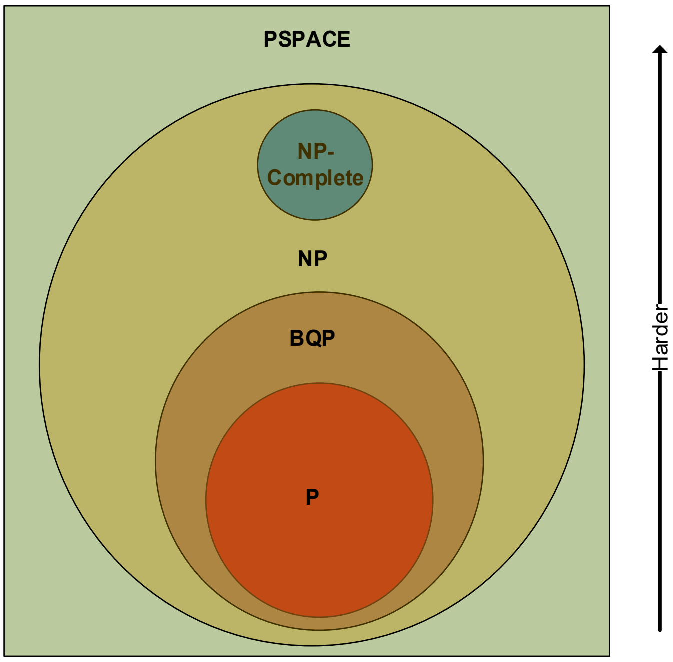


*Figure 4-1: Common complexity classes from P to PSPACE including the quantum class BQP. (cf. Aaronson, 2008)*

The class P stands for solvable in polynomial time. If a problem can be solved in polynomial time it is considered to be efficient. The runtime depends on the input length n and the power of n. The power of n needs to be small because power of 100 will not be efficient anymore. But the most natural problems which are solvable in P have a small power of n. Verifying the result can also be achieved in polynomial time. A classical computer can also solve such a problem in polynomial time, so a quantum computer is not needed for this process. (Shor, 2004) Then we also have the class nondeterministic polynomial time (NP). For such problems, a solution can be verified in polynomial time, but finding the solution takes considerately longer. To find a solution in polynomial time, a nondeterministic Turing Machine would be needed, which is a theoretical construct. NP covers a significant amount practical problems of today. The class of NP-Complete is the most complicated. This means, if an algorithm would be found so solve an NP-Complete problem in polynomial time, all NP and NP-Complete problems would be reduced to P. Until today no such algorithm was found and it is believed that $P \neq NP$. One problem is the traveling salesperson problem. There is also the class bounded-error, quantum polynomial time (BQP). Problems in this class are solvable in polynomial time by a quantum computer. All P problems and some of the NP problems lie in BQP. One famous representative of this class is the Shor algorithm for the factoring problem. This problem is solvable in polynomial time for a quantum computer. (Aaronson, 2008)  As it is not likely to find a quantum algorithm to solve NP-Complete it is more useful to search for NP problems. (Shor, 2004) From a practical point of view, quantum computers can be used everywhere where optimization is needed. Problems such as portfolio analysis in financial markets or protein folding in pharma and chemistry can be increased by using quantum computers. (Artemenko, 2019)

## 4.2 Impact on Security

Besides the improvements which are possible with quantum computers, there are also threats for information security. Which ones are suitable for being used against quantum attacks and which ones need to be replaced? Table 4-1 gives a short overview.

| Crypto algorithm | Type | Purpose | Impact of Quantum Computer |
|---|---|---|---|
| AES-256 | Symmetric key | Encryption | Secure |
| SHA-256, SHA-3 | -- | Hash functions | Secure |
| RSA | Public key | Signatures, Encryption | Not secure anymore |
| ECDSA, ECDH | Public key | Signature, key exchange | Not secure anymore |

*Table 4-1: Overview of the impact of quantum computer on current cryptosystems. (Mavroeidis, Vishi, D., & Jøsang, 2018)*

### 4.2.1 Public Key Cryptography

The Public Key Cryptography (PKC) systems that are used today are all vulnerable to quantum algorithms. (Moody, 2017) There is a chance for one out of seven that a large enough quantum computer will be built to break one of these cryptosystems by 2026 and a chance by one to two by 2031. (Mosca, 2018) The first step would be to move to another method that functions as the standardization for them. The NIST is currently working on a post-quantum standard. The deadline for submissions was November 2017. These algorithms will be analyzed for three up to five years and two years later, draft standards should be presented. (Moody, 2017) Consequently, there is not that much time, because the transformation to new standards also takes time. Mosca

(2013) states that the remaining time for a transition can be reduced to three variables, X,Y and Z. X is the time the information needs to be secret and PKC needs to be unbroken. Y is the number of years it will take to replace the current algorithms with new ones. Z is the time, in years, it will take to break the current PKC with quantum computers. As long as $X + Y < Z$ everything is well. However, if this equation is not true anymore, one has to act fast.

The threat for PKC is the Shor algorithm described in section 2.4.3. Most modern PKC are based on prime integer factorization or the discrete logarithm problem. Both are easy to compute for a quantum computer with the Shor algorithm. (Mavroeidis et al., 2018) Today, ECC is often used because it offers the same security of RSA but with shorter keys. ECC can also be broken with a modified Shor algorithm. Because of the shorter keys, ECC is easier for quantum computers. A 224-Bit ECC will need between 1300 and 1500 qubit and the RSA equivalent RSA-2048 needs 4096 qubits. (Kirsch & Chow, 2015; Proos & Zalka, 2003) But when considering error detection and error correction there are tens of millions up to billions of qubits required. (Mosca, 2018)

### 4.2.2 Symmetric Cryptography

Compared to PKC systems quantum computers do not have such a big impact on symmetric cryptography. The only algorithm known today which has an impact is the Grover algorithm described in section 2.4.2. But the effect is not as dramatic as in PKC systems. It only reduces the strength by $\sqrt{2^n = 2^{\frac{n}{2}}}$. AES-128 would only offer security of 64-bit. The solution would be to increase the key length and use AES-192 and AES-256. (Mavroeidis et al., 2018)

### 4.2.3 Hash Functions

Hash functions are also vulnerable via the Grover algorithm. Furthermore, hash functions can be combined with the birthday paradox. (Mavroeidis et al., 2018) The combination of these two elements results in a security level of $\sqrt[2]{N}$ and required space of $\sqrt[3]{N}$. (Gilles Brassard, Peter Høyer, & Alain Tapp, 1997) For a required security level of $\mathrm{b}$ bits, the hash function needs to produce a hash with a length of at least $3\mathrm{b}$. Old hash functions such as MD5 with 128-bit or SH1 with 160-bit cannot be seen as secure against quantum computers, but they should not be used anymore as described in section 3.2. The longer variants of SHA-2 and SHA-3 are still secure. (Mavroeidis et al., 2018)

# 5 POST QUANTUM CRYPTOGRAPHY

Post-quantum cryptography describes cryptosystems that are hard to break for quantum computers and for classic computers as well. This chapter covers the challenges post-quantum cryptography faces and why post-quantum cryptography is not common today. Furthermore, this chapter punctuates the difference to quantum cryptography. Additionally, the chapter covers what the approaches of modern post-quantum cryptography are and lastly a survey about numerous new algorithms which exist and how do they work will be described.

## 5.1 Challenges

First of all, there are classic PKC systems that exist which can resist quantum computers. While RSA is safe against classic computers, it is useless again quantum computers. The McEliece algorithm is also safe against quantum computers with a key length of a few million bits. The answer to why you should not switch to McEliece, when quantum computers are large enough, is obvious. The key size is not useable for a wide range of applications. Therefore, it is necessary to take one's time to develop better algorithms to be prepared when quantum computers become large enough. As described in chapter four time, the following is the critical part. Time is necessary to:

- Improve the efficiency of quantum cryptography.
- Build confidence in quantum cryptography.
- Improve the usability of quantum cryptography.

The following subsections will illustrate these challenges in more detail. (Bernstein et al., 2009)

### 5.1.1 Efficiency

Elliptic-curve signatures provide a security level of $O(b)$ bits with a key length of $O(b)$ bits and the verification only takes $b^{2+o(1)}$. This is valuable against classic computer but is of no help when it concerns quantum computers. But the algorithms which are safe against quantum computers need to be efficient as well. (Bernstein et al., 2009) Efficiency must be reached in two ways. First, efficient schemes are needed. Most schemes today have a longer key size and need more computational power than their nonquantum safe relatives. Many attempts to reduce the computational power have resulted in a negative impact on security. Therefore, it is possible that post-quantum algorithms will have a higher const on computation and storage. Second, efficiency improvement must be reached with more efficient implementations. Implementations need to be put into hardware to reduce time and energy demands. Compared to current hardware implementations such as AES, more powerful hardware is needed for these algorithms. (Niederhagen & Waidner, 2017)

### 5.1.1 Confidence

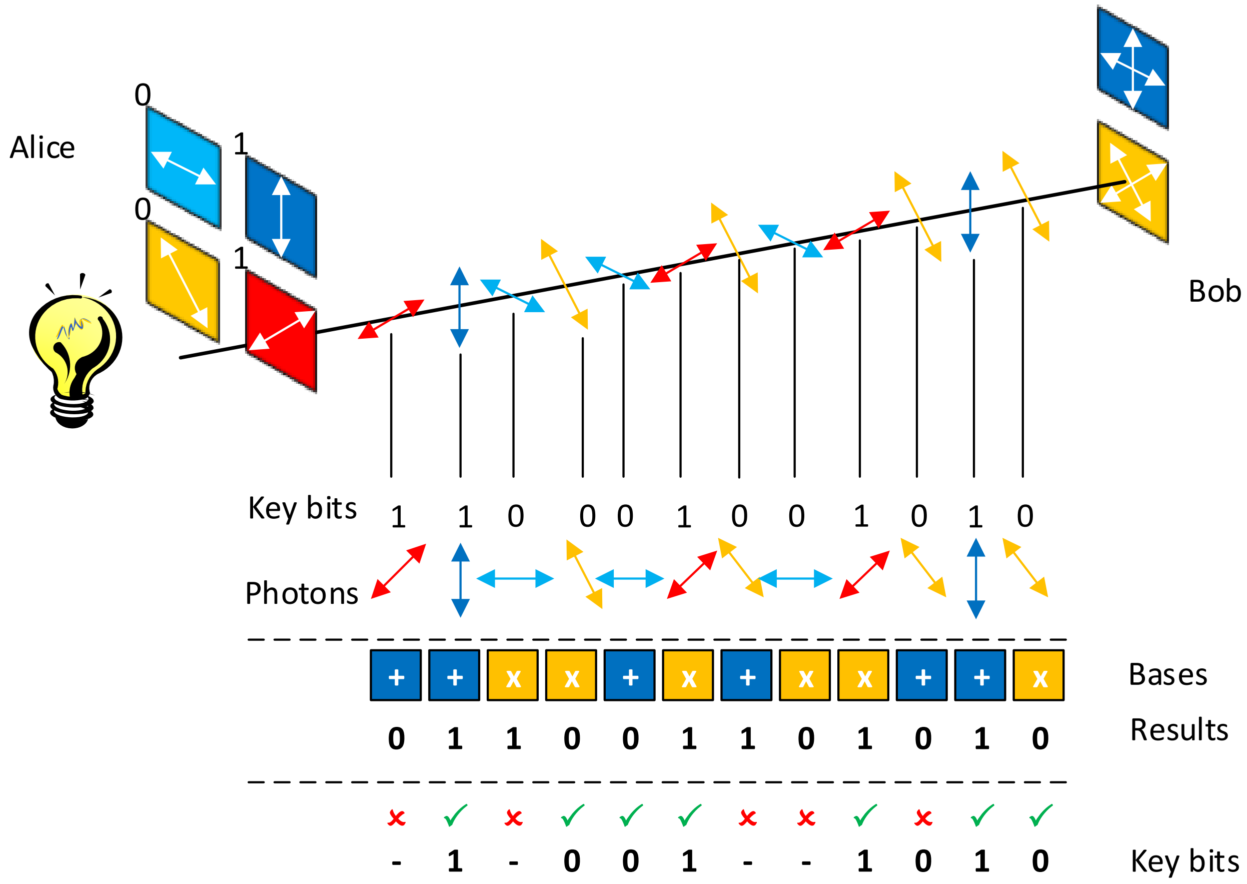
Confidence in cryptosystems grows over time. If there is no major exploit over many years it is considered to be safe for usage. For new cryptosystems which should be deployed over the next years as there is not so much time for aging the trust. It must be done with thorough security analysis and security proofs instead of aging. (Niederhagen & Waidner, 2017)

### 5.1.2 Usability

At the beginning, RSA was a simple trapdoor function. Nowadays, such implementations are considerably more advanced and include randomization and padding of the message. Implementations also encrypt a short random message and use the result of this computation as the key or the original message. All of this took years to accomplish. This shows that there is a long way to go between the standardization and a broad implementation basis in hard and software. These implemenations need to be fast but also prevent timing leaks and or other side-channel attacks. (Bernstein et al., 2009) During this transition phase, both classic and post-quantum cryptography are required. With such hybrid systems it is possible to achieve a high-security level even if the post-quantum part is not secure. (Niederhagen & Waidner, 2017)

## 5.2 Quantum cryptography

Quantum cryptography does not rely on unsolved mathematical problems as classic cryptography as it uses the laws of quantum mechanics. This independency from unsolved mathematical movements can be easily proven under the assumption that the laws of quantum mechanics are correct. Quantum cryptography just uses the behavior of photons or the spin of particles. One of the drawbacks of quantum cryptography is that both communication parties need at least one quantum device. It is not compatible with the classic computers existing today. The most common usage for quantum cryptography is the quantum key exchange (QKE). The most famous QKE protocol is the Bennet Brassard 84 (BB84) protocol. (Fehr, 2010) Figure 5-1 shows the basic scheme for a BB84 key distribution using polarized photons. For BB84 there are two channels required. A quantum channel and an ordinary classic channel. Both of them do not need to secure. Alice chooses a random set of qubits and a random set of polarizations. The polarizations with 0° and 45°, also called rectangularly, are interpreted as a zero, the qubits polarized with 90° and 135°, also called diagonal, are interpreted as a one. Alice now sends the random qubits with a random polarization over the quantum channel to Bob. Bob now measures the qubits he received at random as rectangular or diagonal. He guesses 50 percent correctly and the other 50 percent have a probability of 50 percent to measured right. Therefore, he has, in summary, a total probability of receiving 75 percent of the qubits correctly. The rest of the protocol now uses the classic channel. Bob sends which photons he has received and the polarization of the photons to Alice. Alice acknowledges the correct polarizations. To test against eavesdropping, they share a small number of the bits. If Alice confirms the bits received from Bob, the rest of the bits that are used are used as the shared secret. If the consensus of the bits or the polarizations exceed a defined threshold they suspect an eavesdropper, who is called Eve in the following and restart the key exchange. How can Alice and Bob detect an eavesdropper even if they communicate over public channels? The key is randomized and follows the principle of quantum mechanics. Eve can only guess the polarization if the wrong polarization is chosen. Only with a 50:50 chance Eve can detect the right result und submit it to Bob. Therefore, Eve has the same probability to guess correctly as Bob. The probability to guess right will decrease with only a few qubits, and Alice and Bob restart the key exchange if they detect that a certain number of bits are wrong. (Bennett & Brassard, 2014)



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Key bits | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Photons | ⤢ | ↕ | ↔ | ⤡ | ↔ | ⤢ | ⤡ | ↔ | ⤢ | ⤡ | ↕ | ⤡ |
| Bases | + | + | x | x | + | x | + | x | x | + | + | x |
| Results | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ |
| Key bits | - | 1 | - | 0 | 0 | 1 | - | - | 1 | 0 | 1 | 0 |

*Figure 5-1: BB84 scheme with polarized photons. (cf. Uysal, Capsoni, Ghassemlooy, Boucouvalas, & Udvary, 2016)*

## 5.3 Basic Approaches

This section discusses different schemes which are considered to be safe against quantum computers. It shows the principle of the different schemes and which algorithms use which scheme. The detailed function of the algorithms is explained in section 5.4.

After the first submission round of the standardization of post-quantum cryptography by the NIST, the approaches in the subsections will focus on these results. For PKC, there will only be lattice, and code-based cryptography, for signatures only hash-based and multivariate. Lattice would be also suitable for signatures but will not be covered in this paper. (Alagic et al., 2019)

### 5.3.1 Code-Based Cryptography

Code-based cryptography uses error correction codes to encrypt messages. The original message is hidden in the encrypted version because there is added redundant information, therefore these systems can correct errors during the transmission. The message $\mathrm{m}$ is encoded to a ciphertext $\mathrm{c}$. Now $\mathrm{c}$ is transmitted over a public channel. The recipient might no get $\mathrm{c}$ as some bits can be flipped during the transmission, hence he gets the message $\mathrm{r}$. With the error vector $\mathrm{e}$ with the weight $\mathrm{w}$, where w bits are one and the others are zero, it is possible to correct less ten $\mathrm{w}$ errors. Otherwise, the decoding fails. Now it is possible the decode $\mathrm{c}$ with the inverse operation and the recipient receives $\mathrm{m}$. The security in these systems is based on the difficulty in the decoding of the random codes. With the right parameters it can be infeasible. Currently, however, not many implementations exist. The most important is McEliece in section 3.4.4. (Niederhagen & Waidner, 2017) The downside of these systems is the resulting key size. They can reach a megabyte for high-security levels. In some of the newer systems they tried to implement more structure into the keys which would allow more compression. But many of them are broken. The best recommendation is still the original McEliece with some modifications when post-quantum cryptography is concerned as it has proven its security over several years. (Bernstein & Lange, 2017)

### 5.3.2 Lattice-Based Cryptography

A lattice is a set of collections of points in an $\mathrm{n}$-dimensional space. These points have a periodic structure. This is illustrated in Figure 5-2.

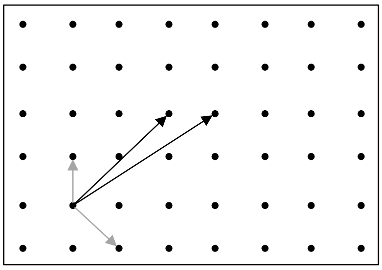


*Figure 5-2: A two dimensional lattice with two possible bases. (cf. Micciancio, 2011)*

The n-dimensional lattice is generated of n linearly independent vectors

$$\mathcal{L}(b_1, \dots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}.$$

The vectors $b_1, \dots, b_n$ are the basis of the lattice. The usage of lattices for cryptography was discovered by Ajtai (1996) and started a new field of research. Lattice-based cryptography relies on the difficulty level of these lattice problems. The simplest one is the shortest vector problem (SVP). This problem searches for the shortest nonzero vector in a lattice with basis B. The best-known algorithm for this is the Lenstra–Lenstra–Lovász-algorithm from Lenstra et al. (1982) which can achieve an approximation in $2^{O(n)}$, where n is the dimensions of the lattice. The advantage of lattices is the efficiency of the algorithms which compete with the best-known alternatives such as the simple implementation. Another benefit is that lattice problems are probably safe against quantum computers because there no algorithm has been found which would solve the problem in polynomial time. (Micciancio, 2011)

### 5.3.3 Hash-Based Signatures

Digital signatures used today rely on the security of the used public-key encryption scheme such as RSA. They are resistant against quantum computers. Hash-based signatures only rely on cryptographic hash functions. The security is based on the collision resistance of the used hash functions. This is also the minimum requirement for the use as a digital signature and to sign documents with the private key. There must not be two documents with the same digital signature. Hash bases signatures are the most promising post-quantum signature candidates. As long as the used hash function is secure it is unimportant which function is used. Hash-baes signatures were invented by Merkle (1990). He used the one-time pad scheme of Lamport (1979) for the signature scheme. They are fundamental and only need a one-way function and are required for a digital signature. This one-time signature is the most fundamental one but has a big disadvantage. It is only possible to sign and verify a single document, this limit does not fit the requirements for most real-world applications. (Buchmann, Dahmen, & Szydlo, 2009) The one-time signature works the following way. For the key pair, Alice chooses two random strings $x_0$ and $x_{1.,}$ these are her private keys. The public key is $(h(x_1), h(x_2))$, where $h$ is a publicly known hash algorithm. If Alice wants to sign a zero, she uses $x_0$. Bob now uses computes the hash of $x_0$ and compares it to the first half of the public key. If the same key is used a second time the security level decreases tremendously. To use this in a more practical way Merkle shows in Merkle (1979) and Merkle (1990) that it would be possible to combine $2^k$ public keys into a single key which can be used to verify $2^k$ signatures. For this a $2^k$ key pair must be created like above. The resulting public keys $Y_1, \dots, Y_{2^k}$ are arranged as a binary tree with k+1 levels. Figure 5-3 shows such a Markle tree with four levels.

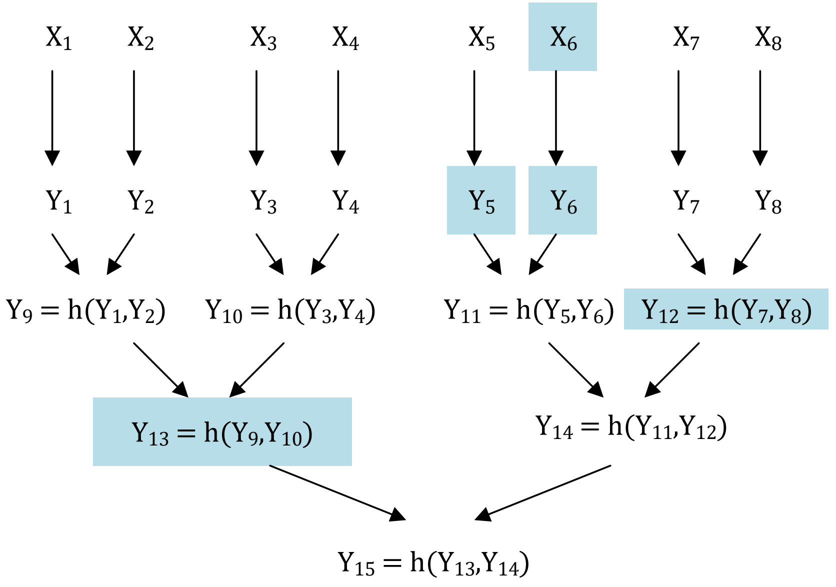


*Figure 5-3: Merkle tree with the demonstration of the public key generation for k=3. (cf. Bernstein & Lange, 2017)*

After generating the $2^k$ public keys for each message X, the public keys in the nodes of the tree are computed starting with $Y_{2^k+1} = h(Y_1, Y_2)$, and ending with the root node. The key in the root node is the public key for the whole system. Even when the public key is now a single hash value, more information for the verification is required. The whole chain starting with X and containing all public keys up to the root node. These systems are only affected by the Grover algorithm and not by the Shor algorithm, therefore they can be used for post-quantum signatures. But because a secret key is not allowed to be used a second time, the singer needs to manage the secret keys which can pose a problem. These systems are called stateful systems. (Bernstein & Lange, 2017) For environments where the state management is too difficult or impossible, Bernstein et al. (2015) introduced a stateless system which could be a drop-in replacement for current signatures, with the downside of larger signatures.

### 5.3.4 Multivariate Signatures

The basic approach multivariate cryptographic systems rely on are multivariate quadratic polynomials. These look like the following:

$$p^{(1)}(x_1, \dots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^{(1)} * x_i x_j + \sum_{i=1}^{n} p_i^{(1)} * x_i + p_0^{(1)}$$

$$p^{(2)}(x_1, \dots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^{(2)} * x_i x_j + \sum_{i=1}^{n} p_i^{(2)} * x_i + p_0^{(2)}$$

$$p^{(m)}(x_1, \dots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}^{(m)} * x_i x_j + \sum_{i=1}^{n} p_i^{(m)} * x_i + p_0^{(m)}$$

This is known as the multivariate quadratic polynomial problem. It states that for $p^{(1)}(x), \dots, p^{(m)}(x)$ in n variables a vector $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$ must be found so that $p^{(1)}(\bar{x}) = \cdots = p^{(m)}(\bar{x}) = 0$. (Petzoldt, Chen, Yang, Tao, & Ding, 2015) The problem is NP-Hard even when it is only used with quadratic polynomial and over a finite field GF(2). (Garey & Johnson, 2009) For a cryptosystem, an invertible quadratic map F is created. The structure of F must be hidden in the public key. This is achieved with 2 linear invertible maps S and T. The public key is a compound function of F, S and T which looks this way $P = S \circ F \circ T$. To sign a message m and get the signature w, Alice has to do the following steps:

$$m' = hash(m) \ \rightarrow y = \ T^{-1}(m') \rightarrow x = F^{-1}(y) \rightarrow w = S^{-1}(x)$$

Bob can check the signature easily by checking if $hash(z) = P(w)$. (Niederhagen & Waidner, 2017; Petzoldt et al., 2015)

## 5.4 Algorithms

The NIST has reduced the number of candidates from 69 to 26. The candidates are divided into the groups encryption and key-encapsulation mechanism (KEM) or signatures. After round one there are only lattice and code-based systems are available for encryption, for signatures are lattice, hash-based and multivariate systems possible. (Moody, 2019) The distribution of the round two candidates are shown in Table 5-1.

| | **KEM/Encryption** | **Signature** | **Sum** |
|---|---|---|---|
| **Lattice** | 9 | 3 | **12** |
| **Code-based** | 8 | - | **8** |
| **Hash-based** | - | 2 | **2** |
| **Multivariate** | - | 4 | **4** |
| **Sum** | **17** | **9** | **26** |

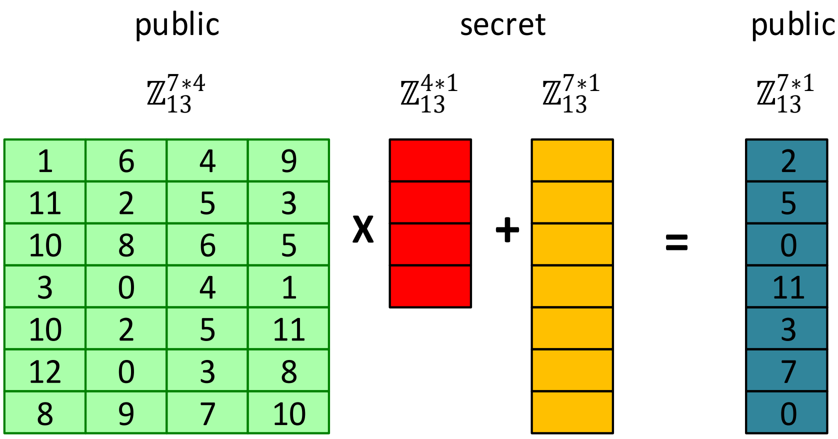
*Table 5-1: Distribution of round two candidates in the NIST PQC challenge. (Moody, 2019)*

To describe all 26 candidates is beyond the scope of this paper. It will handle a representative of each category.

### 5.4.1 Lattice-based encryption

As representative of this category Google's Newhope algorithms is chosen. It provides KEM and also encryption. Besides that, it is used today. Google has implemented it in their Browser Chrome and also in some of the services. This can be checked in the security panel of the browser. If the key exchange algorithm is combined elliptic-curve and post-quantum 1, then the Newhope algorithm was used. (Braithwaite, 2016)

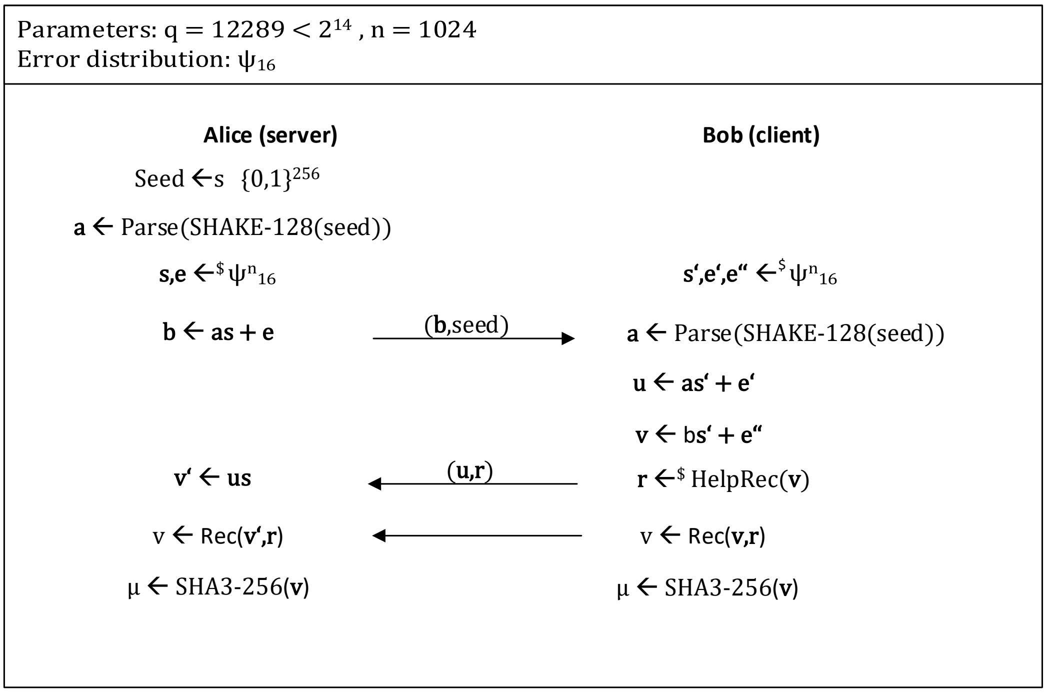
Newhope is based on the ring learning with errors (RLWE) problem. This is a specialized problem of learning with errors (LWE) to polynomial rings over finite fields. Figure 5-4 shows the general. Uses a filed of $\mathbb{Z}$ with modulo $q$ The input field and the result field is publicly known. Only with the multiplication it would be solvable with Gaussian elimination. But adding a small error makes it a difficult problem. In RLWE all elements would be column vectors. (Pöppelmann, 2018)



$$\underbrace{\begin{pmatrix} 1 & 6 & 4 & 9 \\ 11 & 2 & 5 & 3 \\ 10 & 8 & 6 & 5 \\ 3 & 0 & 4 & 1 \\ 10 & 2 & 5 & 11 \\ 12 & 0 & 3 & 8 \\ 8 & 9 & 7 & 10 \end{pmatrix}}_{\mathbb{Z}_{13}^{7*4}} \times \underbrace{\begin{pmatrix} \; \\ \; \\ \; \\ \; \end{pmatrix}}_{\mathbb{Z}_{13}^{4*1}} + \underbrace{\begin{pmatrix} \; \\ \; \\ \; \\ \; \\ \; \\ \; \\ \; \end{pmatrix}}_{\mathbb{Z}_{13}^{7*1}} = \underbrace{\begin{pmatrix} 2 \\ 5 \\ 0 \\ 11 \\ 3 \\ 7 \\ 0 \end{pmatrix}}_{\mathbb{Z}_{13}^{7*1}}$$

*Figure 5-4: LWE problem of a 7*4 matrix with modulo 13. (cf. Pöppelmann, 2018)*

The scheme for Newhope is illustrated in Figure 5-5.



Parameters: $q = 12289 < 2^{14}$ , $n = 1024$
Error distribution: $\psi_{16}$

| **Alice (server)** | | **Bob (client)** |
|---|---|---|
| Seed ← s $\{0,1\}^{256}$ | | |
| **a** ← Parse(SHAKE-128(seed)) | | |
| **s,e** $\xleftarrow{\$} \psi^{n}_{16}$ | | **s‘,e‘,e“** $\xleftarrow{\$} \psi^{n}_{16}$ |
| **b** ← **as + e** | $\xrightarrow{(\mathbf{b},\text{seed})}$ | **a** ← Parse(SHAKE-128(seed)) |
| | | **u** ← **as‘ + e‘** |
| | | **v** ← **bs‘ + e“** |
| **v‘** ← **us** | $\xleftarrow{(\mathbf{u},\mathbf{r})}$ | **r** $\xleftarrow{\$}$ HelpRec(**v**) |
| v ← Rec(**v‘**,**r**) | ← | v ← Rec(**v**,**r**) |
| μ ← SHA3-256(**v**) | | μ ← SHA3-256(**v**) |

*Figure 5-5: Scheme for Newhope key exchange. (cf. Alkim, Ducas, Pöppelmann, & Schwabe, 2015)*

First Alice generates a random 256-bit seed. The seed is hashed with SHAKE-128, a variant of SHA-3, the result is the polynomial coefficient $a$. Now Alice generates the secret values $s$ and $e$. These parameters have a distribution based on the centered binomial distribution with $k = 16$. Alice computes $b$ and sends it with the seed to Bob. Bob has by now computed his own $s'$, $e'$ and $e''$. Now he computes also $a$ from the seed. He computers $u$ with $a$, $s'$ and $e'$ and $v$ which is the same but with $e''$. With the HelpRec with input $v$, he computes $r$ and sends it with $u$ to Alice. Alice now computes $v'$ which is the same as $v$. With the Rec function both can compute the same input for SHA3-256 which returns the key. (Alkim et al., 2015)

### 5.4.2 Code-based encryption

On of the second round candidates in the NIST competition is the classic McElice. (Moody, 2019) A brief explanation of this cryptographic scheme can be found in subsection 3.4.4 and will note be explained further here.

### 5.4.3 Lattice signature

The chosen scheme is the qTESLA scheme. It uses also the RLWE problem and offers a lightweight implementation. It also provides multiple security levels depening on the parameter set. (Alkim, Barreto, Bindel, Longa, & Ricardini, 2019)

The description of the algorithm here is a simplified version of (Alkim et al., 2019), but it is enough to understand the principle of its function. The parameters used in this description are dimension of the Ring $\mathrm{n}$, the modulus $\mathrm{q}$, the standard derivation $\sigma$, the number of nonzero elements of the polynomial $\mathrm{h}$, the polynomial $\mathrm{c}$, the rejection parameters $\mathrm{E}$ and $\mathrm{S}$, the Bounds for coefficients $\mathrm{B}$ in the hidden polynomial $\mathrm{y}$ and at least the number of bits dropped in rounding $\mathrm{d}$. Further a short explaination of the functions used in the algorithm based on (Gérard & Rossi, 2019).

- **PRF:** Pseudorandom function for a random seed.
- **GenA:** Generation of a random, uniformly polynomial $\mathrm{a}$.
- **GaussSampler:** A polynomial sampling based on the gaussian distribution.
- **Check[S,E]:** Checking the polynomial coefficient of $\mathrm{e}$ and $\mathrm{s}$ are not to large.
- **ySampler:** Samples a random, uniformfly polynomial $\mathrm{y}$.
- **H:** A Hash function which is resistant again collisions.
- **Enc:** Encoding a bitstring into the polynomial $\mathrm{c}$.

The algroitm conists of three parts, the key generation, the singing, and the verification part. In this paper a simplified version of (Alkim et al., 2019) is used which was introduced in Gérard and Rossi (2019):

- **KeyGen:** The algorithm starts with the generation of multiple random seeds with the $\mathrm{PRF}$ function. Now the polynomial $\mathrm{a}$ is generated with $\mathrm{GenA}$ and the Input $\mathrm{seed_a}$. The secret random polynomials $\mathrm{s}$ and $\mathrm{e}$ are sampled with the $\mathrm{GaussSampler}$ with the corresponding seed and a counter value as input. The computation is done in a Loop and can only exit it if it passes the check of the corresponding Check function. If this is done the main part of the public key is computed by $t = a * s + e \, mod \, q$. The secret key consists of $sk \leftarrow (s, e, seed_a, seed_y)$ and the private key $pk \leftarrow (seed_a, t)$.
- **Sign:** The input for the signing is the message $\mathrm{m}$ and secret key $\mathrm{y}$. The output is the signature which is the sum of $\mathrm{z}$ and $\mathrm{c}$. First at all a random value is generated out aif a new random value $\mathrm{r}$, the $\mathrm{seed_y}$ and the message $\mathrm{m}$. Than the polynomial $\mathrm{y}$ is sampled with $\mathrm{ySampler}$ with the random value and the counter value. This polynomial is used to compute

$v = a * y \, mod \, q$. The rounded version of v gets now hashed with the message m. This gets encoded into a space polynomial c with h entries in {-1;1}. The signature is now computed by $z = y + s * c$. The signature must pass two tests. First, z must be in $R_{q,[B-S]}$ and $w = v - e * c \, mod \, q$ must be well rounded. If any of the checks fail, the process starts again with a new computed y with a new random value.

- **Verify:** The verification is very lightweighted. The message m, the signature $\sum = (z, c)$ and the public key pk is required. Firt the parameter a is computed like in the key generation. Then w is computed with $w = a + z - t * c$. The signature is accepted if the conditions $z \in R_{q,[B-S]}$ and $c \neq Enc(H([w]_M, m))$

### 5.4.4 Hash-based signature

SPHINCS replaces the one-time pad scheme (OTS) from section 5.3.3 and replaces it with a few-time scheme (FTS) to reduce the size of the signature. (Bernstein et al., 2015) It uses Merkle trees whose leaves are Witernitz OTS (WOTS). The leaves are Hash to Obtain Random Subset Tree (HORST) trees which can sign more than one message. (Aumasson & Endignoux, 2018) Figure 5-6 shows the general construction of the SPHINCS scheme.

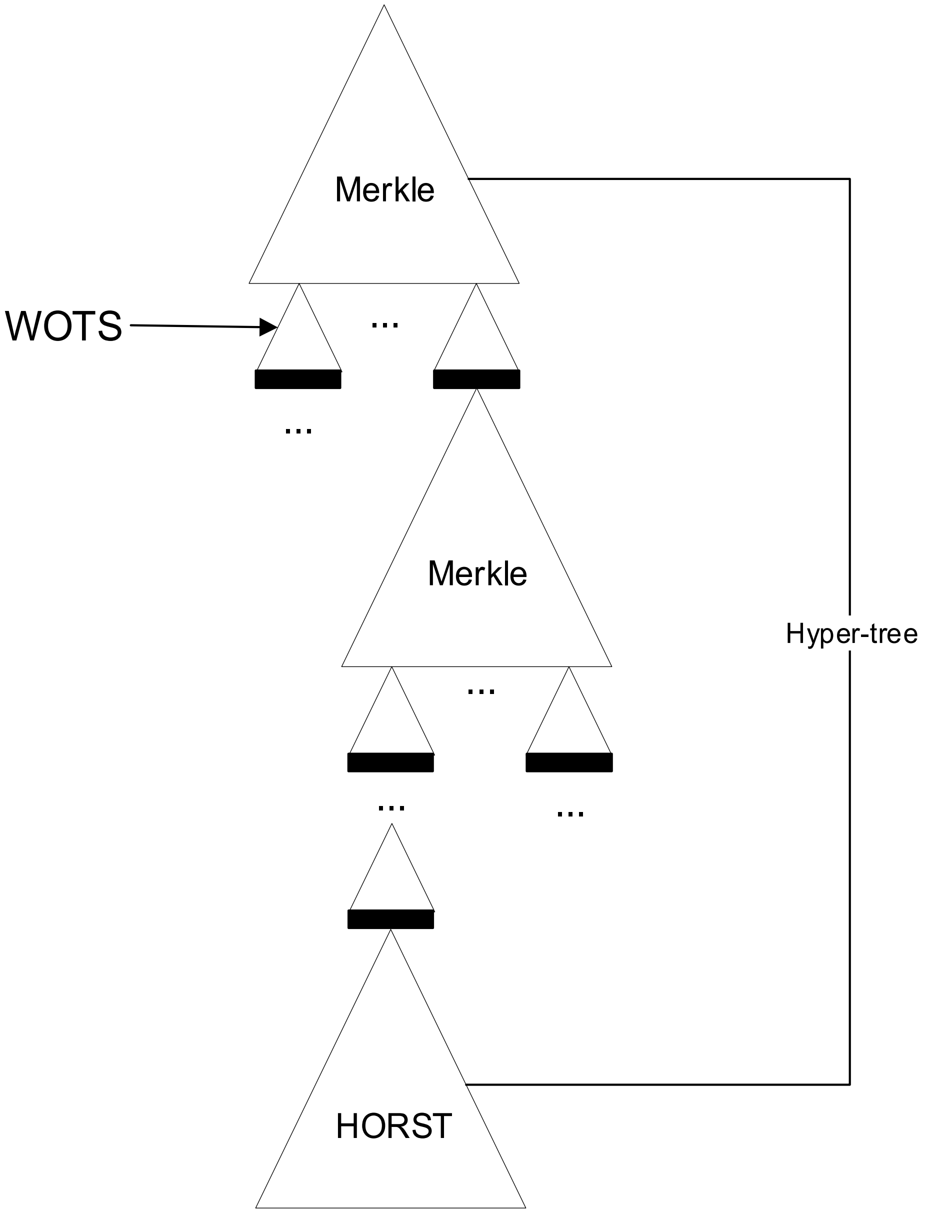


*Figure 5-6: Illustration of the SPHINCS construction. A hyper-tree connected by WOTS instances. At least a HORST tree leads to the singed message. (Adopted from Aumasson & Endignoux, 2018)*

The basis Merkle tree used in SPHINCS is described in subsection 5.3.3.

WOTS$^+$ is defined in Hülsing (2013). WOTS$^+$ is an improved variant of WOTS. WOTS$^+$ us parameterized with a few parameters. The security parameter $\mathrm{n}$ which must be in $\mathbb{N}$, and the length of the message. For the time-memory trade of the Witernitz parameter is used, which is defined as $w \in \mathbb{N}, w > 1$. At least the are the $\mathrm{l}$ parameter. It assembles out $\mathrm{l}_1$ and $\mathrm{l}_2$ which are defined as

$$l_1 = \left\lceil \frac{m}{log(w)} \right\rceil, l_2 = \left\lceil \frac{\log\left(l_1(w-1)\right)}{\log\left(w\right)} \right\rceil + 1, l = l_1 + l_2.$$

WOTS$^+$ uses Functions $\mathrm{F_n}$ with a keyspace $\mathrm{K_n}$. This can be seen as a non-compressing hash function. In there is a chaining function $c_k^i = f_k(c_k^{i-1}(x, r) \oplus r_i)$. The elements of the function are $x \in \{0,1\}^n$ is the input value, $\mathrm{i}$ is defined as the iteration counter which must be in $\mathbb{N}$, the key $k \in$

$K$, and the random elements $r = (r_1, \ldots, r_j) \in \{0,1\}^{n*j}$ where j needs to be bigger as i. Firstly a XOR operation is done in every round with the bitmask and the intermediate value. The function evaluates $f_k$ afterward. WOTS$^+$ consists of three functions. Firstly there is the key generation algorithm. The algorithm chooses $l + w - 1$ n-bit random strings during the input of the security parameter n. The first l elements the secret key $sk = (sk_1, \ldots, sk_l)$ consists of random bit strings. The randomization elements r for c are the remaining w-1 bit strings. Next, a function key is randomly chosen. The public key is computed as

$$pk = (pk_0, pk_1, \ldots, pk_l) = ((r, k), c_k^{w-1}(sk_1, r), \ldots, c_k^{w-1}(sk_l, r)).$$

The signature algorithm uses the inputs an m-bit string message M, the secret key sk and the randomization elements r. Firstly it computes a w representation of the message M where $M\colon M = (M_1 \ldots M_{l1}), M_i \in \{0, \ldots, w-1\}$. Secondly, the algorithm computes the checksum

$$C = \sum_{i=1}^{l_1} (w - 1 - M_i)$$

and the base w representation $C = (C_1, \ldots., C_{l2})$. The base w of C is mostly $l_2$. B is a concatenation of the w representation of the elements of M and C. Lastly, the signature is computed in the following way:

$$\sigma = (\sigma_1, \ldots, \sigma_l) = (c_k^{b_1}(sk_1, r), \ldots, c_k^{b_l}(sk_l, r))$$

At least WOTS$^+$ has a verification algorithm. The algorithm needs as input the binary length m of the message M, the signature σ and the public key pk. First, it computes b as described above and then does a simple comparison which returns a simple true or false: (Hülsing, 2013)

$$pk = (pk_0, pk_1, \ldots, pk_l) = ((r, k), c_k^{w-1-b_1}(\sigma_1, r_{b1+1,w-1}), \ldots, c_k^{w-1-b_l}(\sigma_l, r_{bl+1,w-1}))$$

HORST can sing messages with a length of m. It needs the parameters k und $t = 2^\tau$ where $k\tau = m$. HORST improves Hash to Obtain Random Subset (HORS) and reduces the public key and signature size. HORST can be used to sign multiple messages contrary to WOTS. HORST consists of three algorithms (Bernstein et al., 2015):

- **Key Generation Algorithm:** The algorithm computes the internal secret key $sk = (sk_1, \ldots, sk_t)$ with the input of the seed $S \in \{0,1\}^n$ and the bitmasks $Q \in \{0,1\}^{2n*\log t}$. The tree is constructed with the bitmask and computes the leafs as $L_i = F(sk_i), i \in t - 1$.
- **Signature Algorithm:** The signature algorithm uses the message $M \in \{0,1\}^m$, the seed as above and the bitmask as above. Firstly it computes the secret key as above. Then M ist split into k strings with a length of log t. The signature σ consists of k blocks. Each block consists of $\sigma_i = (sk_{Mi}, Auth_{Mi})$. $Auth_{Mi}$ is the authentication path of the corresponding leaf. The least block consists of the whole binary tree of the level $\tau - x$. Additionally to the signature, the public key is generated.

- **Verification Algorithm:** It uses the input parameters M,Q as above and the signature $\sigma$. Firstly it computes $M_i$ like above. Secondly $N'_{yi,\tau-x}$ is computed with $y_i = [M_i/2^\tau - x]$ for $i \in [k-1]$. All nodes in the tree get computed and are compared to the nodes in $\sigma_k$. If all nodes match, it returns $\mathrm{ROOT}_0$. Otherwise, it returns fail.

The submitted version of SPHINCS, SPINCS+ uses Forest of Random Subsets (FORS) instead of HORST. This does not use a single tree. Instead is uses k trees with height of log t. The leaves are the result of the hashing of t elements with each a length of k. So the secret key is a concatenation of all nodes up to the root. The advantage is the usage of smaller parameters. (Hülsing, 2017)

SPINCS+ public key consists of two-n bit values. Firstly the root node of the top tree in the hypertree. Secondly of a public random seed. The secret key used two additional values. The SK.seed, which is used for the WOTS and FORS secret keys, and SK.prf for the randomized message digest. The signature now consists of a FORS signature of the randomized message digest. In addition, it also consists of a WOTS signature of this FORS public key. To check this WOTS signature public key, an authentication path and WOTS signatures are includes. This is used to reconstruct the public keys in the chains up to the root node of the SPINCS+ hypertree. (Bernstein et al., 2019)

### 5.4.5 Multivariate signature

Ding and Schmidt (2005) used the Oil and Vinegar signature scheme from Kipnis et al. (1999). This signature scheme is used multiple times and the resulting scheme is slightly longer than the document. This scheme is called Rainbow.

Firstly let start with some general definitions and constructions. This describes the Vinegar and Oil scheme. This is used for Rainbow afterward. S is a set in $\{1,2,3,\ldots,n\}$. $v_1,\ldots,v_u$ are u integers so that $0 < v1 < v2 < ..< vu = n$, and a another set of integers $S_l = \{1,2,\ldots,v_l\}$ where $l = 1,\ldots, u$. This results in $S_1 \subset S_2 \subset \cdots \subset S_4 = S$. The number of elements that are in S is $v_i$. Next, let $o_i = v_{i+1} - v_i,\ i = \{1,\ldots,u-1\}$ and let $O_i$ define as $O_i = S_{i+1} - S_i,\ i = \{1,\ldots,u-1\}$ : The polynomial $P_l$ defines the linear space of quadratic polynomial defined by the polynomials

$$\sum_{i\in O_l, j\in S_l} \alpha_{i,j} x_i x_j + \sum_{i,j\in S_l} \beta_{i,j} x_i x_j + \sum_{i\in S_{l+1}} \gamma_i x_i + n$$

This is a Vinegar and Oil polynomial. The $x_i, i \in O_l$ variables are the Oil variables and are also called the l-th layer Oil variable. On the other hand the $x_i, i \in S_l$ variables are the Vinegar variables or the l-th layer Vinegar variable. Any polynomial in $P_l$ is called an l-th layer Oil and Vinegar polynomial., precisely $P_l$, $l = 1, \ldots, u-1$ is a set of these polynomials. The relationship between Oil and Vinegar variables in the polynomials can be described as $S_{i+1} = \{S_i, O_i\}$. Rainbow uses a map F from $k^n$ to $k^{n-v1}$ so that each $\overline{F_l}$consists of $o_1$ random coefficients from $P_i$. So F has u-1 layers. The rainbow oft he variables look the following way

$$[x_1, \dots, x_{v1}]; \{x_{v1+1}, \dots, x_{v2}\}$$

$$[x_1, \dots, x_{v1}, x_{v1+1}, \dots, x_{v2}]; \{x_{v2+1}, \dots, x_{v3}\}$$

$$[x_1, \dots, x_{v1}, x_{v1+1}, \dots, x_{v2,} x_{v2+1}, \dots, x_{v3}]; \{x_{v3+1}, \dots, x_{v4}\}$$

$$\dots ; \dots$$

$$[x_1, \dots, \dots, . . . , \dots, \dots, \dots, x_{v_{u-1}}]; \{x_{v_{u-1}+1}, \dots, x_n\}.$$

The [] expressions represent the Vinegar variable and the expressions in the {} the Oil variables. Each of the rows in the above-illustrated F represents a layer in the Rainbow which has u-1 layers. Now the are two invertible affine maps $L_1$ and $L_2$ where $L_1$ is on the finite filed $k^{n-v1}$ and $L_2$ on the finite field $k^n$. The following construction is now finally used fort he Rainbow signature scheme:

$$\bar{F}(x_1, \dots, x_n) = L_1 \circ F \circ L_2(x_1, \dots, x_n).$$

The public key of Rainbow consists only out oft he field structure of k and n-$v_1$ polynomials of $\bar{F}$. The private key is constructed out of $L_1$, $L_2$ and F. The signing process consists of a few steps. The document to sign is an element of he finite field $k^{n-v1}$ with the structure $Y' = (y_1', \dots, y_{n-v1}'$ . This element must provide a solution to the equation $Y' = \bar{F}(x_1, \dots, x_n) = L_1 \circ F \circ L_2(x_1, \dots, x_n)$. Firstly the inverse of $L_1$ is applied which results in $F \circ L_2(x_1, \dots, x_n) = L_1^{-1} Y' = \bar{Y}'$ . The next step ist to invert $F$, for this the equation $F(x_1, \dots, x_n) = \bar{Y}' = (\bar{y}_1', \dots, \bar{y}_{n-v1}')$ needs to be solved first. Then the values of $x_1, \dots, x_{v1}$ are chosen randomly and are inserted in the first layer $o_1$ equations. Out of this a set of linear $o_1$ equations is produced. The solution provides the possibility to find the values $x_{o1+1}$, .... , $x_{xv2}$ , now all values for $x_i, i \in S_2$ are found. This values are used for the polynomials of the second layer. The same steps are repeated until a solution is found. If no solution is found another set of values for $x_1, \dots, x_{v1}$ is used and the same starts over again. At least the Inverse of $L_2$ is applied which results in the signature of $Y'$ and is denoted as $X' = (x_1', \dots, x_n')$. The verification oft he signature is simple and only the equation $\bar{F}(X') = Y'$ needs be checked. (Ding & Schmidt, 2005)

# 6 EXPERT INTERVIEWS

This chapter outlines the methodology and the results. Therefore expert interviews were chosen to gather the data and a summarizing content analysis was chosen to evaluate the data. This chapter contains also the results of the interviews and how they match with the hypotheses.

## 6.1 Methodology

The inspection of the hypotheses is based on expert interviews. The hypotheses are created with the theoretical input of the first chapters and additional assumptions which are posed in the subsection of each hypothesis. To verify these, the systematized expert interview from Bogner et al. (2014) is the used method, which is a guideline-driven interview. The goal is to examine the hypotheses with the knowledge and experience of the experts. Not only professional knowledge is of interest as well as experience and the estimation of the development in the field. The interviews consist of ten experts who answer seventeen questions. The duration of each interview should be around 45 minutes.

### 6.1.1 Collected Data

The interview is digitally recorded. The interviews will be selectively transcribed as the content of the interview only is the point of interest. The recorded interviews will be deleted after the final transcription and all personal data will be anonymized.

For organizational purposes and to categorize the experts, some additional data is collected besides the questions of the interview guide:

- Profession
- Experience in cryptography
- Experience with post-quantum cryptography

### 6.1.2 Experts

Experts can be identified by their positions, status, and their specific knowledge. This knowledge is divided into three dimensions, the operating knowledge, contextual knowledge, and interpretation knowledge. (Kaiser, 2014) But the definition of who is an expert must also consider the research question and is defined by the researcher. It depends on the targets and the representativeness of the expert. (Bogner et al., 2014)

In this paper, experts are people who have a background in cryptography and have been working and researching it for numerous years. Therefore, not only the scientific community is a point of interest but also the application. It is essential to acquire the view of researchers and also the view of people who use and implement the algorithms. Because of these two perspectives on the topic, it will be possible to detect differences in the assessment of the status of post-quantum cryptography and its development.

### 6.1.3 Interview Procedure

At the beginning the interviewee will be informed that the interview will be recorded, transcribed and analyzed. All personal information which may lead back to the interviewee will be anonymized.

The first question will be the the profession. Furthermore the experience in crytography and post-quantum cryptography will be noted. For each interview an individually printed guide is used. This will contain the other preliminary questions. All questions will be on the guide with room for notes. This will also include futher questions, depending on the answers of the interviewee. The final interview guide is added in appendix A.

### 6.1.4 Interview Evaluation

The evaluation of the interviews is based on the structured content analysis from Mayring (2015). For this paper some of the steps are adapted. The basic structure is illustrated in Figure 6-1.

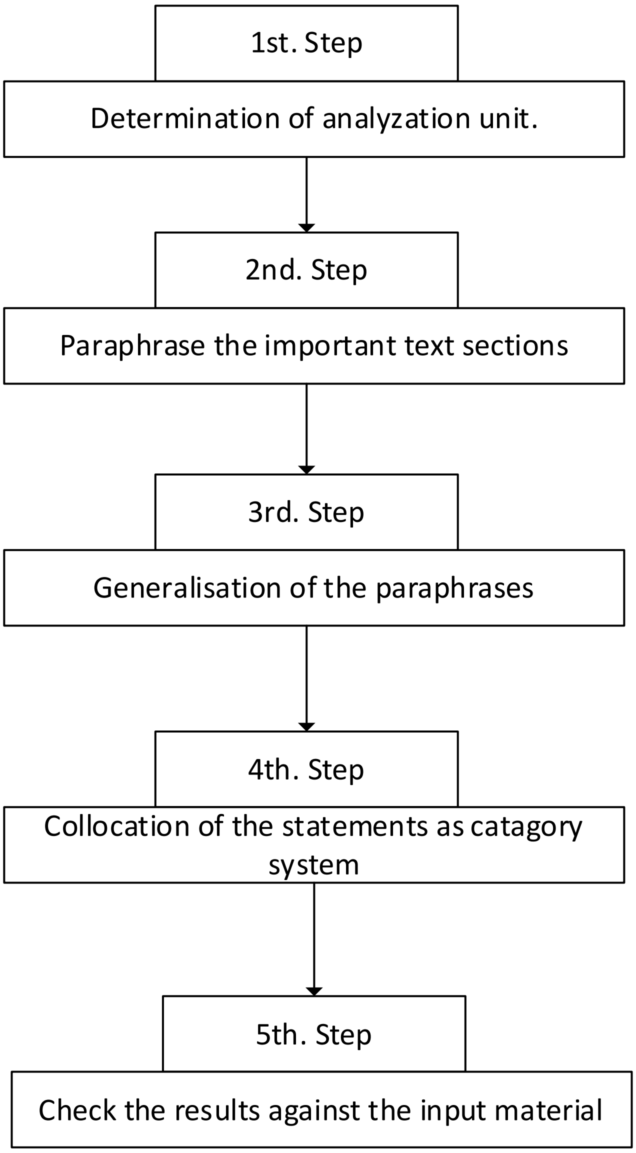

1st. Step
Determination of analyzation unit.
2nd. Step
Paraphrase the important text sections
3rd. Step
Generalisation of the paraphrases
4th. Step
Collocation of the statements as catagory system
5th. Step
Check the results against the input material

*Figure 6-1: Summarizing content analysis. (Adapted from Mayring, 2015)*

In this paper a shortened version of the summarizing content analysis from Mayring (2015) is used. The analysis unit is based on the questions of the interview guide. Each question is based on one hypothesis. The analysis unit is thereby a hypothesis. This means that paraphrases are categorized by interview and the question. Each question is summarized for each interview. When all interviews for a single question are summarized, then a summary over the summaries per question gives the answer to the question. Secondly, there is a quantitative question to each qualitative question. They have a scale from one to four. They are used to map the question into an integer and to make them more comparable. Because of the limited answers, the only evaluation of these question are the mean and the median value. They are used both to see if and how much the answers spread. At least the results are mapped to the thesis and it will be checked if the statements of the experts can confirm then H1 thesis or not.

## 6.2 Derived Hypotheses

The following hypotheses are based on the theory input and should answer the research question. All hypotheses are consists of the these H1 and the antithesis H0.

**Thesis one:** As described in section 4.2.1 RSA will be broken with a probability of 50 percent by 2031 and ECC will be broken before 2031 because of the shorter key sizes. But Gidney and Ekerå (2019) showed that it could be possible with 20 million qubits in eight hours. Based on this the following thesis is set up:

- **H1:** Currently used public key systems (RSA, Diffie-Hellmann, ECC) will be broken in the next 10 years.
- **H0:** Currently used public key systems will not be broken in the next 10 years.

**Thesis two:** Moody (2017) proposed that standardization is the first step to initiate the transition to post-quantum algorithms. With the standardization, the transition can be implemented in hard and software and improves usability.

- **H1:** A standard for post-quantum algorithms is required first before a migration can be done.
- **H0:** A standard for post-quantum algorithms is not required first before a migration can be done.

**Thesis three:** AES was introduced in 2001 but it took several years until it was actually used. Intel implemented AES in their processors in 2010. (Gueron, 2010) Over 150.000 TLS secured websites which are over 65 percent still support TLS 1.0. TLS 1.0 nowadays is 20 years old and its successor TLS 1.1 has been available for 13 years now. (Dierks & Rescorla, 2008; SSLLABS, 2019)

- **H1:** When standards for post-quantum algorithms are defined they will not replace current schemes within a few years.
- **H0:** When standards for post-quantum algorithms are defined they will replace current schemes within a few years.

**Thesis four:** Referring to section 4.2.2, symmetric schemes will be as affected as asymmetric because Groover algorithm is a known threat. Therefore, with AES-256, these schemes will still have a security level of 128-bit. Because of that the following thesis is posed:

- **H1:** No new standards are needed for symmetric schemes if the key sizes are at least 256-bit long.
- **H0:** New standards for symmetric schemes are needed.

<u>**Thesis five:**</u> Because hash algorithms suffer the same weakness as symmetric schemes, only the Grover algorithm can disturb them. In combination with the birthday paraoxon, current implementation of SHA-2 and SHA-3 with long input sizes are secure. (Mavroeidis et al., 2018)

- **H1:** Current hash algorithms (SHA-2, SHA-3) with long input values are suitable for a post-quantum world.
- **H0:** Current hash algorithms (SHA-2, SHA-3) with long input values are not suitable for a post-quantum world.

<u>**Thesis six:**</u> As described in section 5.2, quantum cryptography can be 100 percent safe but it is not compatible with common computers. For the wide usage, it will not be as important as post-quantum cryptography.

- **H1:** Quantum cryptography will not be as important as post-quantum cryptography for wide usage.
- **H0:** Quantum cryptography will be as important as post-quantum cryptography for wide usage.

<u>**Thesis seven:**</u> Personal data needs to be protected. One appropriate measure for protection is the encryption of personal data. The processing of special personal data such as health data is generally prohibited with some exceptions. (General Data Protection Regulation,2016) Depending on these exceptions and the idea that such information should be safe even when quantum computers are created, the concept would be that this data should be secured with post-quantum cryptography as soon as possible.

- **H1:** Sensitive data must be protected with post-quantum cryptography as soon as it is standardized.
- **H0:** Sensitive data must not be protected with post-quantum cryptography as soon as it is standardized.

<u>**Thesis eight:**</u> The focus on crypto-agility is exceptionally important. Because post-quantum algorithms are not ready yet, organizations must be ready to implement them in 10 years from now. So it is compelling to consider this urgency in the design of new software and protocols. (Chen et al., 2016) Housley (2015) defines guidelines for crypto-agility. Protocols such as TLS support this claim, but the effortless migration in every field must be taken considered.

- **H1:** If crypto-agility is taken into consideration today, the migration will be quicker.
- **H0:** If crypto-agility is taken into consideration today, the migration will not be quicker.

**Thesis nine:** Based on the framework from Wang and Wulf (1997), which describes that the security level of a system contains from multiple factors, it possible to assume that the cryptographic strength is based on the weakest link in the chain. In a system where multiple systems are involved in between information generation and delivery, the cryptographic level is only as strong as the weakest link. If all systems in that chain use post-quantum cryptography but one uses classic ECC, the system security is reduced to ECC and cannot be considered to be quantum safe as described in Mavroeidis et al. (2018).

- **H1:** If not all systems in the chain use post quantum cryptography, the security of the system will be reduced.
- **H0:** If not all systems in the chain use post quantum cryptography, the security of the system will not be reduced.

## 6.3 Data Evaluation

In this section, the data from the interviews is evaluated. It is divided into two subsections. One for the qualitative analysis and the other for the quantitative analysis.

### 6.3.1 Qualitative Analysis

The qualitative analysis is shown in Table 6-1 as in Mayring (2015). Their method has been slightly adapted as explained in subsection 6.1.4.

| Category | | Phrase | Generalization |
|---|---|---|---|
| Interview | Question | | |
| 1 | 1 | The problem within the whole subject matter is that there are quantum computer skeptics who have compelling arguments. They state that these systems will never scale sufficiently enough so that they can break current schemes with satisfactory security parameters. | Quantum computers will never scale enough to break current schemes. |
| 1 | 1 | But there are others such as Michele Mosca, who has adapted this idea repeatedly and claimsthat it will be possible in one or two decades. | Some say that it will be possible in one or two decades. |
| 1 | 1 | But if we are more pessimistic, we could say that we have a problem in the next two decades. | We have a problem in the next two decades. |
| Summary | | Probably quantum computers will never break current schemes, but it is likely to happen in the next 20 years. | |
| 2 | 1 | It is said that you need two-time n qubit to factorize RSA. Therefore, for a 2000 bit RSA, you would need 4000 qubits. But the problem is not the number of qubits. The problem is error correction. | The problem is error correction. |
| 2 | 1 | I would say between 10 and 20 years. I am quite sure about 20 years and in 10 years, it is probably around 50 percent. | It will happen in the next 10 to 20 years. |
| Summary | | It will happen in the next 20 years. The main problem is the error correction. | |
| 3 | 1 | That is a major point for discussion in the community. I personally think somewhere between 2030 and 2040. So, in 10 to 20 years. | No consensus in the community but in the next 10 to 20 years. |
| 3 | 1 | The main problem is the engineering. Because we know how it works from the physics side, the question is, are we able to build it because there are so many error sources. | The problem is how to build it. |
| Summary | | It will happen in the next 10 to 20 years, the problem is how to build it. | |
| 4 | 1 | This is not easy to say. In fact, we don't know it exactly. And there is no general opinion in the community. It depends a lot on whom you ask. I think somewhere between the next 10 to 20 years. | No consensus, but in the next 10 to 20 years. |
| 4 | 1 | It really depends on progress in engineering to scale the quantum-computers to such a large number of qubits and the improvement in the error detection | Challenges in engineering and error detection. |
| Summary | | Will happen in the next 10 to 20 years, the problem is how to build it. | |

| | | | |
|---|---|---|---|
| **5** | **1** | The opinion in the community varies very much. From it will happen with a chance of 25 percent by 2026 up to it will never happen | Opinions vayries between 2026 and never. |
| **5** | **1** | But people should be prepared within 10 years. | Be prepared within 10 years. |
| **Summary** | | Timespan between 6 years and never, but be prepared in 10 years. | |
| **Summary question 1** | | **Most likely in the next 10 to 20 years, but error detection and engineering, such computers is a huge challenge.** | |
| **1** | **2** | I think that the result of this and from the NIST will be the standard for what the industry will do, like in the other competitions in the past. NIST has hosted the hashing competition which resulted in SHA-1 and SHA-3, and hosted the symmetric encryption where AES was created. | NIST winner will be the standard like AES and SHA. |
| **1** | **2** | So, it is very likely that the winners of this competition will be the de facto standard for the industry. There are other standardization organizations like iso, which have workgroups in this field, but they will orient themself on the NIST results. | Other standardization organizations will orient on the NIST results. |
| **1** | **2** | They also standardize some questionable ideas because of the influence of the industry. There are some big players like IBM who want to create facts. They say they have quantum-safe hard drive encryption, which was probably made by an intern because it is not so difficult. | Some will be standardized because of industry influence. |
| **1** | **2** | And they also have their own schemes in the competition like all big players, and they can make products besides the NIST results. But most will use the NIST results because they are well-reviewed and tested by the community. | Big companies can use their own schemes, but most will use the reviewed and tested NIST standards. |
| **1** | **2** | But we should not forget about China, which does not care about the NIST and has its own competition and research. We do not know if they are already finished, because they work quite fast, but the Western world views this very critical. Not many Western corporations would use a Chinese standard. | China does its own research. Not clear what status they have reached. But for most outside of China, these will not be that important. |
| **Summary** | | NIST winner will set the standard, and other standardization organizations will abide by it. Will be used by most,but china works on its own standards. | |

| | | | |
|---|---|---|---|
| **2** | **2** | Huge. I attended the NST conference this year. And without them, there would be no progress at all. The problem was shown by Shor in 1984, and the first cryptographers reacted in 2004 or 2005, and now we have 2020, and we still have no standard. So, it takes a long time, and without the NIST, nothing would happen. | Shor exists since 1984, in cryptography known since early 2000, without the NIST nothing would happen. |
| **2** | **2** | But many browsers and TLS 1.3 have some post-quantum methods implemented optional. So, it exists but is not used because the client and server don't choose a post-quantum mechanism. So, it is possible today. | It is possible and implemented optionally in TLS, but the server and client currently don't choose it . |
| **2** | **2** | Because of the big IT companies, it will be deployed quickly with their huge market share.  And in all tests like performance tests the big ones such as Microsoft, Cloudflare, Google, and so on are involved. So, it is not the implementation. They wait for the proof that they are secure. | Big IT companies are involved. They wait for the security proof. |
| **Summary** | | Problems have been known for a long time, and without the NIST, nothing would happen. It is available today, and the big IT companies are involved. They only wait for the security proofs. | |
| **3** | **2** | What wins this competition will be used. In the US, they will blindly use NIST standards. In cryptography and in the US anything  that is standardized is used. And this result will have an impact on Europe too. They will use the results of the NIST as well. | The NIST winner will be used. In the US and also in Europe. |
| **3** | **2** | In China, there is also a post-quantum competition, and they have other candidates. The basic concepts are the same, but you do not notice it much when you do not speak Chinese because all documents are only available in Chinese. | China has its own competition. Not much information about it. |
| **Summary** | | NIST winner will be used, and China has its own competition. | |
| **4** | **2** | When you look at SHA and AES, both of them are the winner of a NIST competition. And both are used in extremely high frequency and are the de facto standards in the industry. And so I think the winners of the NIST competition will be used. | The NIST winners will be used. |
| **5** | **2** | I would say if it is standardized, it will be used. We have seen this with the winners of the last competitions of the NIST, which resulted in SHA and AES. There are a lot of competitions today, but the NIST competition is the most influential one. What they will standardize is what will be used in the future. | What the NIST standardizes will be used like AES and SHA in the past. |

| **Summary question 2** | | **NIST winner will be used. China has its own competition. Not much information about it.** | |
|---|---|---|---|
| **1** | **4** | I believe that the big companies which also have candidates in the NIST competition are working on products already. | The big tech companies are already working on it. |
| **1** | **4** | The big ones will push their candidates, and they will immediately be ready to sell because it is a huge market. | The big tech companies push their candidates and will be ready immediately. |
| **1** | **4** | Yes, the big ones will have it ready very fast because it is a good selling point. There is a lot of fear in the industry, everyone is scared and nobody knows what will happen. So, the big ones are ready to start because there is a lot of money that can be made with it. | The big ones will be fast. Because of the fear, there is a lot of money to be made. |
| **Summary** | | The big tech companies are working on it and push their candidates. They will be fast because of the fear in the industry and there is a lot of monetary gain in it as well. | |
| **2** | **4** | If I look at the past and how it worked back then, yes, I would say 40 years. If you look at 3DES, which is still used and contained in used standards. So, the complete change will take that long. | The complete change will take 40 years. |
| **2** | **4** | Cryptography is used a lot in the industry, and there it is difficult the change the crypto. Therefore, it needs so much downward compatibility. | Compatibility is a problem. |
| **2** | **4** | It takes so long to pervade the market. Not only a few implementations in browsers that work quite fast. But a complete penetration is necessary so that you don't have to downgrade, which would take around 30 years. Or DNSSec is another example. This also took almost 30 years. It takes unbelievably long. Therefore, we need to start as soon as possible. | Broad implementation aside browsers will take that long. |
| **Summary** | | Board implementation aside browsers will take 40 years because of lacking compatibility. | |
| **3** | **4** | This will take a while. Because of this, it is good that the post-quantum competition is now, although it takes longer than it should. The problem is you have to balance security and performance. You want a secure algorithm that is reviewed and tested for a while, and you choose candidates who are good performers but are also safe to use, even in 20 years. | It takes a while because candidates should be safe to use even in 20 years. |
| **3** | **4** | The industry will need 10 to 15 years for the migration. The competition must be finished by 2025. | It takes 10 to 15 years after completed standards. |

| | | | |
|---|---|---|---|
| **3** | **4** | This is the most important thing. That the internet infrastructure is secured with these algorithms. And with TLS, it will be easy. Because most of the big web server vendors are open source and use OpenSSL, and when OpenSSL implements it, then it will be no problem. The current TLS 1.3 version has not defined any post-quantum algorithm. There will be a newer version or an appendix. | Internet infrastructure must be secured. TLS will implement it with a new version or an appendix to TLS 1.3 |
| **Summary** | | Final standards will take a while for security reasons. Afterward, it will take 10 to 15 years for migration. Internet infrastructure is the most important. | |
| **4** | **4** | As I said concerning the internet, we will see it quickly and also in the services provided by the big tech companies. There, I would say in less than two years. But for the rest. Quite hard to tell. They are still using things like DES. So, I would say something around 20 years for extensive usage. | It will be ready for the internet first and quickly, a broad migration will take 20 years. |
| **5** | **4** | That really depends. For existing systems, it can take a while, especially for embedded systems that you cannot update /upgrade and may have a long life cycle. There I would say it will take at least 20 years. | For existing systems, it will take 20 years. |
| **5** | **4** | For new products/systems, they will use it quickly. I would say there will be a standard in around five years after completed standards. And the internet and standard software like browsers and web servers, which you can update quickly and so on, will be developing rapidly, it will take less than five years. | New systems and the internet will require less than five years. |
| **Summary** | | Legacy systems will need 20 years, new systems, and the internet only five years. | |
| **Summary question 4** | | **It will take between 10 and 40 years, but most probably around 20 years for current systems. New systems and the internet will use it in approximately five years.** | |
| **1** | **6** | As far as we know now, AES-256 will be enough. It doesn't look like if something is happening in the quantum symmetric cryptoanalysis. There are some results, but they depend on special modes that have weaknesses in the quantum context. | Standard AES-256 is safe. |
| **2** | **6** | No problem at all. This is because almost all symmetric ciphers create a keyspace. Okay, a hash is no encryption, but they also create such a keyspace. So, with that, you can compute all keys. They are often called rainbow tables. They are huge. This is equivalent to a search in an unsorted space, a linear search. The complexity is equal to n. So, in the real world, it is faster, as the birthday paradox, and you can say it is the square root of n. | No problem. Creates a huge keyspace and equals an unsorted search with speed square root n. |

| | | | |
|---|---|---|---|
| **2** | **6** | The only know algorithm is the Grover algorithm, which brings a significant speedup. So, AES-128 would not be safe anymore because it has only 64 bits of security. The only thing to do in symmetric ciphers is to bring them to a security equivalent of 256 bits. The double key length is no problem because they are quite fast. | Only Grover is known. With double key lengths, there is no problem. AES-128 is not safe enough anymore. |
| **2** | **6** | It is more than enough because the search space is so vast. And the Grover algorithm is a maximum. It is not possible to be faster. Compared to the Shor algorithm. It is not proved that no faster algorithm can exist, and maybe a quicker one will be found someday. | AES-256 is more than enough. Grover is a maximum speedup. |
| **Summary** | | Grover is a maximum speedup with square root n. AES-128 is not safe, AES-256 is safe enough. | |
| **3** | **6** | The big problem is that you cannot say it with 100 percent accuracy. But there are only a few algorithms known today which can use the quantum computer to achive an exponential speedup. There is the Shor algorithm, which affects the asymmetric ciphers. Because of the exponential speedup, you cannot simply increase the key size because they would be huge. Symmetric chippers, on the other hand, do not have these algebraic structures because they use confusion and diffusion, and for this, only the Grover algorithm known, which allows finding one out of n elements in square root n time. | It cannot be said with 100 percent accuracy, but no exponential speedup known. Only Grover with square root n. |
| **3** | **6** | So, AES-256 will be fine, and some analysis shows that even AES-128 does not drop to a 64-bit level but only to 85-bit. The reason is that the Grover algorithm has many constants. IBM and Microsoft have developed programming languages that are made for quantum computers. You can program with it today but not run it on a quantum computer, only on a simulation. And they have resource estimators on board. For AES-128, they say around 80 bits, which is a big step from 64 bit. | AES-256 is acceptable. But AES-128 might only drop to 80bits because of constants in Grover. |
| **Summary** | | Not 100 percent sure, but AES-256 is safe, and AES-128 might not be as weak as feared. | |
| **4** | **6** | At the moment, the best algorithm known for symmetric AES is Grover. But this gives only a speedup of square root n. So, in the worst case, we end up with a | Only Grover is known with square root n speedup. But faster algorithms could exist. |

| | | | |
|---|---|---|---|
| | | security level of 128 bit. And this is strong enough. But we do not know if there could be a faster algorithm. | |
| **4** | **6** | Today, I would say we don't need to worry about AES-256. It will be sufficient. AES-128 should not be used anymore, because it would result in a security level of 64 bit, which is even not sufficient today. | AES-256 is safe, AES-128 should not be used anymore. |
| **Summary** | | Grover is the best-known speedup. Faster ones could exist. AES-256 is safe, AES128 should not be used anymore. | |
| **Summary question 6** | | Symmetric ciphers are not as vulnerable as asymmetric ciphers. Only Grover is known to have a square root n speedup. AES-256 is safe. AES-128 should not be used anymore. | |
| **1** | **8** | It is the same as with symmetric block ciphers. The only known speed up is Groover, and this gives a square root speed up. In symmetric ciphers, you double the key size, and in hashing algorithms, you double the output size. | Like symmetric ciphers just double the output length. |
| **1** | **8** | Current algorithms such as SHA-3 are parameterizable for different output lengths. So, it does not make a very noticeable difference if you do not use algorithms like SHA-1. Use SHA-3 with long outputs, and everything is fine. | SHA-3 with proper output length is good enough. |
| **Summary** | | SHA-3 with long output is accceptable. | |
| **2** | **8** | It is 1:1 like symmetric ciphers. Nothing changes here. It is no encryption but a mapping. But it changes nothing. Instead of 2^128 keys are there 2^128 hashes, and we have the same problem of an unsorted search. | Like symmetric ciphers. Double output length. |
| **3** | **8** | Hashing algorithms are quite similar to block ciphers internal. They have the same attack, the Groover search for preimages, and for hash algorithms, there is another security property as well, the collision resistance. For this, a quantum collision finding algorithm, which decreases the time to find collisions from 2^(n/2) to 2^(n/3) exists, but this algorithm also has many hidden constants. | Similar to symmetric ciphers. Because of collision resistance it is weaker. |
| **3** | **8** | One famous scientist said that this algorithm will never beat the classic algorithm because the additional complexity is so much harder to achieve in the quantum world. When you double the output length, you also have no problem with hashing. | Will maybe never happen. The double output length is enough. |
| **Summary** | | Similar to symmetric ciphers, double output length is enough. | |

| | | | |
|---|---|---|---|
| **4** | **8** | Good hashing algorithms such as SHA are facing the same problems as symmetric ciphers. And there is also the Groover algorithm, the only known algorithm. But hashes also need to be collision-resistant. And there is the birthday paradox,which reduces the security level to cube root n. But in general, it is enough to double the output. | Similar to symmetric ciphers. Birthday paradox reduces to cube root n. The double output length is enough. |
| **5** | **8** | They work like block ciphers. So they are only vulnerable to the Groover algorithm. The only difference here is the fact that they need to be collision-resistant. And there exists the birthday paradox which reduces the security level of hashes a bit more than symmetric ciphers. You now have a speedup of the cubic root. | Similar to symmetric ciphers. Birthday paradox reduces to cube root n. |
| **Summary question 8** | | With cube root n hashes are weaker than symmetric ciphers but in general, the same as symmetric ciphers. Double output length solves the problem. | |
| **1** | **10** | My opinion is it should be named quantum key distribution and not quantum cryptography because it offers very limited functionality, and it is nothing else than a key exchange. And this is no alternative to post-quantum cryptography. Because in post-quantum cryptography, the focus lies on asymmetric methods that offer encryption and digital signatures. That is something you cannot achieve with QKD. | QKE offers limited function, no alternative to post-quantum cryptography. Only key exchange. |
| **1** | **10** | QKD can exchange a key for a symmetric cipher or a MAC, but that is all, and the problem we have today, which will limit the usage of QKD, is that pre-shared key material is required. Because you cannot authenticate the channels. You can exchange the keys, but you don't know if you talk with the right one. So, it is useless to have perfect security against an eavesdropper when I am talking with the incorrect one. That is the reason why it scales poorly and the hardware is very expensive. | It can exchange the key but cannot authenticate the channel. Perfect security against an eavesdropper, but could be the wrong communication partner. Scales poorly and requires expensive hardware. |
| **1** | **10** | ... but it is a technology that has its niche in sectors such as the government, military. There, it is useful and can be used but it will not revolutionize the internet. What could be done is a hybrid. You use a post-quantum signature over a QKD channel. Therefore, you have the security of the QKD channel but also know with whom you are talking. Alongside some military satellite, etc., it will not be that important. | Limited usage for sectors such as the government, military, etc.. Hybrid of the post-quantum signature over the QKD channel could be done. Will not revolutionize the internet. |
| **Summary** | | No alternative to post-quantum cryptography. Perfect security but no authentication. Limited usage for military or governments, etc. Not for internet usage. | |

| | | | |
|---|---|---|---|
| **2** | **10** | That is quite hard to say. The protocol BB84 is quite old. But there is still one problem. How do I know with whom I exchanged the key. There is no authentication. So, there is a need for an exchange of a authentication key, which brings you back to the key exchange problem. You cannot securely establish a secure channel. Therefore, you would need to exchange the key in a sort of USB drive or something similar. This would be quite expensive. | No authentication is possible. Therefore, an authentication key must be exchanged. Expensive. |
| **2** | **10** | It will spread. I would say three, but it will be more of a sham. You still need to exchange the authentication key. Therefore, they will build in some type of hardware token for this. But then I could use fixed storage with perfect randomized numbers on it. It is questionable, but it will happen because of marketing. They invested a lot of money in it, and there will be devices. It is quite interesting scientifically but does not help in practice. | We will see it. The authentication key will be stored in a hardware token. Improvements are questionable but will happen for marketing reasons. |
| **Summary** | | No authentication and authentication keys are required. Will be solved with hardware tokens. Expensive and benefits are questionable. Rather for marketing purposes. | |
| **3** | **10** | I would say no, I am a bit skeptical, especially on smartphones. Most methods that I know use entangled qubits that are sent and are measured by the recipient. Because of the no-cloning theorem, you have the guarantee that nobody has listened in. And you use quantum processes to ensure the generated key is random. One problem is that there is not much research in this field. They trust the classic key exchange at the moment. In the future, they will trust the post-quantum key exchange. It is better researched and less complicated. For the quantum key exchange, you need special devices, which are very expansive. | Skeptical if it has a future. Perfect security and randomness. Not as widely explored as post-quantum key exchange. Requirement is special hardware, which is expensive. |
| **3** | **10** | Yes, this could be more realistic. For example, for a company that wants to exchange keys between sites and wants the highest security. But also, then they will combine it with a classic key exchange and combine the security of both. I would not trust that this exits currently with using traditional methods. | It may be suitable for special purposes. But also requires a combination with a classic key exchange. |
| **Summary** | | The future is unknown. Perfect security and randomness. Expensive hardware required and only suitable for special purposes. | |
| **4** | **10** | Hard to say. We will see how it develops. But I think it will be more for marketing than for measurable other advantages. You need special hardware for it, and it | It will be used for marketing purposes. Few real advantages but rather disadvantages like no authentication or the |

| | | | |
|---|---|---|---|
| | | also has some disadvantages such as no possible authentication. There could be useful for some special purposes, but I do not see a real advantage for regular use. So yes, it will be there as a selling point, but without a real advantage. | requirements for special hardware. |
| **5** | **10** | You have some advantages but also some immense drawbacks. The most significant advantage is that it is 100 percent secure. Because you do not use math functions for it but rather the laws of quantum physics. But there are some significant drawbacks. First, you need special equipment for it. And you cannot authenticate. You do not know with whom you re communicating. So, it could be used for some special purposes, but a general usage and the restrictions are too big. | Is 100 percent secure because it is based on the laws of quantum mechanics. But immense drawback like the requirement for special hardware or no authentication. It will be used for special purposes but not for general usage. |
| **Summary question 10** | | Uncertain future. Perfect security but no authentication possible. Authentication must be done with its own keys. Special, expensive hardware required. It will be used for special purposes but not in general. | |
| **1** | **12** | It can be useful when you have data where you want to have long-term security. But you have to evaluate what is the attacking model. Against what do you want to be protected? If you are in a closed network, there is the question if there is not a more efficient attack that will not break the crypto and steal the data. I think it will be relevant when you transmit your data over networks you do not control. | Useful for long term security, longer than 20 years. Question if there is not a more efficient attack model? When transported over the internet, usage in the present is recommended. |
| **1** | **12** | So, when you transport information over the internet which is sensible and must be secure, also in 20 years, then you should evaluate it. But most companies will not be affected by this. But it will affect companies in the health sector where you have sensible data. But when you look at how they work today, I do not think they will be the early adopters. | Most companies are not affected by it. The health industry could be, but they will not be the early adopters. |
| **Summary** | | When data is transported over public networks and must be secure in 20 years as well, then it should be considered. Most companies will not be affected by it. | |
| **2** | **12** | Yes, and some people are beginning with it. For you and me, the end-user, it is not essential today. Otherwise, it would be implemented in the Internet. But this would cause a performance impact. It is essential for large organizations, such as countries. There it must be done nowadays. It is a difference if u can see why someone got president after 15 years, as an example. | Not necessary for individuals now. But for large organizations and governments. |

| | | | |
|---|---|---|---|
| **2** | **12** | No, they are not so important (health data). So, no one will be interested in normal people like us. But for the military and so it is important. In general, it is called critical infrastructure and government in general. They should use it. They should start immediately. | Even health data is not so important. Only governments and critical infrastructure should start immediately. |
| **Summary** | | Individuals and even health data are not so important. They must not act now. But governments and critical infrastructure should start now. | |
| **3** | **12** | You need to classify on your own as a company or person how long this data is essential and needs to be secure. Most personal data will not sensitive anymore in 20 years. But for military data and so on, you need long term security. | You have to classify your data if they are still so important in 20 years. Most data is not that important. |
| **3** | **12** | And one big problem is that organizations such as the NSA and other state actors can capture and store some of the traffic today, and in 40 years, when quantum computers are fast enough, they can decrypt the whole history. But they can only store information of some people, and maybe a lot of them are already dead in 30 years. So, you really need to consider which of your data is important. | One problem is that organizations such as the NSA can store the traffic today and encrypt it in 40 years with a proper quantum computer. But some people will be dead by then. Therefore, you must consider which data is important. |
| **Summary** | | Most data besides military and governmental data is not so important because the data must continue to be confidential in 20 years. | |
| **4** | **12** | Definitely. But it depends. In 20 years, it should be used everywhere. But now, that is the question. You have to classify the data, and when they are still confident in 30 years, you should start with it now. But must data is not so sensitive. The kind of ordinary data that most people have are not relevant in 30 years from now. But data in the governmental field or military data can be rather sensitive. They should use it now, and I think they do. | If data needs to be confidential in 20 years, then start now. Otherwise, it is not necessary. Affects sensitive data from military and government. |
| **5** | **12** | For the most types of data, you will not need it now or in the near future. And in the distant future, it will be standard for everything. But for specific kinds of information, it can be essential, even today. This could be financial data, some sort of health data, or government and military. The question you must ask is, must this data be secure, even in 30 years. If yes, then you should start with post-quantum cryptography today. The problem is that organizations such as the NSA are storing data. And when they have a big quantum computer, they can decrypt all the stored data afterwards. | Not useful for most types of data. Only if the data must be confidential in 30 years. For sensitive data such as health, financial, government, or military data. Stored traffic can be decrypted afterwards. |

| | | | |
|---|---|---|---|
| **Summary question 12** | | Not necessary for most types of data currently. But for the government and critical infrastructure, it could be. When data must be confidential in 20 to 30 years, you must start now. | |
| **1** | **14** | Most crypto libraries that are used in software today are written in a way so that you have crypto agility. So, it depends only on the interface. Well written crypto software does not allow the user to look into it, and it should be foolproof. It should be easily parameterizable, and it should make no difference which method is used. You can still have some compatibility issues. Let's say we have public key encryption with signatures. Then you need a PKI. So, you relay on the features they offer. If you cannot control this, you can have a problem here. | With well written libraries are not a significant problem. Simply change the parameters. But when a PKI is required, you can have some troubles if the features are not supported. Crypto agility is achieved by them. |
| **1** | **14** | I would not say that there is no problem, but it is realistic if you want to do it. With external dependencies, it will not work. Even when I only want to have email security, and I use Exchange, then I will have no chance if Microsoft does not support me. As soon I want to exchange data with some else, it will be difficult, so I think it will take a while. There are some standards available. Some RFCs from the IETF for stateful signatures are available. But you need a use case for stateful signatures. And even then you depend on external libraries because to implement cryptography independently is rather complicated and should not be done. | With external dependencies, it will probably not work. Exchange information is difficult. And you always depend on the crypto library because you should not do it yourself. |
| **Summary** | | Always depending on the libraries but not so a big problem with good ones. But when you cannot control everything, it will be quite hard. But crypto agility is achieved with good libraries. | |
| **2** | **14** | Firstly, yes, you can use it. We have at least one really decent implementation ineach category. Places where you can download a library, you can actually trust. They are an open source, and you can analyze the source code and implement it in the way you need it. Even Bernstein said in 2018 that there are some you can use today. But it does not affect the encryption. The encryption itself is always done with symmetric ciphers. You use it for the key exchange or for digital signatures. It would be possible, but it is way too slow. It is the same with RSA. A file is never encrypted with RSA. | It is possible to use it today. Good libraries are out there with proper implementation but only affect only the key exchange. The encryption itself is always done with a symmetric cipher. |
| **3** | **14** | It depends. When you want to use the hybrid versions today, you have to pay attention to it. But if you want to change it | If a hybrid method is used, you need to pay attention to it. If in the future, only the change to |

| | | | |
|---|---|---|---|
| | | in the future to a post-quantum method, then you will have no problem at all because the API is still the same. A proper implementation has encryption, a description and a key generation method and those are the same in the post-quantum world. The only thing you might pay attention to is the size of the ciphertexts and keys because they will change. And depending on the method, they can differ enormously. Some encryptions have keys with several megabytes and small ciphertexts, some like RSA where keys and ciphertexts have several kilobytes and others with minor keys and huge ciphertexts, and this is also true for signatures. | post-quantum should be done, you must not. Only change the API call. But ciphertexts and key lengths can differ enormously. That needs to be considered. |
| **4** | **14** | It depends a bit. In general, I would say you must not pay much attention to post-quantum cryptography in particular. When you use cryptography right and implement it properly in your software, it will not be a big problem because you will use the standard libraries. And there you use the API. So, you only have to change the API call, and the rest will work as usual. But if you did a bad job with the implementation of cryptography in the first place, or you want to use hybrid encryption, then you must pay attention. | You will only have to pay attention if you have implemented cryptography poorly or use hybrid methods. Otherwise, just change the API call. |
| **5** | **14** | Yes, you can consider it in your projects. The first implementations already exist and can be used. The more critical job is to implement cryptography properly. Because then you will have no problems to switch the used method. You only change the parameters you send to the API. You should not implement anything on your own. Because you will also have a lot of bugs in it, and then it is useless. It is no problem to use the candidates in the competition if one of them is implemented in a crypto library. They may not be in the standard later, but they work too. | Most important is to implement cryptography properly, then you only have to change the API call. Implementations exist and can be used today. Do not implement anything by yourself. |
| **Summary question 14** | | With a proper crypto implementation and good libraries, you have achieved crypto agility. You must only change the API call. You must pay attention if hybrid methods should be used. Never implement anything by yourself. | |
| **1** | **16** | That is really a big problem. I have no idea. When you look at the web, most of the attacks in the last years only happened because of legacy reasons. Most attacks on TLS used the fact that TLS implementations need to be backward compatible. And a lot of them where simple downgrading attacks. The patterns over the years are similar because the people do not update their systems. And when you do not control this, what should you do? | It is a big problem, and there is no known solution. Most attacks in the past used the backward compatibility. But you cannot control the user's systems. |

| | | | |
|---|---|---|---|
| **2** | **16** | Yeah it is not possible. In cryptography, you always have to downgrade to the weakest link. Except the protocol has a mechanism implemented that communication is not possible if a minimum standard is not supported. But this does not happen very often. You usually have a downgrade attack where you enforce a weak cipher which you can break. | Not possible to solve. In crypto, you are always downgraded to the weakest link. Minimum standards would help, but it does not happen often. Downgrade attacks will happen. |
| **3** | **16** | Yes, we have this problem even today. Many systems use DES or some homebrew encryption. Yes, but the ones in TLS 1.2 are pretty good compared to what some companies are using. For example, Smartcard vendors have built their own encryption, which can be broken in one minute. | DES or homebrew encryptions are still used and easy to break. Also, some Smartcards use own weak encryption. |
| **3** | **16** | You have to pay attention, and you will downgrade to the weakest link. As far as I know, will most legacy systems be disconnected from the internet, and you have an internal legacy system. And if your threat model says that is ok, then it is ok. But if it is on the internet, you have a problem. Because in 10 or 20 years, you can send the RSA key to a cloud service and can break the encryption. Because of the availability of quantum computing in the cloud, everybody can do it. Quantum computing will not be restricted to state actors and big companies. It will be accessible to everyone. | Disconnect legacy systems from the internet, if possible. Otherwise, you have a problem. In 10 to 20 years, quantum cloud services will be accessible for everyone, and RSA will not pose a big problem. |
| **Summary** | | Legacy systems are problematic, quantum cloud services will be available in 10 to 20 years for everyone, and RSA will be no challenge then. DES or homebrew encryption is still used. Problematic. | |
| **4** | **16** | They are the same effects as today. All downgrade attacks are based on the fact that you have to deal with legacy systems. Therefore, you get downgraded to the weakest link of the chain. Shure, you can do this. But it is the question if you can or want to exclude some people from your services. Because you want as many people as possible to be able to use it. Because of that, a lot of web servers still support TLS 1.0, which is 20 years old. Realistically you may be able to enforce post-quantum cryptography in 20 years as the earliest point in time. | Downgrade attacks will happen and are also a problem today. You can exclude legacy systems but will not happen. It can be enforced in 20 years. |
| **5** | **16** | We have the same problem now, and we have had it for decades. And there is no real solution. A lot of attacks that have happened in the past or happen today are exploiting the downward compatibility. When someone does not support a standard, a weaker one is used. Most TLS attacks also worked that way. But most times, you do not want to exclude the | Downgrade attacks have been done for decades now and they will continue. Exclusion of legacy systems will not happen. If possible, isolate them, then the risk is reduced. |

| | | | |
|---|---|---|---|
| | | ones which do not support modern standards. And sometimes, you have devices that do not get any updates anymore. This often happens in embedded systems or industrial facilities. But some of these systems can be isolated. Then the risk is reduced. | |
| **Summary question 16** | | Downgrade attacks will happen. Enforcement will be possible in 20 years, but then there are also cloud quantum computer services accessible for everyone, and RSA will be no problem. If possible, legacy systems should be isolated. | |

*Table 6-1: Analysis of the qualitative questions based on the summarizing content analysis from Mayring (2015).*

### 6.3.2 Quantitative evaluation

Besides the open questions, every question has an additional question which maps the previous question on a scale from one to four. The questions can be found in the interview guide in appendix A.

| Question 1 | | | | | |
|---|---|---|---|---|---|
| **Interview** | **1** | **2** | **3** | **4** | **5** |
| **Answer** | 20 | 20 | 15 | 15 | 10 |

*Table 6-2: Evaluation question one*

The opinions of the experts differ here. They stated the timespan until public key encryption can be broken lies between ten and twenty years. But the mean and median in Table 6-2 are located in the middle with sixteen years for the mean and fifteen years for the median.

| Question 3 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (not likely)** | **2** | **3** | **4 (likely)** |
| **Interview 1** | | x | | |
| **Interview 2** | | | | x |
| **Interview 3** | | | | x |
| **Interview 4** | x* | | | x |
| **Interview 5** | | | | x |

*Table 6-3: Evaluation question three. * usage in the whole industry sector*

Most experts believed that implementations will occur before we have final standards. Only one expert said it is rather unlikely, and another one said it will depend on the sector. For industries besides the technology sector, it is not likely. This can also be observed with the median and mean. The mean value is a bit lower with a value of three point six compared to the median with a value of four, however, it also shows that it seems likely.

| Question 5 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (does not apply)** | **2** | **3** | **4 (applies)** |
| **Interview 1** | X | | | |
| **Interview 2** | X | | | |
| **Interview 3** | X | | | |
| **Interview 4** | | x | | |
| **Interview 5** | | x | | |

*Table 6-4: Evaluation question fife*

All of the interviewd experts agree that no new IT systems will be required. This can also be seen with the median and mean value. The median has a value of one and the mean value is a bit higher with one point four. But two experts stated that there could be problems in some particular cases. This is especially true for embedded systems. They often cannot be updated or have limited hardware resources. However, in general, there should not be any problems.

| Question 7 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (not secure)** | **2** | **3** | **4 (secure)** |
| **Interview 1** | | | | x |
| **Interview 2** | | | | x |
| **Interview 3** | | | | x |
| **Interview 4** | | | | x |
| **Interview 5** | | | | x |

*Table 6-5: Evaluation question seven*

Table 6-5 shows that AES-256 was rated as secure by all experts. This also is covered by the literature, as only Grover can be applied to it. All five experts rated it with a four. Hence, there is no difference between the mean and the median as both have a value of four.

| Question 9 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (not vulnerable)** | **2** | **3** | **4 (secure)** |
| **Interview 1** | X | | | |
| **Interview 2** | X | | | |
| **Interview 3** | X | | | |
| **Interview 4** | X | | | |
| **Interview 5** | x | | | |

*Table 6-6: Evaluation question nine*

Hashing algorithms behave identically to symmetric ciphers. Table 6-6 shows that all experts said that hashing algorithms are not vulnerable as long as a proper hashing algorithm is used, and the output size is large enough. Since all answers were the same, there is no difference between mean and median. Both have a value of one.

| Question 11 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (not relevant)** | **2** | **3** | **4 (relevant)** |
| **Interview 1** | x | | | |
| **Interview 2** | | | x | |
| **Interview 3** | x | | | |
| **Interview 4** | | X | | |
| **Interview 5** | | X | | |

*Table 6-7: Evaluation question eleven*

In question eleven, the relevance of quantum cryptography on end-user devices was not as clear as most other questions. The expert's responses varied here between not relevant, somewhat relevant. But as illustrated in question ten, it will be more relevant for marketing reasons than for security reasons. But four out of five said it would be rather not relevant or not relevant. The median says it is rather not relevant and the mean value with the values of two for the median and one point eight for the mean.

| Question 13 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (does not apply)** | **2** | **3** | **4 (applies)** |
| **Interview 1** | | | | x |
| **Interview 2** | | | | x |
| **Interview 3** | | | x | |
| **Interview 4** | | | | x |
| **Interview 5** | | | | x |

*Table 6-8: Evaluation question thirteen*

The results from question eleven were quite clear. Table 6-8 shows that all experts said that it at least somewhat applies. The median and the mean do not differ much with the values of four for the median and three point eight for the mean. Also, in interview three, it was clear that it is necessary to use post-quantum cryptography for certain kinds of data. This can be seen in question twelve. But most organisations do not have such data.

| Question 15 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (does not apply)** | **2** | **3** | **4 (applies)** |
| **Interview 1** | | | | x |
| **Interview 2** | | | | x |
| **Interview 3** | | | | x |
| **Interview 4** | | | | x |
| **Interview 5** | | | | x |

*Table 6-9: Evaluation question fifteen*

Question fifteen did not show any disagreements. Table 6-9 confirms that. All experts answered it with 'applies'. They said that crypto agility is already achieved if cryptography is implemented properly. Therefore, the focus should lie on the implementation of cryptography and the usage of the right libraries. Because of the same answers for this question in all interviews, the mean and median both have a value of four.

| Question 17 | | | | |
|---|---|---|---|---|
| **Answers** | **1 (does not apply)** | **2** | **3** | **4 (applies)** |
| **Interview 1** | | X | | |
| **Interview 2** | x | | | |
| **Interview 3** | | | x | |
| **Interview 4** | | X | | |
| **Interview 5** | | X | | |

*Table 6-10: Evaluation question seventeen*

Post-quantum cryptography as a mandatory requirement in ten years was assumed as 'applies rather not'. It is also represented by the mean and median. Both have the value two. Because it will take a few years till standards are finished, the time period will probably not suffice. This also depends on type of software. It will work quickly in the web, but standard software and industry will take a while to change.

## 6.4 Results

Now both, the quantitative and qualitative questions are evaluated, it is time to match the questions to with the corresponding thesis. With that done, it will be possible to answer the research question.

The H1 for thesis one is:

- **H1:** Currently used public key systems (RSA, Diffie-Hellmann, ECC) will be broken in the next 10 years.

Table 6-2 shows that only one expert said that it will be happen in ten years. All others said, as the mean and the median showed, that it will not happen in ten years. Additionally, the qualitative evaluation of question one showed that the breaking will happen somewhere between the next 10 and 20 years and the biggest challenges are the error correction and the engineering to build such large computers. Therefore, H1 cannot be confirmed, and H0 is true.

The H1 for thesis two is:

- **H1:** A standard for post-quantum algorithms is required first before a migration can be done.

In Table 6-3 it is illustrated that it will be likely that implementations will happen even before we have finished standards. It is possible today, according to question two from interview two. But in general, the winners of the NIST competitions will be utilized. Other organizations will orientate on the NIST results. However, China will use their own standards because they also have their own competition. H1 can be confirmed. It is possible to migrate, but it is useful to wait until the final standards are passed.

The H1 for thesis three is:

- **H1:** When standards for post-quantum algorithms are defined, they will not replace current schemes within a few years.

The summary of question four shows that it will not happen in a few years. It will take somewhere between ten and 40 years. Most probable it will take around 20 years. But as question five shows, it will be possible to run on current IT systems. H1 can be confirmed.

The H1 for thesis four is:

- **H1:** No new standards are needed for symmetric schemes if the key sizes are at least 256-bit long.

The result of question six states that symmetric ciphers are not as vulnerable as asymmetric ciphers. Hence, it is enough to double the key length. Therefore, AES-256 is still safe and offers valid resistance against quantum computers. Table 6-5 show that all experts agreed that AES-256 is secure enough. Based on these results, it is possible to confirm H1, but it needs to be a secure scheme like AES.

The H1 for thesis five is:

- **H1:** Current hash algorithms (SHA-2, SHA-3) with long input values are suitable for a post-quantum world.

Here, we a have a very similar circumstance with symmetric ciphers. As illustrated in Table 6-6 all experts said that hash functions are not vulnerable against quantum computers. Question eight also says that they are weaker than symmetric ciphers, but if a proper scheme is used and you double the output lengths, they are still good. This means H1 can be confirmed.

The H1 for thesis six is:

- **H1:** Quantum cryptography will not be as important as post-quantum cryptography for wide usage.

Question eleven varied. Four out of five interviewees said it would be rather irrelevant or not relevant and only expert said it might be somewhat relevant. This is also illustrated in Table 6-7. But the mean and median support the statement that it will be rather irrelevant. Question 10 supports this. The future of quantum cryptography on end-user devices is uncertain. It might hit the market for marketing purposes, but it will not increase the security level. It offers seemingly perfect security, but you have no authentication, and you need special hardware for it, which makes it quite expensive. It will be used for some special purposes. Therefore, H1 can be confirmed, and quantum-cryptography will not be as important as post-quantum cryptography for general usage.

The H1 for thesis seven is:

- **H1:** Sensitive data must be protected with post-quantum cryptography as soon as it is standardized.

Question twelve shows that there are various factors to consider. For the most data, it is not necessary. If you classify them, so that they must be secure even in 20 or 30 years, then the process should be started as soon as possible. But there is no clear definition what kind of data is has such importance. Most experts stated that governmental data, military data, and also critical infrastructure could be as important. If health data is that important is not clear. Some experts said they are extremely important, others not. The evaluation of question thirteen in Table 6-8 shows that all experts said that for that type of sensitive data, the migration to post-quantum cryptography should be done now. H1 can be confirmed but with the restriction that it does not apply to all types of data.

The H1 for thesis eight is:

- **H1:** If crypto-agility is taken into consideration today, the migration will be quicker.

Question fourteen showed clearly that crypto agility is achieved today if the cryptography is implemented properly. This means that no homebrew crypto implementations should be used. Instead, a standard crypto library should be used together with the API. If it is done that way, then crypto agility is achieved. To change the currently used crypto algorithm, only the API call must be adopted and can be changed very quickly. This was also confirmed by question fifteen as illustrated in Table 6-9. Therefore, H1 can be confirmed. If crypto is implemented properly today, then the migration will be quicker.

The H1 for thesis nine is:

- **H1:** If not all systems in the chain use post-quantum cryptography, the security of the system will be reduced.

As question sixteen shows, it will not be possible to enforce post-quantum algorithms in less than 20 years. This corresponds with question seventeen from Table 6-10 which says it will not be possible to make post-quantum cryptography a mandatory requirement in ten years. Because it will take a few years until final standards are published. It might be possible for some internet applications but not for everything. Question sixteen also shows that RSA could be a problem when quantum-computer cloud services with large computers are introduced to the market. Downgrade attacks will still happen like they did in the past. Therefore, H1 can be confirmed. This issue is not a quantum specific problem, we have to deal with such problems for decades.

What can be said to answer the question: How to secure encrypted connections and data against quantum computers? We have quite some years of time to adapt our products and services. You only have to act in the near future if you have data that must be secure within the suggested 20 or 30 years. This time should be used to implement cryptography properly, if not done already. Current public key encryption will be broken in the next two decades. First, at all, you should analyze what encryption schemes and hashing algorithms you use currently. In the beginning stages, you should aim to get rid of methods such as 3DES or MD5, which are not even used today. When this is done, you should check what key sizes and output length you are using. For AES, you should use 256-bit keys, For hashing algorithms an output with at least 384 bit is needed. With these parameters, you have a security level of still 128 bit, even against quantum computers. RSA might be affected in 20 years, but you can stall it if you increase the key size. RSA-2048 should not be used anymore. Instead moving to RSA-3072 or RSA-4096 is suggested.

However, there will be systems that cannot use these post-quantum algorithms in the future. It is vital to plan what to do with these legacy systems. If possible, isolating them is encouraged. Then the risk is reduced.

# 7 CASE EXAMPLE

The result from subsection 6.4 shall now be applied to a real-world example. This example is the Elektronische Gesundheitsakte (ELGA) system of the Austrian government. For this, the used encryption and hashing algorithms shall be analyzed and be checked if they are compatible with the results in this paper.

The whole technical specifications of the system can be found in Elga GmbH (2017). The used and accepted algorithms and protocols of EGLA are described in section 9.3 of the document.

Firstly, ELGA does not accept MD5 and SHA1 hashes. The hashes must be at least SHA256. That is quite acceptable for today. The document also says that longer outputs, SAH384 and SHA512, are supported and recommended. To provide a sufficient security level for the future the minimum supported length should be increased to 384 bit.

ELGA basically only accepts AES-256 for sensitive data. But for temporary encryptions, it also allows AES-128, AES-192, and 3DES. It is not clear why this is allowed and what temporary encryption means. It might mean for a short time such as a few weeks because long term is defined with months or years. If the possibility to use exists, AES-128, AES-192 and 3DES will be deprecated in the near distant future, it should be fine.

For asymmetric ciphers, RSA, and Elliptic-Curve Digital Signature Algorithm (ECDSA) are accepted. RSA must have a minimum length of 2048 bits, and ECDSA allows the NIST prime curves P-256, 384, and 512. Other elliptic curves are not allowed. Elliptic curves are quite secure today, but from the quantum view, they are more vulnerable than RSA. Hence, they should be deprecated. RSA on the other side should have a minimum length of 4096 bit. This will delay the point when quantum computers break it. The same recommendations are valid for digital signatures.

Finally, it can be said that the used algorithms in ELGA are quite suitable. Hashes should be increased to at least 384 bit,and is supported today. And ECDSA should be traded for RSA-4096.

# 8 CONCLUSION

Most of the thesis could be confirmed. This means that the opinions of the experts matched quite closely with the current state of published literature concerning this topic. The detailed results for each thesis can be found in section 6.4. What can be said in summary? Quantum computers are not an immediate threat. But they will be in 15 to 20 years, however, this threat can be prevented. Therefore, legacy algorithms such as 3DES and MD5 should be replaced as soon as possible. Then it would be useful to increase the key length of AES to 256 bit and of RSA to 4096bit in the next few years. AES will be quantum-safe with this key length, RSA 4096 will be a bit longer secure than RSA2048. Instead of MD5 or SHA1, SHA2 or SHA3 with a minimum output length of 384bit should be used, also in the next two few years. Besides that, the implementation of cryptography in the products and services should be checked. There should not be any self-done implementations. Anything should be based on a crypto library and only the API should be used. Afterwards it will be much easier to change the used cipher because only the API call must be replaced. If this is achieved, it will not be a challenge to move to post-quantum cryptography. But this will take more than five years und can be seen as a long-term goal. However, it is important to consider legacy systems and how to work with them.

The future of this field is unclear, and there is also a lot of speculation as it is not possible to say if and when quantum computers will scale big enough to be a real threat to public-key encryption that is currently used. The opinions in the field vary from ten years to never. But the interviews showed that organizations should prepare and be ready in ten years.

It can be said that there is room for improvement within this field. One question this paper cannot answer, is how this topic is anchored in companies and organizations. Therefore, it could be of interest to survey companies if they are aware this issue and what they are meaning to do about it. Consequently, interviews with IT representatives of companies could be conducted.

# Appendix A - Interview Guide

## INTERVIEW GUIDE POST-QUANTUM CRYPTOGRAPHY

Nr:
Date:
Profession:
Sector:
Experience:
Experience with PQK:

**1.) In which period to you think will today’s public key encryption be broken?**

Why you you think that?

**2.) What influence will have the standardization of PQC for the distribution?**

Why do you think that? Have you seen something similar? What standard could be the one which will be established?

**3.) On a scale from one (not likely) to four (likely), how likely will broad implementations be before we have finished standards?**

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| □ | □ | □ | □ |

**4.) If standards are published, how long will the transition take?**

Why do you think that? What could be the challenges?

**5.) On a scale from one (does not apply) to four (applies), will we need new it systems for the usage of PQC?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**6.) How ist the situation for symmetric ciphers? Do we need new ones, or are current ones suitable?**

Why do you think that? What approaches could this be?

**7.) On a scale from one (not secure) to four (secure), how would you rate the resistance of AES-256 against quantum-computers?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**8.) How ist the situation for hashing algorithms?**

Why do you think that? What approaches could this be?

**9.) On a scale from one (not vulnerable) to four (vulnerable), how vulnerable are current hashing algorithms against quantum.computers?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**10.) How do you see the chances of quantum cryptography against post-quantum cryptography?**

Why do you think that? Which one could be more important? Coexistence?

**11.) On a scale from one (not relevant) to four (relevant), how relevant will quantum cryptography be four end-user devices? (PC, Smartphone,…)**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**12.) Is PQC for some data more important than for others, or is it independent from the data?**

Why do you think that? When should you start with this data? Wait for standard? What kind of data is affected?

**13.) On a scale from one (does not apply) to four (applies), it the migration to PQC for sensitive data required today?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**14.) Can PQC considered today, and do you have any benefits in the future when you do it?**

Why do you think that?

**15.) On a scale from one (does not apply) to four (applies), can crypto agility as a goal in software projects speed up the migration?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

**16.) What are the effects if not all parts in the chain support PQC?**

Why do you think that? How to deal with legacy systems?

**17.) On a scale from one (does not apply) to four (applies), can PQC be declared as a mandatory requirement to systems in ten years?**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| □ | □ | □ | □ |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AES | Advanced Encryption Standard |
| BB84 | Bennet Brassard 84 |
| BSI | Bundesamt für Sicherheit in der Informationstechnik |
| BQP | Bounded-error, Quantum Polynomial time |
| CNOT | Controlled Not |
| DES | Data Encryption Standard |
| ECC | Elliptic-Curve-Cryptography |
| ECDSA | Elliptic-Curve Digital Signature Algorithm |
| ELGA | Elektronische Gesundheitsakte |
| FTS | Few-Time Scheme |
| HORS | Hash to Obtain Random Subset |
| HORST | Hash to Obtain Random Subset Tree |
| KEM | Key-Encapsulation Mechanism |
| LWE | Learning With Errors |
| MAC | Message Authentication Code |
| MD | Message Digest |
| NAND | Not And |
| NIST | National Institute of Standards and Technology |
| NMR | Nuclear Magnetic Resonance |
| NOR | Not Or |
| NP | Nondeterministic Polynomial time |
| OTS | One-Time Pad Scheme |
| PKC | Public Key Crypotogtaphy |
| QFT | Quantum Fourier Transformation |
| QKE | Quantum Key Exchange |
| QUBIT | Quantum Bit |
| RLWE | Ring Learning With Rrrors |

| | |
|---|---|
| RSA | Rivest–Shamir–Adleman |
| SHA | Secure Hash Algorithm |
| SVP | Shortest Vector Problem |
| TLS | Transport Layer Security |
| WOTS | Witernitz OTS |
| XOR | Exclusive Or |

# LIST OF FIGURES

List of Figures

# LIST OF TABLES

# 9 REFERENCE LIST

Aaronson, S. (2008). The limits of quantum. *Scientific American, 298*(3), 62–69. Retrieved October 10, 2019, from https://www.ime.usp.br/~pf/clippings/quantum/quantum-computing-200803.pdf.

Ajtai, M. (1996). Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 99–108).

Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.-K., et al. (2019). *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process: NISTIR 8240.* National Institute of Standards and Technology. Retrieved October 19, 2019, from https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

Alkim, E., Barreto, P. S., Bindel, N., Longa, P., & Ricardini, J. E. (2019). The Lattice-Based Digital Signature Scheme qTESLA. *IACR Cryptology ePrint Archive, 2019*, 85. Retrieved November 10, 2019, from https://eprint.iacr.org/2019/085.pdf.

Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2015). Post-quantum key exchange - a new hope. Retrieved November 09, 2019, from https://eprint.iacr.org/2015/1092.pdf.

Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., & Wang, L. (2009). Preimages for Step-Reduced SHA-2. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Lecture Notes in Computer Science. Advances in Cryptology – ASIACRYPT 2009* (pp. 578–597). Berlin, Heidelberg: Springer Berlin Heidelberg.

Artemenko, M. (2019). *Google achieves Quantum Supremacy? What it means for you.* Retrieved October 06, 2019, from quantaneo: https://www.quantaneo.com/Google-achieves-Quantum-Supremacy-What-it-means-for-you_a242.html.

Aumasson, J.-P., & Endignoux, G. (2018). Improving Stateless Hash-Based Signatures. In N. P. Smart (Ed.), *Lecture Notes in Computer Science. Topics in Cryptology – CT-RSA 2018* (pp. 219–242). Cham: Springer International Publishing.

Bennett, C. H., & Brassard, G. (2014). Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science, 560*, 7–11.

Bernstein, D. J., Buchmann, J., & Dahmén, E. (2009). *Post-quantum cryptography. Lecture Notes in Computer Science: Vol. 6061.* Berlin: Springer.

Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., et al. (2015). SPHINCS: Practical Stateless Hash-Based Signatures. In E. Oswald & M.

Fischlin (Eds.), *Lecture Notes in Computer Science. Advances in Cryptology -- EUROCRYPT 2015* (pp. 368–397). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bernstein, D. J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., & Schwabe, P. (2019). The SPHINCS + Signature Framework. In L. Cavallaro, J. Kinder, X. Wang, & J. Katz (Eds.), *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security - CCS '19* (pp. 2129–2146). New York, New York, USA: ACM Press.

Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. *Nature, 549*(7671), 188–194. Retrieved October 17, 2019, from https://www.nature.com/articles/nature23461.

Bogner, A., Littig, B., & Menz, W. (2014). *Interviews mit Experten: Eine praxisorientierte Einführung. Lehrbuch.* Wiesbaden: Springer VS.

Braithwaite, M. (2016). *Experimenting with Post-Quantum Cryptography.* Retrieved November 09, 2019, from Google: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html.

Brands, G. (2011). *Einführung in die Quanteninformatik: Quantenkryptografie, Teleportation und Quantencomputing. eXamen.press.* Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

BSI (2019). *Kryptographische Verfahren: Empfehlungen und Schlüssellängen.* Bonn: Bundesamt für Sicherheit in der Informationstechnik. Retrieved September 05, 2019, from https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf;jsessionid=3193308B6BFBC956340260AAA5F1542C.2_cid341?__blob=publicationFile&v=10.

Buchmann, J. (2008). *Einführung in die Kryptographie* (4., erweiterte Auflage). *Springer-Lehrbuch.* Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Buchmann, J., Dahmen, E., & Szydlo, M. (2009). Hash-based Digital Signature Schemes. In D. J. Bernstein, J. Buchmann, & E. Dahmen (Eds.), *Post-Quantum Cryptography* (pp. 35–93). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chaves, R., Kuzmanov, G., Sousa, L., & Vassiliadis, S. (2006). Improving SHA-2 Hardware Implementations. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Lecture Notes in Computer Science. Cryptographic Hardware and Embedded Systems - CHES 2006* (pp. 298–310). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography: NISTIR 8105:* US Department of Commerce, National Institute of Standards and Technology.

Cirac, J. I., & Zoller, P. (1995). Quantum computations with cold trapped ions. *Physical review letters, 74*(20), 4091.

Clement, A. NSA Surveillance: Exploring the Geographies of Internet Interception. In M. Kindling & E. Greifeneder (Eds.), *iConference 2014 Proceedings* (pp. 412–425). iSchools.

Cochran, J. F., & Mapother, D. E. (1958). Superconducting Transition in Aluminum. *Physical Review, 111*(1), 132–142.

Devoret, M. H., & Schoelkopf, R. J. (2013). Superconducting circuits for quantum information: an outlook. *Science (New York, N.Y.), 339*(6124), 1169–1174.

Dierks, T., & Rescorla, E. (2008). RFC 5246-the transport layer security (TLS) protocol version 1.2. *Internet Engineering Task Force.* Retrieved September 14, 2019, from https://tools.ietf.org/html/rfc5246#section-6.1.

Ding, J., & Schmidt, D. (2005). Rainbow, a New Multivariable Polynomial Signature Scheme. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Lecture Notes in Computer Science. Applied Cryptography and Network Security* (pp. 164–175). Berlin, Heidelberg: Springer Berlin Heidelberg.

Eastlake, D., & Jones, P. (2001). RFC 3174: US secure hash algorithm 1 (SHA1). *Network Working Group.* Retrieved August 29, 2019, from https://tools.ietf.org/html/rfc3174.

Elga GmbH (2017). *ELGA Gesamtarchitektur.* Retrieved November 08, 2019, from https://www.elga.gv.at/fileadmin/user_upload/Dokumente_PDF_MP4/Technisches/ELGA_Gesamtarchitektur_2.30a.pdf.

EUROPEAN PARLIAMENT (2016). General Data Protection Regulation: GDPR. In *REGULATION (EU) 2016/679,* p. 1.

Fehr, S. (2010). Quantum Cryptography. *Foundations of Physics, 40*(5), 494–531.

FIPSPUB180-4 (2015). Secure hash standard (SHS). *FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION 180-4.* Retrieved August 30, 2018, from http://dx.doi.org/10.6028/NIST.FIPS.180-4.

FIPSPUB197 (2001). Advanced Encryption Standard. *Federal Information Processing Standards Publication 197.* Retrieved August 31, 2019, from https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf.

FIPSPUB202 (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. *FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION 202.* Retrieved September 10, 2019, from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

FIPSPUB46-3 (1999). Data Encryption Standard (DES). *National Institute of Standards and Technology, 25*(10), 1–22. Retrieved September 07, 2019, from https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf.

Garey, M. R., & Johnson, D. S. (2009). *Computers and intractability: A guide to the theory of NP-completeness* (27. print). *A series of books in the mathematical sciences.* New York u.a: Freeman.

Gérard, F., & Rossi, M. (2019). *An Efficient and Provable Masked Implementation of qTESLA.* Cryptology ePrint Archive. Retrieved November 10, 2019, from https://eprint.iacr.org/2019/606.

Gidney, C., & Ekerå, M. (2019). *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.* Retrieved October 27, 2019, from https://arxiv.org/pdf/1905.09749.pdf.

Gilles Brassard, Peter Høyer, & Alain Tapp (1997). Quantum Cryptanalysis of Hash and Claw-Free Functions. In *ACM Sigact News* (pp. 163–169). Springer-Verlag.

Google (2019). *HTTPS-Verschlüsselung im Web.* Retrieved April 16, 2019, from https://transparencyreport.google.com/https/overview.

Grover, L. K. (1997). Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Physical review letters, 79*(2), 325–328. Retrieved September 13, 2019.

Gueron, S. (2010). Intel advanced encryption standard (AES) new instructions set. *Intel Corporation.* Retrieved October 27, 2019, from https://www.intel.com.bo/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper.pdf.

Homeister, M. (2018). *Quantum Computing verstehen: Grundlagen - Anwendungen - Perspektiven* (5. Aufl. 2018). *Computational Intelligence.* Wiesbaden: Springer Fachmedien Wiesbaden.

Housley, R. (2015). RFC 7696: Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms. Retrieved 31.10.219, from https://www.rfc-editor.org/rfc/pdfrfc/rfc7696.txt.pdf.

Hromkovič, J. (2014). *Theoretische Informatik: Formale Sprachen, Berechenbarkeit, Komplexitätstheorie, Algorithmik, Kommunikation und Kryptographie* (5., überarbeitete Auflage). *Lehrbuch.* Wiesbaden: Springer Vieweg.

Hülsing, A. (2013). W-OTS+ – Shorter Signatures for Hash-Based Signature Schemes. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Lecture Notes in Computer Science. Progress in Cryptology – AFRICACRYPT 2013* (pp. 173–188). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hülsing, A. (2017). *SPHINCS+ – The smaller SPHINCS.* Retrieved February 08, 2020, from https://huelsing.net/wordpress/?p=558.

Jones, J. A. (2000). NMR Quantum Computation: A Critical Evaluation. *Fortschritte der Physik, 48*(9-11), 909–924.

Kaiser, R. (2014). *Qualitative Experteninterviews: Konzeptionelle Grundlagen und praktische Durchführung. Lehrbuch.* Wiesbaden: Springer VS.

Kipnis, A., Patarin, J., & Goubin, L. (1999). Unbalanced Oil and Vinegar Signature Schemes. In G. Goos, J. Hartmanis, J. van Leeuwen, & J. Stern (Eds.), *Lecture Notes in Computer Science. Advances in Cryptology — EUROCRYPT '99* (pp. 206–222). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kirsch, Z., & Chow, M. (2015). Quantum Computing: The Risk to Existing Encryption Methods. Retrieved October 10, 2019, from http://www.cs-tufts.edu/comp/116/archive/fall2015/zkirsch.pdf.

Knill, E., Laflamme, R., Ashikhmin, A., Barnum, H., Viola, L., & Zurek, W. H. (2002). *Introduction to Quantum Error Correction,* from https://arxiv.org/pdf/quant-ph/0207170.

Lamport, L. (1979). *Constructing digital signatures from a one-way function.* Technical Report CSL-98, SRI International Palo Alto. Retrieved October 18, 2019, from http://lamport.azurewebsites.net/pubs/dig-sig.pdf.

Lenstra, A. K., Lenstra, H. W., & Lovász, L. (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen, 261*(4), 515–534. Retrieved October 17, 2019, from https://openaccess.leidenuniv.nl/bitstream/handle/1887/3810/346_050.pdf.

Löndahl, C. (2015). *Some Notes on Code-Based Cryptography,* Faculty of Engineering, LTH. Retrieved October 15, 2019, from https://portal.research.lu.se/ws/files/6280818/4934007.pdf.

Mavroeidis, V., Vishi, K., D., M., & Jøsang, A. (2018). The Impact of Quantum Computing on Present Cryptography. *International Journal of Advanced Computer Science and Applications, 9*(3).

Mayring, P. (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (12., überarb. Aufl.). *Beltz Pädagogik.* Weinheim: Beltz.

McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. *Coding Thv, 4244*, 114–116. Retrieved October 15, 2019, from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19780016269.pdf#page=123.

Merkle, R. (1979). Secrecy, authentication, and public key systems. *Ph. D. Thesis, Stanford University.* Retrieved 18.10.19, from http://www.merkle.com/papers/Thesis1979.pdf.

Merkle, R. C. (1990). A Certified Digital Signature. In G. Brassard (Ed.), *Lecture Notes in Computer Science. Advances in Cryptology — CRYPTO' 89 Proceedings* (pp. 218–238). New York, NY: Springer New York.

Micciancio, D. (2011). Lattice-Based Cryptography. In van Tilborg, Henk C. A & S. Jajodia (Eds.), *Encyclopedia of Cryptography and Security* (pp. 713–715). Boston, MA: Springer US.

Mlynárik, V. (2017). Introduction to nuclear magnetic resonance. *Analytical biochemistry, 529*, 4–9.

Moody, D. (2017). *The ship has sailed: The NIST Post-Quantum Crypto "Competition".*

Moody, D. (2019). *Round 2 of the NIST PQC “Competition”: What was NIST thinking?* Retrieved November 08, 2019, from National Institute of Standards and Technology: https://csrc.nist.gov/CSRC/media/Presentations/Round-2-of-the-NIST-PQC-Competition-What-was-NIST/images-media/pqcrypto-may2019-moody.pdf.

Mosca, M. (2013). *Setting the Scene for the ETSI Quantum-safe Cryptography Workshop.* e-proceedings of 1st Quantum-Safe-Crypto Workshop. Sophia Antipolis.

Mosca, M. (2018). Cybersecurity in an Era with Quantum Computers: Will We Be Ready? *IEEE Security & Privacy, 16*(5), 38–41. Retrieved April 18, 2019, from https://ieeexplore.ieee.org/abstract/document/8490169.

National Academies of Sciences, E. a. M. (U.S.) (2019). *Quantum computing: Progress and prospects.* (Grumbling, E., & Horowitz, M., Eds.). Washington, DC: the National Academies Press.

Niederhagen, R., & Waidner, M. (2017). *Practical Post-Quantum Cryptography:* Fraunhofer SIT.

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th anniversary ed.). Cambridge: Cambridge Univ. Press.

O'Brien, J. L. (2007). Optical quantum computing. *Science (New York, N.Y.), 318*(5856), 1567–1570.

Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., & Ding, J. (2015). Design Principles for HFEv-Based Multivariate Signature Schemes. In T. Iwata & J. H. Cheon (Eds.), *Lecture Notes in Computer Science. Advances in Cryptology -- ASIACRYPT 2015* (pp. 311–334). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pöppelmann, T. (2018). *Post-quantum cryptography on embedded microcontrollers* (Graz Security Days for Industry 2018). Infineon. Retrieved November 09, 2019, from https://www.silicon-alps.at/wp-content/uploads/2018/10/PQC_Poeppelmann.pdf.

Proos, J., & Zalka, C. (2003). Shor’s discrete logarithm quantum algorithm for elliptic curves. *arXiv preprint quant-ph/0301141.* Retrieved October 10, 2019, from https://arxiv.org/pdf/quant-ph/0301141.pdf.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM, 21*(2), 120–126. Retrieved September 05, 2019, from http://doi.acm.org/10.1145/359340.359342.

Rivest, R. (1992). The MD5 message-digest algorithm. Retrieved August 29, 2019, from http://altronic-srl.com.ar/md5%20algoritmo.pdf.

Sanadhya, S. K., & Sarkar, P. (2008). New Collision Attacks against Up to 24-Step SHA-2. In D. R. Chowdhury, V. Rijmen, & A. Das (Eds.), *Lecture Notes in Computer Science. Progress in Cryptology - INDOCRYPT 2008* (pp. 91–103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Savage, N. (2018). Quantum computers compete for “supremacy”. *Scientific American, 27*, 108–111. Retrieved October 17, 2019, from https://www.nature.com/articles/scientificamericantimerevolution0718-108.

Scherer, W. (2016). *Mathematik der Quanteninformatik: Eine Einführung.* Berlin, Heidelberg: Springer Spektrum.

Shor, P. & Shor, J. *Peter Shor.* Retrieved October 20, 2019, from www-math.mit.edu/~shor/.

Shor, P. W. (2004). Progress in Quantum Algorithms. *Quantum Information Processing, 3*(1), 5–13, from https://doi.org/10.1007/s11128-004-3878-2.

Sobti, R., & Ganesan, G. (2012). Cryptographic Hash Functions: A Review. *International Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol 9*, 461–479.

Spitz, S., Pramateftakis, M., & Swoboda, J. (2011). *Kryptographie und IT-Sicherheit: Grundlagen und Anwendungen* (2., überarbeitete Auflage). Wiesbaden: Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden.

SSLLABS (2019). *SSL Pulse.* Retrieved October 27, 2019, from SSL Labs: https://www.ssllabs.com/ssl-pulse/.

Stevens, M. (2006). Fast Collision Attack on MD5. *IACR Cryptology ePrint Archive, 2006*, 104. Retrieved August 29, 2019, from http://crppit.epfl.ch/documentation/Hash_Function/Examples/Code_Project/Documentation/104.pdf.

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The First Collision for Full SHA-1. In J. Katz & H. Shacham (Eds.), *Advances in Cryptology - CRYPTO 2017* (pp. 570–596). Cham: Springer International Publishing.

Tanenbaum, A. S., & Goodman, J. (2004). *Computerarchitektur: Strukturen, Konzepte, Grundlagen* (4. Aufl., Bafög-Ausg; [Nachdr. des 3. Dr., 2002]). *Informatik Grundlagen.* München: Pearson-Studium.

Uysal, M., Capsoni, C., Ghassemlooy, Z., Boucouvalas, A., & Udvary, E. (2016). *Optical Wireless Communications.* Cham: Springer International Publishing.

Vedral, V., & Plenio, M. B. (1998). Basics of quantum computation. *Progress in Quantum Electronics, 22*(1), 1–39.

Wang, C., & Wulf, W. A. (1997). A framework for security measurement. In *Proc. National Information Systems Security Conference, Baltimore, MD* (pp. 522–533).

Xie, T., Liu, F., & Feng, D. (2013). Fast Collision Attack on MD5. *IACR Cryptology ePrint Archive, 2013*, 170. Retrieved August 29, 2019, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.4421&rep=rep1&type=pdf.