Post-Quantum Cryptography

*

Holly Baker
CS student
Trinity College
Dublin, Ireland
bakerho@tcd.ie

***Abstract*—Post-Quantum Cryptography is the area which consists of cryptographic systems that are secure against classical and quantum computers. This report details the modern cryptography infrastructure secure from classical computers. Aspects explored are type's of current cryptosystems and why these are breakable. This allows us to understand why Post-Quantum Cryptography is needed and the dangers of a quantum attack. These attacks will be issued from quantum computers running two main algorithms, Shor's and Groove's. These new computers, based on quantum physics are explained in the report, alongside the algorithms.**
A way to combat these attack's, is the use of cryptographic scheme's which rely on hard problems, such that the quantum computers will not efficiently solve these hard problems. The report details the main families being examined by the cryptography community, each family relies on a mathematical problem which is considered hard to solve by a quantum computer. Within each family there are different schemes, with their own security and efficiency. Some of these post-quantum algorithms are in the process of being standardized by the standardizing organisation NIST. This is detailed in the report with specific leading examples in the current competition.

***Index Terms*—Post-Quantum Cryptography, , Quantum Computers, Cryptosystem's, Public-Key Cryptosystem's, NIST.**

I. Introduction

The research on **quantum computers** has increased in recent years. These Quantum Computers exploit quantum mechanical phenomena to solve mathematical problems that are difficult or intractable for conventional computers. This means if large-scale quantum computers are manufactured, they will have the power to break many of the **public-key cryptosystem's** currently used in the world. *(Chen,**2016**)*

Even though this threat is hypothetical at the moment, if these significant technical advances are made the confidentiality and integrity of digital communications would be heavily compromised. This problematic situation has led to the discovery of Post-Quantum Cryptography,a solution to this threat.

To recognize the risk and what can be done, it's essential to look closely at the current digital cryptography and how it's used - and broken. Additionally understanding the power of **quantum physics** and quantum machines, will allow us to grasp the desired technical advancements that have yet been made. And lastly understanding,Post-Quantum Cryptography, the solution to combat this quantum attack.

Identify applicable funding agency here. If none, delete this.

II. What is Cryptography ?

Protecting data from unauthorized access is crucial for the information society to run successfully. There are many different aspects in the world which need security. For example, everyday services such as cash withdrawal from ATMs, email and file storage or services surrounding governments such as the military's arrangements.Cryptography enforces this data confidentiality needed in society. It is the science of **hiding information** in plain sight, through constructing protocols that **prevent third parties** from reading private messages. *(en.wikipedia.org,**2020**)*

These complex protocols and algorithms that are developed to implement a particular security service can be called **Cryptosystems**. They guarantee certain security properties, alongside the implementation of cryptographic techniques. These systems use complicated **mathematical formulas** to transfer **Plaintext** data,into- and out-of- securely encrypted messages called **Ciphertext**.

Cryptographic primitives typically rely on:

- Symmetric Encryption
- Asymmetric Encryption
- Signatures
- Key Exchange
- Hashing

Encryption can be split into two main types: **Symmetric-Key Encryption** and **Asymmetric-Key Encryption**. In an Symmetric cryptosystem, the **same key** will be used to encrypt and decrypt the data. In an Asymmetric cryptosystem, also known as **public key cryptography** each person generates a pair of **linked keys**.One key is **shared** publicly to let people encrypt messages for the owner of the pair of linked keys, and the the second key is kept **secret** by the owner to decrypt messages *(Denning*

Symmetric encryption is used to encrypt all communications and store data, specific examples are **DES(Data Encryption Standard)** or **AES(Advanced Encryption Standard)**. But the Symmetric key algorithms are not always convenient due to the **key exchange and trust problem's**. Public-key encryption fixes this problem of the necessity of sharing the key although it **consumes more time**, therefore it is used for securely exchanging symmetric keys. *For example on regular "https" website the browser makes use of asymmetric encryption to verify the sites certificate, it then introduces a symmetric key for encrypting information that's exchanged to and from the website.*

Digital Signature's are used for **authentication** and **data integrity**. It is a *cryptographic value* that is computed from the data and a private key known only be the signer. **Hashing** plays a big role in the signature process. Rather than computing the cryptographic value on the entire data, the hash of the data is used as it is more efficient. The hashed value of the data is the input into the *signature algorithm* alongside the signers private key, which outputs the digital signature. The digital signature is then fed into the *verification process* which inputs the signature into the verification algorithm and makes use of the *same hash function*.
The hash value and the output of the verification algorithm are compared. This **comparison**, will detect if the **signature is valid**. Digital signatures are combined with *encryption systems* in order to achieve information security. *(tutorialspoint,**2020**)*

At this instant, this is how the basic cryptography is being implemented. **Digital signatures** are added features to systems to increase data confidentiality. Then there are the two main encryptions, **symmetric** and **asymmetric**, which are being combined in many applications. These two systems follow two different mathematical paths, this difference can be reflected in the corresponding *security levels* for the two types of cryptography.

III. Quantum Algorithms?

There are two leading quantum algorithms that have been used in crypt-analysis.**Shor's algorithm** which is the algorithm for **integer factorization** that runs in polynomial time. This advancement made in 1994 by Peter Shor, has endangered *RSA* and *discrete log-based cryptosystems*. This implied all **asymmetric cryptosystems** are breakable with a large-scale quantum computer.

And Secondly **Grover's algorithm** which was devised by Lov Grover in 1996.This has the ability to **invert functions** in *O(squareroot(n))* time. Although Grover's algorithm would not break **symmetric key cryptosystems**, it would reduce the security by a root factor. *For example the symmetric cryptosystem AES-256, if Grover's algorithm were to be applied it could brute-force a 256-bit key in around 2128 iterations. Therefore it would only offer 128-bits of security (Perez,**2018**)*

However this algorithm will not destroy the symmetric cryptosystems, if cryptanalytics **increase the security** by a *factor of 2* it will diminish the issue and resolve the reduced security back to its initial level.

IV. How to break them Public-Key system's?

With these quantum algorithms in mind,Quantum computers impose the biggest threat on *public-key cryptography*. The main algorithms for public-key systems are **RSA(Rivest-Shamir-Aldeman) cryptosystem** , **Diffie-Hellman key exchange** and **Elliptic Curve cryptosystems**. The security of these algorithms relies upon the difficulty of particular number **theoretic problems** such as *Integer Factorization* or the *Discrete Log Problem* over various groups.*(Chen,**2016**)*

Public-Key Systems	
Algorithm	Hard-Problem
RSA	Prime Factorization
Diffie-Hellman	Discrete Logarithm
Elliptic Curve	Elliptic Curve Discrete Logarithm

*(Vladimir,**2017**)*

The main reason for the attack to impose a bigger threat on public-key systems is that all the stated algorithms above can **start with a public key** and mathematically **compute the private key** without attempting all the possibilities.
The descriptions of these three main public-key system's are listed. RSA is explained with a *detailed example* illustrating how a quantum computer could dismantle the RSA system.

A. RSA:

The following example of RSA, will detail how the cryptosystem functions and how it can be broken down by Quantum Computers:

Note: Theorems and complicated mathematical equation's are used .They're explanations are outside the scope of this report.

There are two stakeholders **Alice** and **Bob** who want to communicate in private, such that if a third party listens in they will not be able to understand the message between the two stakeholders. This scheme will be based on the **RSA cryptosystem**.

Firstly, Alice randomly produces 2 **big prime figure's**, x and y. With these she **computes** N such that it equals $x * y$ AND ϕ which equals $(x-1)*(y-1)$.

ϕ equals to the amount of numbers **co-prime to** N which is named the Euler ϕ function. Following this Alice

randomly picks a **number** z co-prime with ϕ. She then **calculates** s given $sz = 1 * mod(\phi)$. Alice is computing these calculations in order to achieve a private and public key for a RSA cryptosystem.

The **private key** equals s and the **public key** equals N and z. Alice now wants to communicate a message as she has her private key and public key. To interact with Bob, she must show Bob her public key. With this Bob handles his **message** M and **computes** $(a * M)^z * mod(N)$ with the input of Alice's public key, $N$and z, and also his message M. Alice is then forwarded the **value** a from Bob.

Alice is able to work out the message M through **decryption**. The decryption process takes a to the sth power modulo N and subbing this into $(a^s) * mod(N) = (M^{z*s}) * mod(N)$.This expression will result in the message, however more calculations are needed to derive it.
Recall the **int k** that $s * z = k * (\phi) + 1$. We can assign $p = k * (\phi)$. Then $(M^{k(\phi)}) * modx = (n^{x-1})^{k*(y-1)} mod(x)$ and by a theorem called **Fermat's Little** this will result to one:
$M^{k*(\phi)} mod(x) = 1$.Additionally, $M^{k*(\phi)} mod(y) = 1 * mod(y)$.
Another theorem called **Chinese remainder** is also used, this states that $n^{k*\phi} mod(x * y) = 1$.Therefore we see that $M^{z*s} * mod(N) = M^{k*\phi+1} * mod(N) = M * mod(N)$.

B. Diffie-Hellman key exchange:

This is not a public key encryption system, however it allows two parties, e.g Alice and Bob to establish a **shared secret** in a public channel. This enables secret communication and therefore an eavesdropper cannot read the message. If the eavesdropper where to solve the **Discrete Log Problem**, they would be able to solve the Diffie-Hellman problem.
The problem of calculating x below is known as the Discrete Logarithm Problem:
Given a *cyclic group, G* and a *generator, g from G*, and the following element $h = g^x$How hard is it to find x?
*(Winkleer,**2020**)*

C. Elliptic Curve cryptosystems:

This public-key cryptography scheme focuses on the algebraic structure of **elliptic curves over finite fields**. These protocols security relies on the ability to compute **point multiplication** and the hardness of the problem to **compute the multiplicand** given the original and product points. Another factor which alters the hardness of the problem is the **size** of the elliptic curve.
This scheme has the ability to be integrated into *key-exchange systems* and *digital signatures*. There are many different cryptosystems which have adapted this mechanism, for example **Elliptic Curve Diffie-Hellman** or **the Elliptic Curve Digital Signature Algorithm**. *(en.wikipedia.org,**2020**)*

V. What are Quatnum Computers ?

The machines that break these systems are based on quantum physics rather than more standard electronics that build the classical computers. Quantum computing uses **quantum bits**, known as *qubits*. The difference between these and binary bits(0's and 1's), Is quantum bits can exist in **both states simultaneously**, as well as many other states in between.*(Marr,**2018**)*
These qubits are usually **subatomic particles** such as *photons* or *electrons*. They possess certain properties that allow a linked group of qubits to perform with more processing power than any corresponding number of binary bits. Examples of these properties are **superposition**, **entanglement** and **interference**.

However, Scientists in this field struggle to develop and control these qubits. None of the existing quantum computers have reached the processing power needed to break the current cryptosystems. A metric which can estimate quantum capability is **quantum volume**. This measures the relationship between *number* and *quality of qubits*, *circuit connectivity* and *error rates of operations*. Quantum computers with larger quantum volume will bring the first quantum attack on cryptosystems. *(IBM,*

VI. Post-Quantum Cryptography ?

Estimating the exact arrival time of these large scale quantum computers and relying on this timing is unrealistic. In this present moment, our society must actively engage with our current security infrastructure and brace it for quantum computing.
To protect the public-key system and keep it intact, many cryptographers are designing new **"quantum-safe"** algorithms. This is Post-Quantum Cryptography and it refers to the *cryptographic algorithms* that are thought to be **secure** against an attack by a **quantum computer**. These algorithms run on classical computers and have the mathematical capability to withstand attacks from a quantum computer.

The mathematical hard problems that quantum computers cannot solve can be spilt into **different families**. Each families security depends on that problem and its hardness.

The current families being discussed by the cryptographic community are **lattice-based**, **code-based**, **hash-based**, **multivariate** and **supersingular elliptic-curve isogney cryptography**. The key aspects used to compare these are the *computational cost's*, the *efficiency* which is measured on the size of the public and private keys and most importantly the *security* of the scheme which depends on variables such as the amount of trust in a problems strength and the maturity of the problem.

The **Code-based** and **hash-based families** were established in the 1970s. Due to their age, they are said to be trusted

and well-understood. Following this the **Multivariate Cryptography family** was introduced in the 1980s. Although its mathematical problem is accepted, there are not many multivariate public-key cyrptosystem's that exist. To build a efficient version of this system is a universal challenge in the cryptography community. In the 1990s, the **Lattice-based schemes** were drafted. This family has been a popular contestant in the standardisation process. It's reliability and potential has increased due to the advancement's made in the crypt-analysis of **hash-based algorithms**.
The most recent family was developed in 2011 after elliptic-curve isogenies were reshaped with supersingular curves. **Elliptic-curve isogenies** were introduced in 2006, this family has not yet been deeply studied which makes it the weakest opponent.

A. Lattice-based:

Lattices are a natural next step from *matrices in linear algebra*. When given a set of vectors that point in different directions and cannot be expressed as the skill sum of any other vector in the set, we can take the skill sum of these vectors to create other vectors that are also in the lattice.
Understanding the fundamental concepts behind lattice problems, we will be able to see how these relevantly **new cryptosystems** will be able to replace all of the current endangered systems. Lattice problems also make way for an entire new class of powerful **cryptographic tools**.

In general we describe a lattice by organizing its **basis vectors** into a *n x m* matrix.Manipulating these matrices is the core of lattice-based cryptography. We can define certain problems that are really hard to solve. *Recall the hardness of factoring is what makes RSA so secure.*
By finding hard problems that are easy to construct but hard to crack we can develop a new method for a public-key cryptography. Such hard problems are the **closest vector** and **bounded distance decoding.**
The **closest vector problem** states when given a basis of a *lattice* and a *vector, V*, not in the lattice, find the closest vector to *v* that is in the lattice.
The **Bounded Distance Decoding** is similar to the closest vector problem. It states to find the lattice point, *s*, closest given that *v* is known to be close to *s*.

The particular **advantage** of lattice problems is there is **no efficient algorithm** classical or quantum that can solve these problems in better than **exponential time**.
This detailed example below is derived from **the bound distance decoding hard problem** and it shows the Alice and Bob infrastructure withstanding a quantum attack.

Recall Alice and Bob's RSA system, such that Alice publishes a public key and chooses a private key. Then Bob uses the public key to encrypt a private message, this encrypted message is sent back to Alice. Using her private key, she can easily decrypt his message. However an eavesdropper must solve a very hard problem in order to decrypt the private message.
In this case this hard problem is called **Learning with error**. **Alice** and **Bob** choose a **random matrix** A with dimensions m, n that is **public**. The only note about **A** is that each of its **entries** is actually taken $modq$ for some further larger integer q and that integer is also **public**.
Alice then chooses a **private vector** called x. x is unique because each of its m entries is either 0 or 1 , binary. She computes

$$A * x = u$$

and publishes the u **vector** as her **public key**. Crucially this multiplication is actually a **collision resistant hash function** and it is therefore very difficult for an eavesdropper to determine x.
So Bob now has the tools to send an encrypted message to Alice. Bob computes 2 **vectors** $b1$ and $b2$, and will send them both to Alice. First Bob chooses a **secret random vector** s with m entries.Now $b1$ is defined as $s * A + e1$, where $e1$ is another random vector with m integers except that now the magnitude of the entries in $e1$ must be quite small.
To compute $b2$ Bob takes s as before but this time multiplies it by **Alice's public key** u. As a **different error term** $e2$ with the same description but different vice and finally as a **third term** $bit * q/2$, where the bit is his message 0 or 1 that he wants to send and $q/2$ is an integer of **substantial magnitude** $modq$ that is also much larger than any error values. In general **error** is much less than $q/4$.
Alice can find out if Bob sent a 0 or 1. She multiples $b1$ by her **private key** x, then she takes the **difference** $b2 - b1 * x$ by distributing the x from the first equation we can expand out and write $b1 * x = s * u + e1 * x$. The s u terms **cancel** in the subtraction leaving just $(e2 - e1 * x) + bit * q/2$.
The key insight is that the **parentheses term** is quite small compared to $q/2$ so if the bit is 0 the expression will **evaluate** to something close to 0, and **Alice** will **recognize** that bob sent her a 0 but if bob sent a 1. The magnitude of the value will be quite far from 0 and Alice will be able to determine he sent a 1.

Why is this problem hard? Well the algorithm above made no mention of the *learning with error problem* but we know cracking the cryptosystem above is as hard as solving the learning with error problem. And we think the *LWE problem* is hard in the same way we think *factoring* is hard on *standard computers*, we don't have a good way of cracking it.

So what exactly is learning with error?Picture a *simplified lattice*, and we think of *matrix A* from our problem as providing the *basis vectors* for this lattice. The lattice interpretation is that given some point *b2* near a particular lattice point *s*, **trying to recover the exact entries in s is**

hard. This problem is called *bounded distance coding* and is strongly believed to be hard enough to be the basis of a cryptographic system.

B. Hash-based cryptography:

This family is centralized on **digital signatures** created using hash functions. The **one-way hash functions** maps bit-strings of an arbitrary length to short fixed-length bit-strings.These short fixed-length bit-strings are called **hash values**. There are three properties based on hardness that must be meet for it be a secure cryptographic hash function, these are *preimage resistance*,*second preimage* resistance and *collision resistance*.
Preimage resistance means it must be hard to invert .**Second preimage resistance** means with a given bit string, it is hard to find another bit string with the same hash value. **Collision resistance** means to find two bit arbitrary string's that give the same hash value is hard.

In terms of authentication, these properties secure the hash function and makes it reliable against quantum computers. Hash functions are not damaged by *Shor's algorithm*. But they have yet to build a public-key cryptosystem independently as computing the **inverse of a hash function** is not computationally feasible. Despite this, signature schemes can be constructed from hash functions.

C. Code-based cryptography:

Error-correcting codes is the key foundation to this cryptography, they hide the contents of a message throughout a transmission. Outside the cryptography community, they're functionality is to catch and correct bit errors when messages are transmitted over an unreliable channel.
This protocol is adjusted for the use of it in the code-based cryptography. Specifically in the **McEliece code-based cryptosystem**, it is adjusted in such a way that **errors** are deliberately **added** in order to protect the contents of a message against a third-party. *(Niederhagen,**2017**) A description of the McEliece cryptosysetm is detailed later in the paper.*

D. Multivariate cryptography:

The Multivariate family is formed on the hardness of the **multivariate quadratic polynomial (MQ) problem**. The hardness of solving this problem depends on certain **input parameters**.These parameters are the *number of variables*, *the size of the base finite field* and *the number of equations*.
Multivariate algorithms favour **signature schemes**, as public-key cryptosystems of multivariate nature are not efficient enough.
An example of a digital signature scheme is called **The unbalanced oil and vinegar (UOV)** which comes from "Oil and Vinegar". The main concept is the mathematical procedure of **hiding quadratic equations** in n unknowns named "oil" and v ¿ n unknowns named "vinegar" over a finite field K alongside linear secret functions. This **minimal quadratic equation** system must be solved in order to crack the hard problem.

E. Supersingular elliptic-curve Isogenies cryptography:

Recall that Shor's algorithm on a quantum computer solved the *discrete log problem on elliptic curves*.The relatively new "quantum-safe" algorithm adjusts this current protocol, such that it **defines operations between different elliptic curves** rather than computing on points of an elliptic curve.
Instead of performing operations like addition and subtraction on points of a curve, operations will be computed on **separate elliptic curves**. The mathematical operators that map a curve onto another curve have its own set of unique properties. Maps with these properties are called **isogenies**. An example of this scheme receiving consideration in today's world is in the following section.

VII. Post-Quantum Cryptography in Today's World

Researchers have estimated that it is possible by 2030 for a quantum computer to be built on a 1 billion dollars budget that will have the quantum volume to break a 2000-bit RSA. The organisation **NIST**, *National Institute of Standards and Technology*, standardized the current cryptosystems in place , it is one of the most actively engaged organisations and they are currently in the **process of standardizing** post-cryptography algorithms.

In 2016 the organization NIST began accepting nominations for new post-quantum cryptographic algorithms with the potential of being standardized. They received **82 unique submissions**, 59 of these for encryption and 23 for signed signature schemes. Their selection of round 2 candidates was based on the specific attributes, **security** and **performance**. These selected candidates were announced on January 30, 2019.

They are considering two main categories **key-establishment algorithms** which is the equivalent to the Alice and Bob situation. These candidates will potentially replace the public-key cryptosystems, *e.g RSA*. And Secondly **digital signatures** that will provide authenticity of data. These signature systems will feature in code-signing applications used to establish confidence in the program.

The following are strong second round candidates:

A. SPHINCS

Is a **stateless hash-based signature scheme**.It has **security proofs** that only depend on the security of a hash function. These hash functions have been present in cryptography for decades, therefore their security has **high confidence**.Additionally, it has very **small public keys** - *32 to 64 bytes*. However, *signing* is very **slow** and *signatures* are relatively **large** *(Alagic,**2019**)*

B. Classic McEliece

The system is based on the **first code-based public-key cryptosystem**. The *security problem* is focused on the hardness of **decoding a general linear code**. It also has *short ciphertexts* and *good performance* in terms of encapsulation and decapsulation. The security problem has a long history of analysis, making the security more **creditable** for the cryptosystems. However the main disadvantage is the **large public key size**.

C. Rainbow

This is a **multivariate digital signature scheme**. It is a generalization on the structure of *unbalanced oil and vinegar (UOV)*. But it allows parameterizations which are more **efficient** with the cost of added algebraic structure.

D. SIKE

This **key-establishment algorithm** is based on the arithmetic properties of **elliptic curves over finite fields**. This was the only candidate submitted related to *Isogenies cryptography*. It has the **smallest key size** among all candidates. It also holds the benefit of an **easier integration** compared to other submissions. This is because it can be easily combined with the currently used *elliptic curve cryptography* to create the needed post-quantum scheme.
However the area of which the security problem is based on, finding isogenies between supersingular elliptic curves, has **not been researched** as much compared to other submissions *e.g the lattice-based security problem*. Therefore, the probability of the security failing is higher. Another issue is the performance is **slower** when compared to the other submissions.

E. NTRU

Is a public-key cryptosystem which uses **lattice-based cryptography**, to encrypt and decrypt. NTRU is split into two algorithms, **NTRUEncrypt** used for *encryption* and **NTRUSign** used for *digital signatures*. NTRUEncrypt has received the most attention from the NIST in terms of submissions.The security has been scrutinized for decades. This means the security is **reasonably-well understood**, making it a strong candidate. However it **lacks security proofs** in contrast to *hash functions* which means it must rely on the fact it has yet to been broken after years of attempts. NTRU construction produces a *lattice* which has **more structure** than in similar cryptosystems *RLWE*.

VIII. Conclusion

Quantum computing is a fascinating industry that has the full potential, to dismantle many of the crucial cryptosystem's. The area may or may not reach the large-scaled quantum computers required to release such attacks but the constructing of one is possible. Therefore,The threat of quantum computers must be taken seriously and resources must be put in to making the preparation for a successful changeover from the **current cryptography** to **post-quantum cryptography**.

The future of Post-Quantum Cryptography will take a huge step forward this year, 2020. As NIST plans to either select finalists or make the final choice of candidates algorithms to be standardized. This report has evaluated the strongest candidates alongside their families they originate from.

After the selection, the implementation of these schemes will begin. Current software must be integrated with these post-quantum schemes. And specific new hardware devices will be developed and combined with quantum-safe algorithms such as *smart cards*. With these future adjustments, society must be educated on the power of quantum computing and it's impact on our cyrptosystem's. This understanding is conveyed in the report, illustrating the overall subject of Post-Quantum Cryptography.

References

[1] EN.WIKIPEDIA.ORG - *Cryptography* ,2020, https://en.wikipedia.org/wiki/Cryptography.

[2] TUTORIALSPOINT- *Cryptography Digital Signatures* 2020, https://www.tutorialspoint.com/cryptography/cryptography$_{d}igital_{s}ignatures.htm$.

[3] NOLAN WINKLEER - *The Discrete Log Problem And Elliptic Curve Cryptography*,2020, http://math.uchicago.edu/ may/REU2014/REUPapers/Winkler.pdf.

[4] EN.WIKIPEDIA.ORG. - *Elliptic-curve cryptography*, 2020,https://en.wikipedia.org/wiki/Elliptic-curve$_{c}ryptography$.

[5] LILY CHEN - *Report on Post-Quantum Cryptography* 2016, https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf.

[6] DOROTHY DENNING - *Is Quantum Computing a Cybersecurity Threat?* https://www.americanscientist.org/article/is-quantum-computing-a-cybersecurity-threat.

[7] VLADIMIR VALYUKH - *Performance and comparison of post-quantum cryptographic algorithms* 2017,https://liu.diva-portal.org/smash/get/diva2:1111159/FULLTEXT01.pdf.

[8] BEN PEREZ - *A Guide to Post-Quantum Cryptography* 2018,https://blog.trailofbits.com/2018/10/22/a-guide-to-post-quantum-cryptography/.

[9] BERNARD MARR - *20 Mind-Boggling Facts About Quantum Computing Everyone Should Read* 2018,https://www.bernardmarr.com/default.asp?contentID=1361J.

[10] IBM - *What is quantum computing?* 2018,https://www.ibm.com/quantum-computing/learn/what-is-quantum-computing/.

[11] RUBEN NIEDERHAGEN - *Practical Post-Quantum Cryptography*
2017,https://www.sit.fraunhofer.de/fileadmin/dokumente/studien$_u nd_t echnical_r eports/$
$Practical.PostQuantum.Cryptography_W P_F raunhoferSIT.pdf?_=$1503992279.
[12] GORJAN ALAGIC - *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process*
2019,https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.