OST

Student research project

Post Quantum Cryptography

Department of Computer Science
Eastern Switzerland University of Applied Science
Campus Rapperswil-Jona

Autumn Term 2021

Authors Isaac Würth
Marco Zanetti

Advisor Nathalie Weiler
Project Partner Kai Schramm

Abstract

Quantum computers are becoming a reality in the industrial sector. With the quantum cloud from IBM putting quantum computing resources within reach of everyone with an internet connection. The computing power of these machines is starting to surpass their conventional counterparts and they are opening up new opportunities for solving problems unfeasible on traditional computers. These can be complex logistical optimizations, simulations of molecular interaction in drug development or the quick calculation of incredibly hard mathematical equations, etc.. One of these mathematical problems, of which they can reduce the calculating time, is the so called factorization problem. The issue with this is, the impossibility to efficiently factorize large numbers on conventional computers has been the foundation of modern cryptographic algorithms like RSA and ECC.

Back in 1994, Shor's Algorithm was invented for a more efficient way of breaking the factorization problem. This algorithm, if run on a quantum computer, could crack RSA (Rivest-Shamir-Adleman), which is based on said problem. But since the algorithm could only run efficiently on a future quantum computer, no change was needed in cryptography. Now that quantum computers have matured, this algorithm was put to the test, and it showed that it is capable of cracking asymmetric cryptography as expected. Rendering RSA useless is a huge problem for our modern IT infrastructure, since RSA is used in almost every data exchange via the internet. Quantum computers allow possible attackers to decrypt and read sensitive data in the near future, no matter how well it was encrypted. Secure communication as we know it, would cease to exist. A possible prevention with quantum resistant cryptography, so called "post quantum cryptography" will be the focus of this essay.

The goal of this essay is to provide an overview over what is currently being done to prepare the IT infrastructure for the coming quantum threat. This is done by showcasing the most pressing issues of post quantum cryptography and some relating topics which we picked to be researched further. These include the current technical development of quantum computers, the current status of NIST (National Institute of Standards and Technology) standardization process for post quantum algorithms, certificates, cryptographic agility, hardware security modules and quantum key distribution. All of these topics were regarded in relation to a future, where quantum computers are available to possible attackers. We want to show how the issue of broken cryptography is being handled at the moment, and which technologies can help to increase the security.

The procedure, with which this essay was created, is literature research. Sources are researched and afterwards assessed, evaluated and correlated.

During the writing and research of this thesis, we were able to show that quantum computers are taking shape, and already have surpassed their conventional counterparts in certain use cases. IBM managed to build a 127 quantum bit processor [1] which shows the development of quantum computers is coming along. Steadily increasing the qbit count every year.
Meanwhile, the cryptographic community has worked tirelessly to produce algorithms that can withstand a powerful quantum computer. The standardization process of these algorithms, lead by NIST is coming towards an end in early 2022, they aim to release a rough draft in the first half of next year [2]. Furthermore, the standardization procedure will continue, yielding quantum safe algorithms.
But the change of one cryptographic algorithm to another cannot be performed without adequate tools and preparation. This is where we show how the idea of hybrid certificates can help to tie us over this period of change. It enables us to use old and new algorithms side by side until the infrastructure has been adapted to the new algorithms, phasing out the deprecated ones.
As we can see, the cryptographic landscape is about to change, and the chances are high that it will be more fluid in the future. We need to say goodbye to our rigid and static understanding of cryptography and embrace it's new fluid and agile nature. Crypto agility is the next spotlight of this essay in which we show how important it will be in the future to have an agile architecture, to be prepared for future algorithm changes.
Hardware security module manufacturers have also started to prepare for quantum computers. They develop solutions for post quantum cryptography capable devices. New exciting hardware and software solutions are being trialed. Thereby, they ease the transition from current cryptography to post quantum solutions.
Lastly, we could show how incredibly powerful and secure the technology of quantum key distribution is, but also highlight it's drawbacks. While it is a highly interesting technology, its costs and limited range, which only lasts up to approximately 200-400km, prevent it from being used in many use cases. But it manages to perform well in certain use cases, such as short distance communications which need high security.

Our recommendation, for companies of any scale, is to start cataloguing their infrastructure. It is crucial to know your assets and have a clear understanding of ones own architecture. Be this physical assets, software or the used cyphers. Only then, you will be able to prepare for the transition to a post quantum cryptography architecture. While the transition does not need to start immediately, it is advisable to keep a close eye on releases by NIST. Once standards and guidelines have been published, which should happen in a timely manner, it will be important to evaluate possible solutions and to start planning the future transition to post quantum cryptography. The choice of the most suitable solution is highly dependent on the existing architecture and the business needs of a company, and cannot be universally determined. It is thus crucial to create a tailored solution for every company. Our essay gives solutions applicable to different use cases.

Keybords: *algorithm, certificates, cryptography, ECC, NIST, QKD, quantum, RSA*

Executive summary

The National Institute of Standards and Technology estimates, that by 2030 quantum computers have become powerful enough to break a 2000 bit RSA encryption. While the projected date varies from research institute to research institute, they all agree on one common conclusion: Quantum computers are here to stay, and they are growing more powerful by the day, putting current day Internet encryption at risk of being completely broken!

With this essay, we provide an overview for employees of companies which are starting, or already performing the process of preparing their cryptographic infrastructure to be quantum resistant. Furthermore, we offer people who are generally interested in the topic the chance to get an overview over possible solutions for the quantum threat. We start with an introduction into quantum computing, by showing how far along the worlds leading researchers have come with quantum computers. The focus then shifts towards different solutions, to adapt modern cryptography to the post quantum era.

Next, we will delve into the realm of Post Quantum Cryptography. A competition of current cryptography researchers under the guidance of the National Institute of Standards and Technology. The next standard of NIST will consist of viable candidate algorithms to replace the ones, broken by quantum computers.

On quantum agility, we investigate the process of shifting your enterprises cryptography from a static one time "install and forget" task, to a periodically performed routine which enables your cryptographic solutions to be dynamic and agile. Thus allowing you to react to new algorithm changes, as they arise. This could become necessary if post quantum algorithms would fail like RSA.

A further addition is a closer look at hybrid Certificates. A technology which aims to bring Post Quantum Cryptography to internet traffic and enable the continuous use of Public Key Infrastructure. Helping to slowly move from old algorithms to new ones, while the underlying infrastructure is changing, by allowing for two certificates simultaneously.

Hardware Security Modules is another topic, which shines the light on implementing Post Quantum Cryptography to encapsulate whole Networks in quantum safe perimeters. This allows for devices, which might not be able to be adapted to quantum secure cryptography, to be secured. But it also is a promising approach for the increased workload of post quantum cryptography in high throughput and low latency environments,

such as core banking systems. And it could even help to enable quantum agility, harbouring the quantum proof algorithms of tomorrow.

Quantum Key Distribution is a technology that harvests quantum states to create impenetrable cryptography schemes. It offers interesting possibilities for provable secure data transmission, but has many drawbacks, with it's biggest one being it's extremely limited range, due to quantum mechanical properties.

It can seem like a daunting task to change the whole cryptographic landscape of a company. But our essay will show that not all is lost and that we still have some time to implement the changes. The changes will be extensive and thus require meticulous planning, but your security architecture will be stronger than ever, with the implementation of the right technologies.

This essay was written by Isaac Würth and Marco Zanetti. We are two bachelor students in the field of information technology at the Ostschweizer Fachhochschule in Rapperswil, short OST.
Isaac finished an apprenticeship as a system administrator. In his leisure time he competes in capture the flag hacking challenges and reads about current developments in the cryptographic field. This enables him to have a complete picture about the current cryptographic landscape.
Marco Zanetti is currently working as a junior security engineer at the Institute for Networked Solutions, where he helps prepare course material for courses such as "Cybersecurity basics". This provides him an insight into current developments of cryptographic algorithms and their use.

Aknowledgements

We want to thank everybody who supported us during the writing of this essay. While countless supporters have helped in their way, we wanted to single out a few for a special mention.

Nathalie Weiler, Professor for cybersecurity at OST Rapperswil. Mrs Weiler was our coach and main supporter during this work. She helped with great inputs, spot-on criticism and great ideas to help us improve our essay. We thank her for her determined mentorship and support for this essay, combined with the improvements she provided for us.

Kai Schramm is a security architect and our primary contact from our industrial partner. Mr. Schramm helped us with his expertise and knowledge of the current cryptographic landscape and kindly reminded us of requirements for the essay in the many times where we got lost along the way. We thank him for his enduring support and suggestions regarding the content of our essay.

And last, but certainly not least we want to thank you. Thank you, the reader of this essay, very much for taking your time to read what we have gathered over the last 3 months. This essay would be pointless without people reading it. And we sincerely-hope it was of use to you. Thank you.

Contents

1 Introduction

1.1 Purpose and Scope

The current cryptographic landscape is composed of many different ciphers, being used in a variety of applications, protocols and hardware. All of them play their respective part in the big system of secure communication around the globe. Most of modern cryptographic cyphers are based on mathematical problems, which are incredibly hard to solve for a computer, let alone a human.

But the recent advances in quantum computing pose a realistic threat to break a big portion of the asymmetrical cryptography that is currently in use. This would mean that a big part of our daily traffic would no longer be sufficiently protected from attacks. So NIST set out, to usher in a new era of cryptographic algorithms, with the main requirement at heart, that they need to be resistant to quantum computers.

This essay paints a picture of the current status of post quantum cryptography and the status of the current NIST evaluation for the standardization of said algorithms. Further, it provides an overview of the current technical implementations of the different algorithms and aims to provide advice for companies on how to proceed with this change. This is done by showcasing some of the important topics surrounding quantum secure communication.

1.2 Audience

This essay is directed at an audience less knowledgable in the field, as well as well informed experts.

Firstly we want to reach employees of companies, which just started the evaluation phase for post quantum cryptography. This essay offers an unbiased outlook for the coming post quantum cryptography transition. We provide an overview over how urgent the quantum threat is, what can currently be done to prepare for it, and what needs to be done down the line to transition to quantum secure communications. This is combined with an in depth analysis of current solutions that can be implemented.

If your company has not yet started to look into this field, this essay can offer you a broad overview over why the quantum threat is real, and what is done to make the

transition easier for you.

If you are in no way related to this kind of process, you can still read this essay. Some sections might be not of interest for you, but we nevertheless provide an overviews over all the basic aspects for the topics mentioned. Each topic then generally dives deeper into the details, the further you read along. This can help to gain a good understanding of post quantum cryptography for a bystander.

1.3 Document structure

The document is generally divided into different topics. Each topic usually starts with a non technical overview, and gets further into technical details, the further you read along. In the end of each topic is a conclusion section, which offers a personal opinion on the topic, written by the author of said section. A general conclusion over the whole situation regarding all the topics can be found at the end of the essay.

2 Current technical development

The following section will give you an overview for the different topics, which will be essential for your understanding of this essay. First, it will provide you with the general understanding for the underlying concepts at hand, and will then paint a picture of the current technical development in the field of quantum computing. This is important, so that you can see how the rest of this essay is relevant. The current technical development has a big influence on this essay, since quantum computers are being developed at a rapid pace. That means that topics discussed in this essay, which seem far out of reach, can become urgent within the coming years.

2.1 Status quantum computing

2.1.1 Declaration

> **Quantum computing** is a type of computation that harnesses the collective properties of quantum states, such as superposition, interference, and entanglement, to perform calculations. The devices that perform quantum computations are known as **quantum computers**. [3]

2.1.2 Fundamentals

Quantum computers are a completely new type of computers which cannot be compared to the conventional computers, which we use in our daily lives. While a conventional computer operates with classical bits, which can be either one or zero, the quantum computer works with quantum bits, so called Qbits These qubits are not binary, they instead have an amplitude of possibilities of being in the one state and the zero state at the same time. As long as they are not measured, they stay in this state of being one and zero at the same time. This is called a superposition. As soon as a measurement is taken, their binary value can be determined to either one or zero. At this point the super position gets destroyed. A comparison of a classical bit and a qbit can be seen on Figure 2.1.

A futher important aspect of the quantum computer is, that the different Qbits are linked to each other, based on quantum physics. This means that quantum computers can use these quantum physical laws to let the Qbits influence each other, to create high powered algorithms and logical gates. Such an algorithm is Shor's algorithm, which

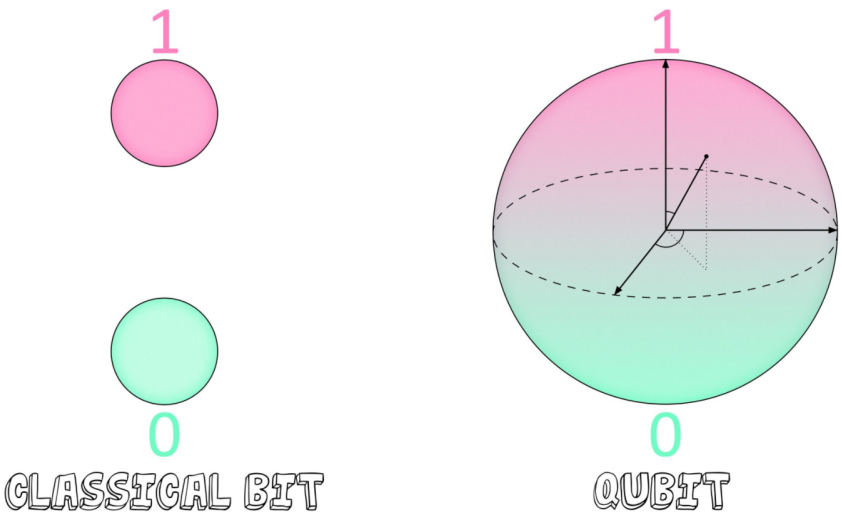

Figure 2.1: A classical bit compared to a qbit
[4]

was specifically made to break asymmetrical encryption.

2.1.3 Current technical development

But what exactly is the current technical development of quantum computers? The most important key players in this field are multi billion dollar companies in the information technology and electrical engineering field, such as Google and IBM IBM Quantum is fairly open about their process and releases press releases in regular intervals.

IBM is currently researching quantum computers at three different sites worldwide. The main branch is located in Zürich Switzerland. Their current statement regarding their goals is quite ambitious. They just recently announced the release of their newest addition to their quantum processor family. The processor called Eagle, has 127 stable Qbits which make it the first quantum processor of IBM which cannot be emulated on a classical computer anymore. [1] Their next goal is a 433 qbit processor, following in late 2022. The current objective is to then produce a 1000 qbit processor before the end of 2023. [5] You can see their current quantum computer depicted in Figure 2.2.

The interesting aspect of IBM's strategy is, that they aren't just testing this technology and keeping it close to their chest as Google does, but instead actively selling it. IBM currently fields the only commercially available Quantum computer, called the "IBM Quantum System One". It can be bought by big industrial partners of IBM and is installed at their sites. It can be fitted with all quantum processors of the first generation, up to and including the "Osprey" with 433 Qbits The system is currently being used in Europe, Northern America and Japan.

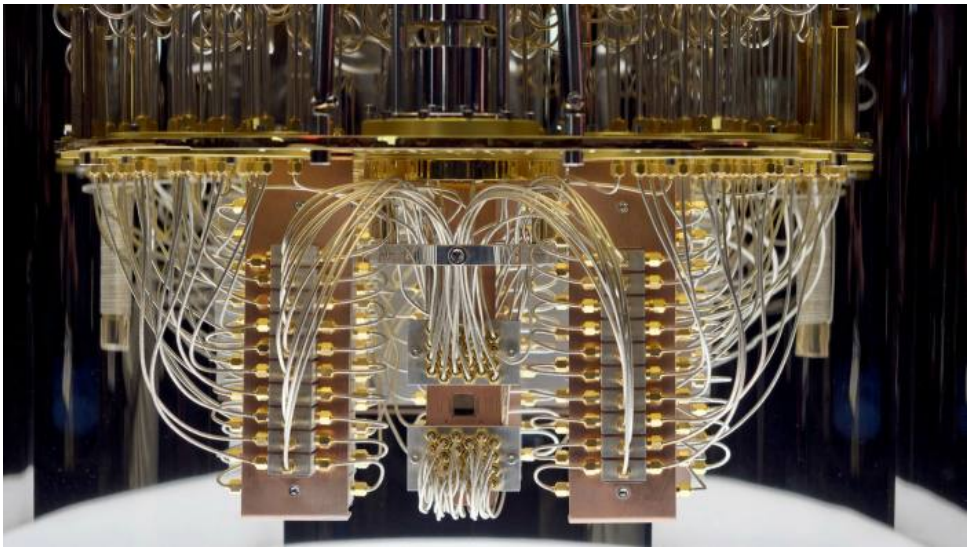

Figure 2.2: A current IBM quantum computer. The quantum processor can be seen bellow in the middle (small black square). The computer will be lowered into a liquid helium pool during operation for cooling purposes. [6]

2.1.4 Future development

IBM is actively preparing for the implementation of quantum computers with over a million Qbits But they did not yet state in which time frame they deem this endeavour possible. To show their dedication, they are currently developing the infrastructure for said quantum computer:

> That's why we're also introducing a 10-foot-tall and 6-foot-wide "super-fridge," internally codenamed "Goldeneye," a dilution refrigerator larger than any commercially available today. Our team has designed this behemoth with a million-qubit system in mind [5]

An image of their Goldeneye prototype can be seen in Figure 2.3. So as we can see they are quite persistent in their pursuit to develop a powerful quantum computer. This poses a realistic threat to our current cryptographical systems, as algorhitms would be vulnerable to those million qbit behemoths.

2.1.4.1 Asymmetrical Cryptography

Because of their special quantum mechanical properties, quantum computers in combination with Shor's algorithm are perfectly capable of cracking the discrete logarithm difficulty problem and the prime factorization difficulty problem efficiently.

If we take an RSA cryptosystem as an example, which derives it's complexity from the prime factoring difficulty, we can see the massive speedups the quantum computer provides for cracking it. While an attempt on a classical computer would encompass an exponential runtime to n, a quantum computer manages to perform the same task

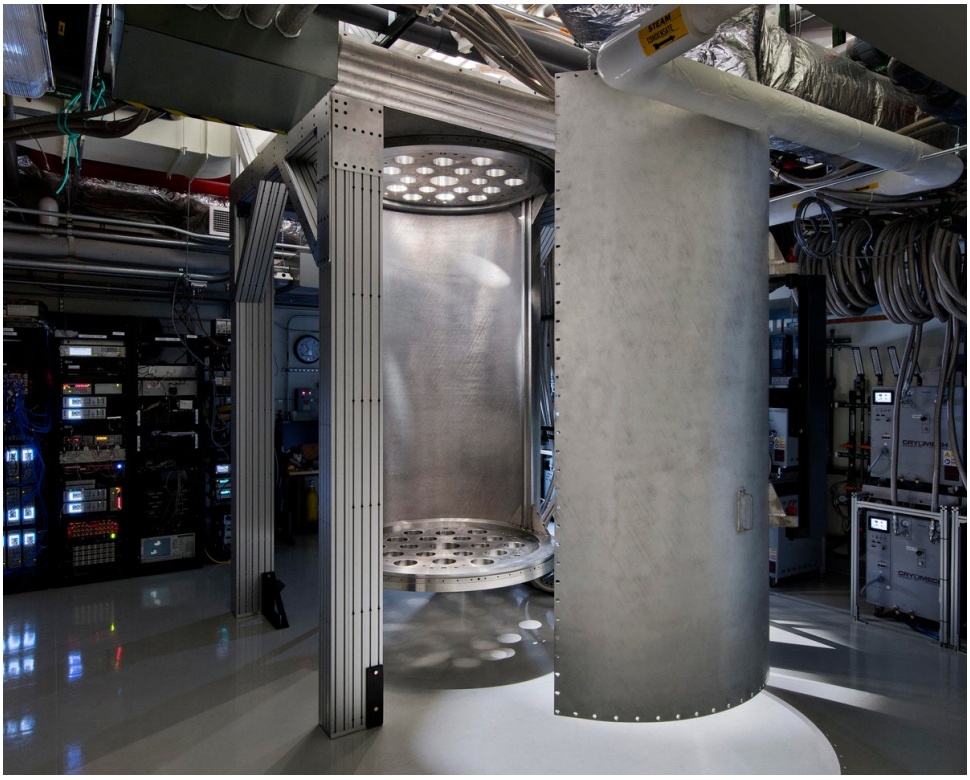

Figure 2.3: IBMs new dillution refrigerator prototype. Goldeneye. [7]

with only $O(n2(logn)(loglogn))$. This is a massive speedup as soon as n is chosen to be big enough, which it is in RSA encryption. So the breaking of an RSA key would now only require mere months, weeks or even days, compared to before, where it took a millenia, or more likely a couple trillion years. [8]

This poses a huge threat to those cryptosystems and basically renders them useless. So a big chunk of our current day cryptography becomes unusable if anyone manages to build a powerful quantum computer in the near future.

2.1.4.2 Symmetrical Cryptography

Symmetrical cryptography is also at risk from quantum computing. Grover's algorithm can achieve a substantial speedup for cracking these kinds of systems. But while a cracking attempt on a classical computer would run proportional to n, the same attempt on a quantum computer would take $\sqrt[2]{[n]}$. This may sound like a substantial speedup, which we are not going to diminish here, but as mister Baumhof put it nicely:

> I get a squared speedup in the quantum version, which is a fantastic speedup. If I can speed it up by 150 Trillion years, that is a lot of years, but its still obviously another 150 trillion years to break it, so that's not really too interesting [9]

So we can conclude that currently, quantum computing is no real threat for symmetrical algorithms. They do not provide a big enough speedup to pose any real danger, and

if big quantum computers come to be in the enar future, a doubling of the key sizes is already enough to mitigate the speedup entierly. This would pose an additional overhead and would thus reduce performance, but the development of classical hardware should be able to catch up in the meantime.

2.2 Post Quantum Cryptography (PQC)

2.2.1 History of cryptography

As discussed before, modern quantum computers will render asymmetric cryptography useless once they become sufficiently powerful. This will force us to change to a new generation of cryptographic algorithms in the future. WHile this seems like a daunting task, and it's by far no easy feat, the fact that we will need to change our cryptography is by all means not a new discovery. As previously discussed, this necessity arose back in 1994 when Peter Shor presented his algorithm to the public, and thus proved that one day this day would come. It became apparent that the current cryptographical algorithms could be broken, the only question remaining was if it would take a decade, a century or a millenia. So the scientific community started to analyze and design what the future of cryptography could look like. They took the algorithms in use back then, which don't differ much from todays cryptography, and analyzed if they could withstand attacks from quantum computers, or if they needed replacing. An example to illustrate, that this topic was already of high importance in the mid two thousands, is a book from 2009, titled:

> Is cryptography dead?
> Imagine that it's fifteen years from now and someone announces the successful construction of a large quantum computer. [10]

The title might sound a little dramatical, but the time frame was pretty solid. Quantum computers would become reality within a forseeable future, and cryptography would need to adapt, but how would that be possible?

So the work began, to design algorithms that would provide adequate protection against the almighty quantum computer. This work continued to span into the period of 2010. Progress was made towards the goal of quantum safe algorithms. Or even better, the scientific community distinguished algorithms that were already quantum safe, and improved them, to suit the needs of modern cryptography.

But why is this huge effort even undertaken, if the only reason for it is a vague "What if" scenario, which still seemed years, if not decades, away? Back in 2009, no one could say, if it was even possible to build a quantum computer. It seems a little hasty, to spend so much resources and time of the scientific community, which could have been put to better use, than to invest it in a scenario that might not even happen at all. And why didn't the community just use the already quantum safe algorithms that were in place, and simply waited for the urgency to arise for the implementation?
Mister Bernstein has three good arguments as to why this topic carries urgency.

- It takes time to implement post quantum cryptography, or any cryptography for that matter.
- It takes time to establish trust in the new post quantum algorithms.
- It takes time to improve the usability and performance of post quantum cryptography.

[10]

The scientific community realized this urgency early on and thus started to hold the convention for post quantum cryptography in 2006. The goal of that convention is to be an open platform for discussions about the topic post quantum cryptography, which they state on their webpage:

> PQCrypto is the main conference series devoted to post-quantum cryptography [11]

2.2.2 NIST involvement

While the scientific community made good progress during the last decade, it was clear that some form of official governing body would need to get involved at some point. This role was taken up by NIST when they joined the post quantum cryptography process in the year 2016. They started this by holding a first round of submissions for any willing applicants that wanted to enter the pool of contestants to become a recognized standard. Those algorithms were then subsequential evaluated and the pool of contestants was narrowed down. Initially, there were 89 applicants to the first round. The current round is the third iteration of this process and has only 7 contestants left, with 8 possible alternates. Some of them will be part of the new proposed standard by NIST. [12] If you want to know more about NISTs involvement, make sure to read the chapter 3.

3 NIST

As shown in the previous chapter, asymmetric cryptography will soon be fundamentally broken. But NIST has set out to pool all feasible candidates for new cryptographic algorithms and standardize the most promising ones, for a quantum secure future. This chapter will give an overview over how their process to achieve this is structured, and what they did up until now. It will look at the candidates in detail and show what will be done in the near and far future to standardize them, and subsequently roll them out as the future of asymmetric cryptography.

3.1 Introduction

NIST is the National Institute of Standards and Technology of the United States of America. It has a role as standardizing body for technological advances in the US. Thus, it is heavily involved in the research, development and implementation of IT solutions for the wider use across the U.S., and in most cases, also across the world. NIST was the leading force behind the new AES and SHA implementations [13], which it spearheaded with competitions to find reliable algorithms, before putting them to the test and deciding which would be standardized and adapted for public use.

So it was only to be expected that NIST would also lead the efforts to change the cryptographic landscape once more, with the looming threat of quantum computing. They realized the threat that quantum computers posed, and started their process for the search of new algorithms. In 2016 [14], they issued a directive, that asked for the submission of current PQC algorithms that were being developed at the time. NIST set out rules and guidelines of what each submission should encompass. By the end of the submission period, which came in November 2017, they had received 82 candidates, of which 69 met the stated requirements for formal submission. [15] The first round of evaluations was thus started in late December 2017.

NIST analyzed the candidates for over 2 years regarding their security, performance, and other characteristics. In that time it looked at different requirements for which they tested how well the candidates would fulfill those duties. They then release an updated list with 26 remaining candidates [15], that had passed the first round of evaluation in January 2019. It stated that these algorithms would be selected for a further round of evaluation.

The second round was launched in April 2019, giving the selected candidates committees enough time to alter their submission and tweak them slightly if they wished to do

so. After this period had run out, NIST set out once more to prove the viability of the remaining algorithms. This time being more thorough and more precise in their efforts to determine which of them were ready to replace current asymmetric cryptography. NIST announced the finalists of the second round, which would move on to the third round in July 2020. The number of viable algorithms had dwindled to 7 finalists and 8 alternate candidates. While the 7 finalists would directly advance to the third round and be considered a viable option for becoming part of the future standard, the eight alternate candidates lacked this confidence, and would thus be subjected to at least 2 more rounds of testing and developing (round three and four). They will not be included in the first released standard, but NIST made it clear that it aims to diversify its portfolio of possible algorithms and aims to keep a wide selection of different candidates. [2]

This brings us to the third round of NIST evaluation. NIST offered the selected finalists once more the opportunity to optimize and tweak their submission until the deadline of October 1st 2020. Once that passed, they started the third round of evaluations, with the clear goal to produce a standard at the end of that phase. The phase has been ongoing ever since and is currently not completed as of writing. The final date set for an announcement is not defined, however NIST stated in an earlier presentation:

> The 3rd Round will end sometime close to the end of 2021 [16]

3.2 Closer look at PQC

The previous section contained the following statement: "...NIST made it clear that it aims to diversify its portfolio of possible algorithms and aims to keep a wide selection..." This is due to the fact that the new proposed algorithms don't simply follow one hard math problem to create secure cryptography, but use a variety of them. The following section will take a closer look at the different types that are currently being evaluated for the next generation of cryptography.

So, NIST is discussing a variety of different candidates for PQC solutions. They do this to diversify the portfolio of possible algorithms that can take over and ensure a quantum safe communication in the future. The field of quantum secure cryptography is still in it's relative infancy, and while some algorithms might be promising as of now, it is not unheard of that a possible weakness could be found in the future, be it with a new quantum or classical algorithm. If all new algorithms would be of the same underlying technology, it would render all of them useless, and the process would need to start from the beginning. This is why NIST is trying to diversify the portfolio, to make sure this will and cannot happen in the future. In the case that one of the new algorithms would be deprecated, another one with a different underlying mathematical problem could take it's place instead.

This is achieved by basing the different candidates on substantially different theoretical approaches to post quantum cryptography. There are four main categories of algorithms, which are currently in contention, and this chapter takes a closer look at the

individual theoretical approaches, to give you an overview. Please note that these four categories are by no means exhaustive, as there are countless approaches to PQC solutions. But to narrow it down, this paper will only touch on the four most widely used ones. [8]

3.2.1 Code-based PQC

Code based PQC is based on error correcting codes. These are used to help transmit data over channels which introduce bit faults and thus need correcting algorithms to reliably transmit data. There are many ways on how this can be implemented, from sending multiple bits as backup, to checksums that check the data integrity and alert the receiver, if the transmitted data is faulty, to using so-called binary Goppa codes[1]. These leverage mathematical properties of matrices to create error correcting codes. These only necessitates a small overhead and promises strong, and depending on the size of the matrix, adjustable fault tolerance. Now there is the possibility to base a cryptographic system off of this premise. [8]

One of the oldest ones is the McEliece crypto system, which was developed in 1978 by Robert McEliece. It leverages the basis of the last example with matrices. But instead of using the matrix to correct errors in the code, the McEliece system introduces systematical errors by scrambling the message with a public key matrix. Now the receiver needs to unscramble said message with his private key matrix to receive the message. Calculating such a result without sufficient information is an NP-hard problem[2], which makes it resistant to quantum computer attacks. [8]

NIST Candidates:

- **F** [3] Classical McEliece
- **B** [4] BIKE
- **B** HQC

Advantages: [17] [18]

- \+ Provable unbreakable by quantum computers
- \+ Has not suffered any security degradation at all since it's development
- \+ System is in development since 40+ years and thus well understood.
- \+ System is quite fast, as encryption and decryption have low complexity.

Disadvantages: [8] [18]

- − Big key sizes of up to 1 MB (RSA) in comparison, has 2 KB)

[1] Binary Goppa codesare a special family of error correcting codes, which operate with matrices.
[2] NP-completeness describes a class of problems which a certain complexity. You can find more about it in this Wikipedia article
[3] F = Finalist
[4] B = Backup candidate

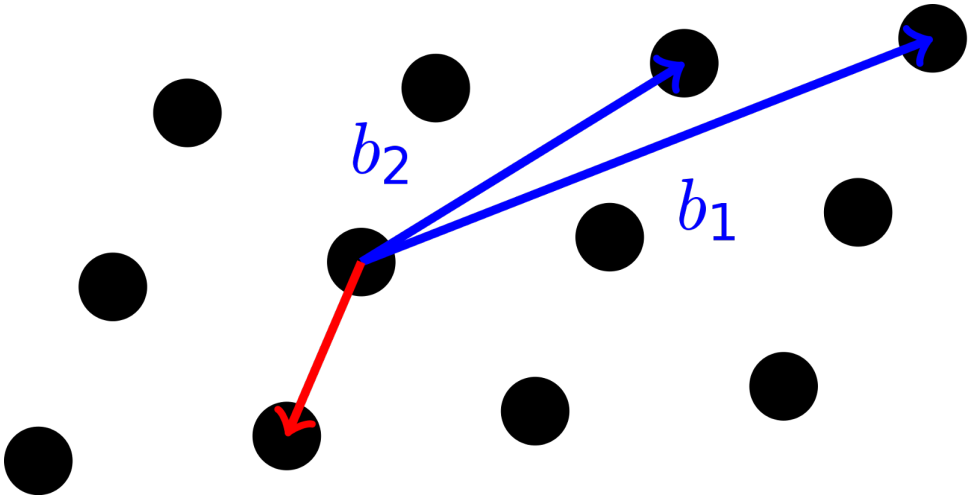

Figure 3.1: An example of the Shortest Vector Problem in a 2 dimensional lattice. [19]

- Security and efficiency propose a known tradeoff issue.
- Require a large amount of memory.

3.2.2 Lattice-Based PQC

Lattice based cryptography uses specific lattices. The mathematical lattices used in this process consist of points in an n dimensional space. A simple example for this is if $n = 2$ which means it's simply a grid with points in a 2 dimensional space. Now, how can we use lattices to create a cryptographic algorithm? This is where the Closest Vector Problem (CVP, or also called Shortest Vector Problem (SVP)) comes into play. CVP puts a point at a specific location within the lattice and now poses the question of finding the shortest vector to the nearest point. An example of this can be seen in the Figure 3.1. While this may seem trivial in 2 dimensional space, lattice based cryptography usually does not operate with just 2 dimensions. The lattices can have as many dimensions as the algorithm desires, which are completely impossible to visualize for the human mind. [8]

The most promising implementation of lattice based cryptography is NTRU. NTRU has been developed in 1996 and has since been improved and updated, even implementing the system under an open-source license since 2011. Two of the candidates currently in the competition for PQC by NIST are directly based on NTRU.

NIST Candidates:

- **F** Kybers
- **F** NTRU
- **F** Dilithium
- **F** Falcon
- **B** NTRU Prime

B FrodoKEM

Advantages: [20] [17]

+ There are currently no quantum algorithms which would reduce lattice problems hardness.
+ Offers a big variety of implementations with the same underlying system (As seen in number of candidates).
+ Due to this characteristic, they can be tuned for different key goals (Performance, key sizes, cypher size, etc...)
+ Most versions are comparably easy to implement.

Disadvantages: [8] [18]

− Possibility that only worst-case scenarios are secure, while average scenarios remain relatively easy to crack
− Possibility that approximations are easier than previously thought, undermining security

3.2.3 Multivariate PQC

This type of cryptography has a lot of similarities with breaking multivariate equations. They are usually combined for a system, which contain many equations, that are hard to find the solution to. This is an NP-hard[5] problem, which is called multivariate quadratic equations' problem (MQ). The keys in this crypto system would therefore consist of a multitude of equations. Without these equations the posed MQ would be incredibly hard to solve. [8]

NIST Candidates:

F Rainbow

B GeMSS

Advantages: [21] [8]

+ Efficiency and thus performance
+ Low computational requirements
+ Short signatures

Disadvantages: [21] [20] [8]

− Implementations can prove difficult
− Large public keys

[5] NP-hardness describes a class of problems which a certain complexity. You can find more about this in this Wikipedia article

- Security isn't sufficiently proven for large-scale implementation
- Hardware intensive and thus slower

3.2.4 Hash-Based PQC

Hash based cryptography is, as it's name suggests, based on well established hash functions, such as SHA2, SHA3 and variations of them such as SHAKE256 This is possible, because hash functions in itself are already quantum secure. They are secure by having an attacker randomly guess for hash collisions, which is not possible to speed up on quantum computers (Until now no quantum algorithm has been developed or at least openly published that could break hash functions). [22] One of the main downsides of using hashing algorithms is their very specific use. Due to the characteristic of them being a one way function, they can only be used for signing purposes and a very restricted set of cryptographic functions which necessitate certain requirements. [8]

NIST Candidates:

B SPHINCS+

Advantages: [18]

+ Uses relatively small key and signature sizes.
+ Nearly every hash functions can serve as basis for hash based signature schemes, which offers flexibility.
+ Hash scheme can be adapted to hardware and offer performance increases.

Disadvantages: [8] [18]

- Usually operates using a one time password, necessitating a new key for each transmission
- Not yet proved to be secure
- Only applicable for signatures
- Only applicable for short messages

3.2.5 Overview

Finalists	
Public-Key Encryption / KEM	Digital signatures
Classical McEliece	CRYSTALS-DILITHIUM
CRYSTALS-KYBER	FALCON
NTRU	Rainbow
SABER	
Alternate candidates	
Public-Key Encryption / KEM	Digital signatures
FrodoKEM	GeMSS
SABER	Picnic
HQC	SPHINCS+
NTRU Prime	
SIKE	

Table 3.1: An overview over the round three NIST candidates [2]

3.3 Benchmarking

3.3.1 Introduction

One of the most important aspects for a new PQC algorithm that wants to be the next standard is, besides security, performance. [2] The algorithm needs to be impeccably secure, but it also needs to be performant to be competitive in the current competition. This is underlined by the fact, that past selections of algorithms by NIST usually resulted in the selection of schemes that were under the most performant ones early on, for example Rijndael(AES) and Keccak(SHA3). [13]

Due to this, this chapter will give an overview over the current finalists of round three, in regard with their performance characteristics. This may seem trivial, as benchmarks are something done for nearly every computer program. But since this is a competition that determines the future of cryptography, there is considerably more at stake. This results in a huge amount of papers and reports written about this topic, for software and hardware benchmarks, over a variety of platforms, from ARM(Acorn RISC(Reduced Instruction Set Computer[6]) Machines) Cortex-M4[7] controllers, over common consumer

[6] A **reduced instruction-set computer** (**RISC**) is a computer designed to simplify the individual instructions given to the computer in order to realize a task. [23]

[7] The **AES Cortex-M** is a group of 32-bit RISCAES processor cores licensed by Arm Holdings. These cores are optimized for low-cost and energy-efficient integrated circuits, which

computers up to connected network machines. To go into detail for all these papers would simply be impossible in this chapter, so the most interesting ones (by our personal, and biased choice) are linked below for further reading. They are categorized into topics that might be of interest for you.

- Testing and Benchmarking NIST PQC on AES Cortex M4 [25]
- Evaluations of second round candidates on IoT(Internet of Things) devices [26]
- Implementations of the second round candidates on dedicated hardware with the help of FPGAs (Field Programmable Gate Array [8]) [13]
- NIST Post-Quantum CryptographyA Hardware Evaluation Study [27]
- Benchmarking in TLS implementations [28, p. 72]
- Post-Quantum Authentication in TLS 1.3 [29]

For the sake of this essay, it was important to us to find an unbiased benchmark from an independent source, that could provide datasets for all finalists, which turned out to be not too easy to find. Because many papers only contain a subset of algorithms due to the sheer amount of them, which often leads to some finalists being left out due to the fact that back when these papers were written, they weren't finalists yet, but just another algorithm in the second round pool. But the webpage of eBACS(ECRYPT Benchmarking of Cryptographic Systems) offers a huge variety of datasets, relating to all possible kinds of algorithms, including PQC KEMs and signatures. It was the only set we found, which contained all seven finalists for a complete comparison.

3.3.2 Information about the dataset

The dataset from the webpage eBACS is managed by the Virtual Application and Implementation Research Lab (VAMPIRE). It contains more than 150 different implementations of cryptographic algorithms (Including some PQC candidates) on almost 60 different architectures, with 3-4 parameters per set. This results in an estimated twenty thousand datasets for benchmarking, at the time of this writing. For the simplicity of this essay, we will constrain the following list to the seven current finalists, listed in Table 3.1, with similar security levels[9]. Sadly it was not possible to compare the algorithms on similar hardware, due to some analyzed datasets missing one of the finalists. So the hardware used for the KEMs and the signatures are not the same. But, and this was most important to us, all KEMs were tested on the same hardware and all signatures were tested on the same hardware. A further interesting aspect of this analysis would have been the comparison with a non quantum secure algorithm which is currently in use, such as ECC or RSA But sadly, our chosen datasets did not include such a comparison, due to the fact that they mainly targeted NIST candidates.

have been embedded in tens of billions of consumer devices. [24]

[8] FPGAs are integrated circuits which allow for a change of their architecture with the help of software, thus providing flexibility.

[9] Security levels as defined by NIST [30]

The architecture for the KEM benchmark selected for the visualization was the

> **aarch64; Firestorm (610f0230); 2020 Apple M1; 4 x 3200MHz; unstable; minimac,** [31]

with the help of the benchmarking tool version

> **supercop-20211108** [31]

While the signatures were benchmarked on the

> **amd64; TigerLake (806c1); 2020 Intel Core i7-1165G7; 4 x 2800MHz; unstable; pascalinspiron75062n1,** [32]

with the help of the benchmarking tool version

> **supercop-20210125** [32]

The security levels indicated by the Roman numerals, ranging from one to five, are according to NIST requirements, shown in the Table 3.2. [30] They indicate NIST requirements, compared to previous algorithms standardized by NIST. Rules I, II and V compare the strength of the algorithm to various implementations of AES with an exhaustive key search[10], with I being the weakest and V being the strongest. Rules II and IV compare the strength of the algorithm to various implementations of SHA with a collision search[11], with II being the weaker and IV being the stronger one.

Level	Level Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

Table 3.2: The NIST security levels [30]

So following, you will find a visualization of the size of all KEMs and their performance, as well as the size of the signatures and their respective performance. Please keep in mind that all scales are **logarithmic**, this was necessary due to the huge differences in size and performance of the different algorithms. The images have the whole description in their caption, to avoid any confusion if the images end up in a different position that originally intended by the authors.

3.3.3 Visualization

Following, you will find the figures correlating the data taken from the benchmarks and visualized together. We grouped all KEM's and signature schemes into one figure each, with size and speed being the two other factors, leading to four figures in total. All figures contain additional comments in their caption.

[10] An exhaustive key search is a brute force attempt at guessing every possible key until obtaining the correct one.

[11] A collision search is a brute force approach to generating hashes until a collision is found. Meaning an equal hash.

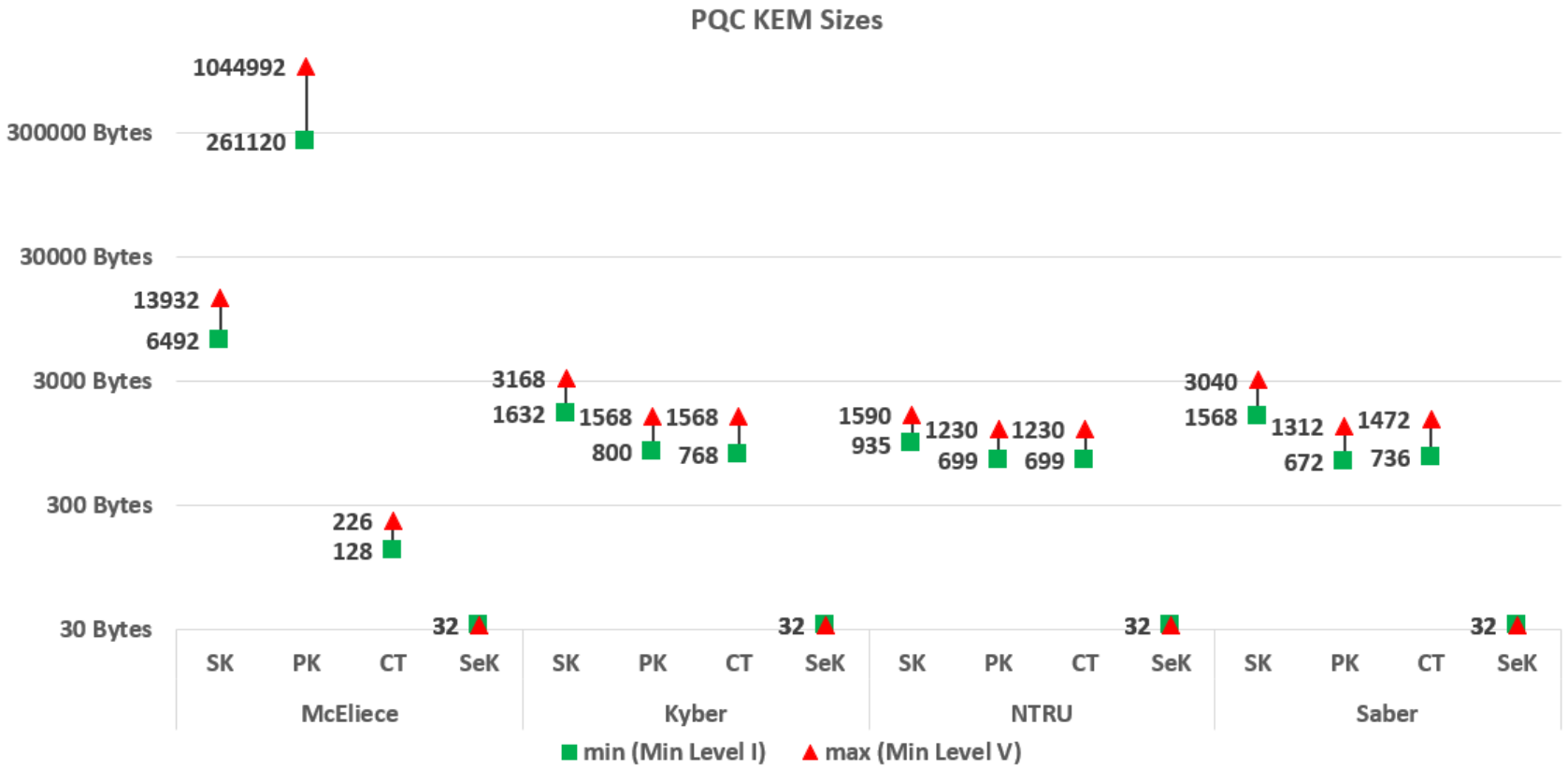

Figure 3.2: SK = Secret key, PK = Public key, CT = Cypher text, SeK = Session key
The minimum is the minimal size achieved by the algorithm
The maximum is the minimum size achieved by the security level V of the algorithm
It is clearly visible that all lattice based contenders are close together regarding key sizes, while classical McEliece clearly stands out with its big secret key, and its huge public key. The session keys are exactly the same for all KEM schemes. Data correlated from sources. [31] [33] [34] [35] [36] [37]

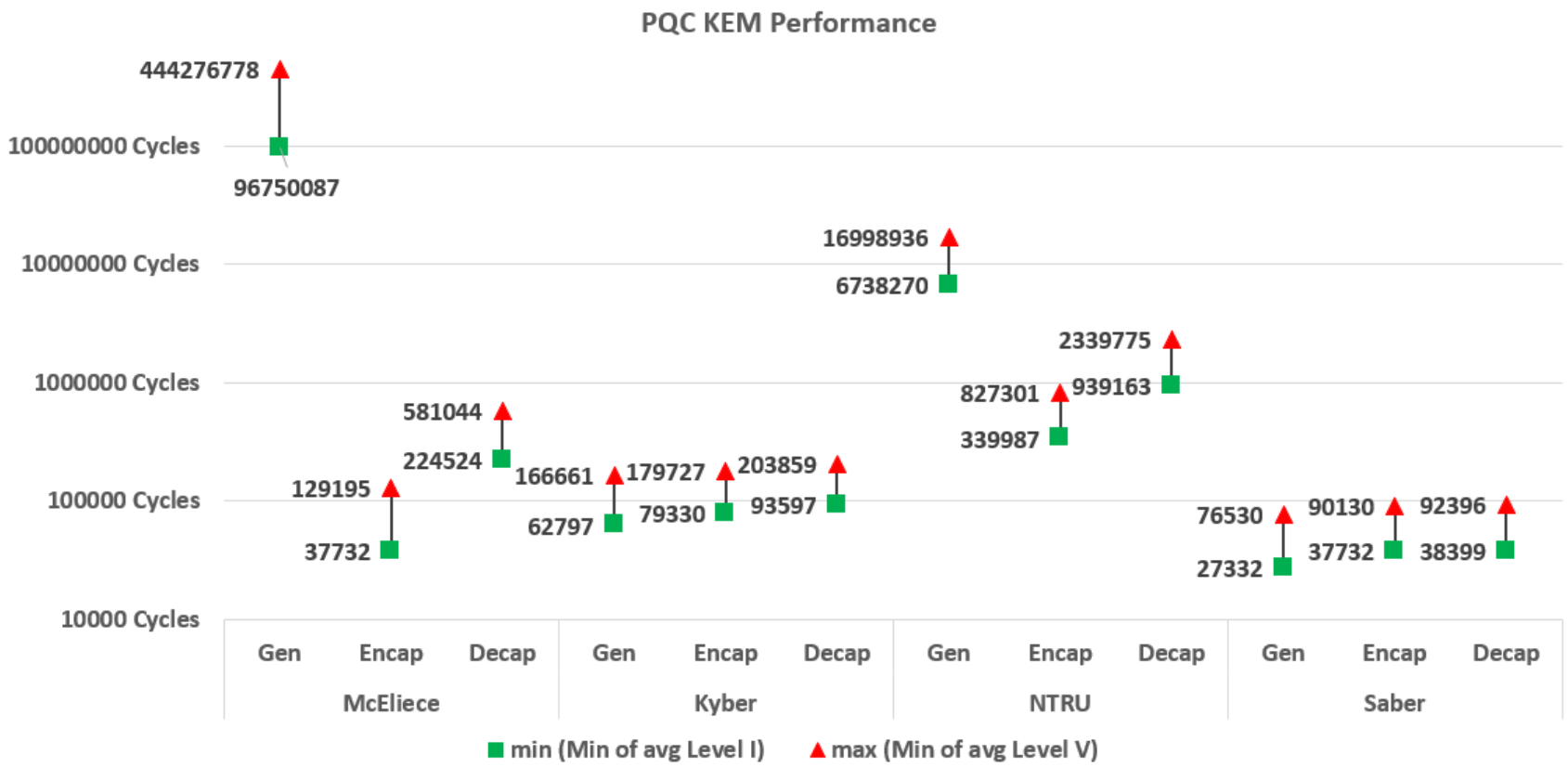

Figure 3.3: Gen = Generate key pair, Encap = Encapsulation, Decap = Decapsulation
The minimum is the minimal cycles used by the algorithm
The maximum is the minimum cycles used by the security level V of the algorithm
While Saber and Kyber clearly have the best performance overall, it shows that the NTRU implementation is already quite a lot slower in comparison. The classical McEliece is comparable to the first two schemes in regards to encapsulation and decapsulation, but the key generation takes substantially longer than any other scheme. It takes roughly 10times as long as its second competitor and 1000 times longer than Saber or Kyber. Data correlated from sources. [31] [33] [34] [35] [36] [37]

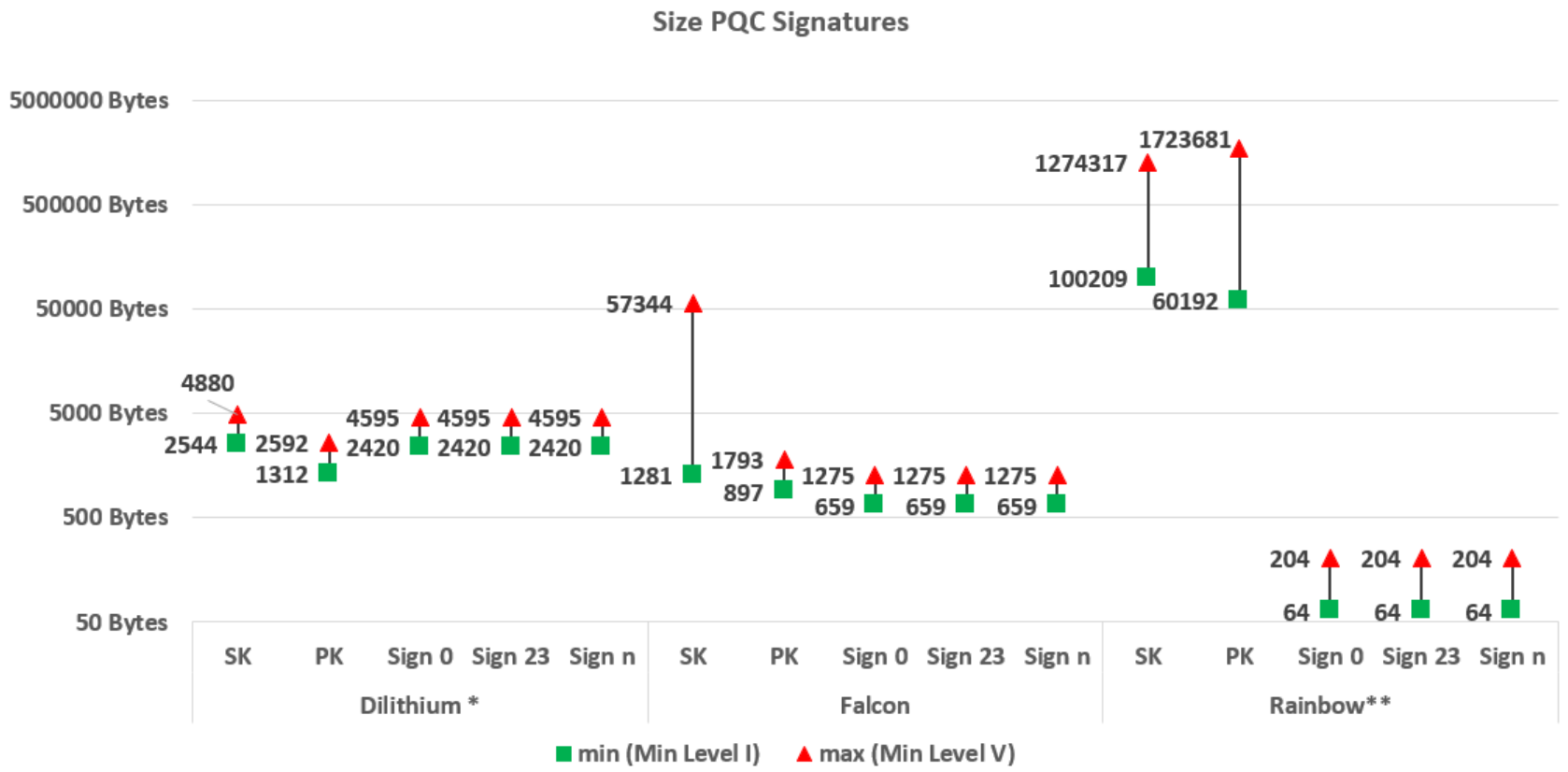

Figure 3.4: SK = Secret key, PK = Public key, Sign 0 = Signing 0 bytes, Sign 23 = Signing 23 bytes Sign n = Signing many bytes (No exact specification given)
The minimum is the minimal size achieved by the algorithm
The maximum is the minimum size achieved by the security level V of the algorithm
* = Dilithium has no security level I implementation, so the Dilithium values are from security level II
** = The Rainbow documentation does not clearly correlate security levels to implementations, so these implementations are not to be taken as precise values, but as approximations
As we can clearly see, all schemes operate with similar sizes, the only clear deviations are the scalability of Falcon, which seems to be slightly worse than its competitors, and the key sizes of rainbow which are substantially bigger than others. But as a benefit, the signatures of rainbow are considerably smaller than its competitors. Data correlated from sources. [32] [38] [39] [40] [41]

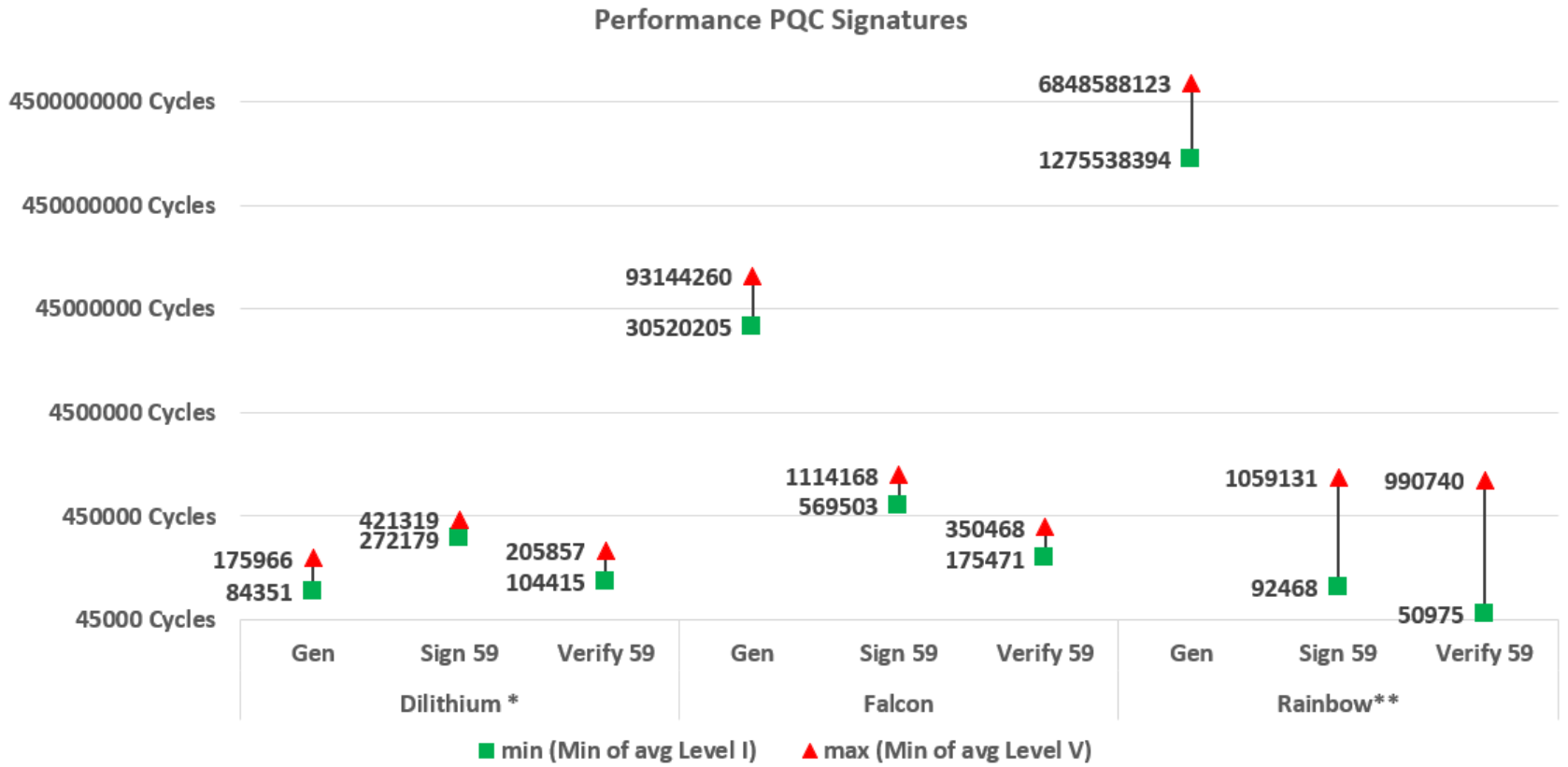

Figure 3.5: Gen = Generate key pair, PK = Public key, Sign 59 = Signing 59 bytes, Verify 59 = Verifying 59 bytes
The minimum is the minimal cycles used by the algorithm
The maximum is the minimum cycles used by the security level V of the algorithm
* = Dilithium has no security level I implementation, so the Dilithium values are from security level II
** = The Rainbow documentation does not clearly correlate security levels to implementations, so these implementations are not to be taken as precise values, but as approximations
It's visible that all schemes operate in a similar performance range, with the only clear deviations being Falcons and Rainbows key generation performance. While Falcon takes roughly 100 longer than Dilithium, it takes Rainbow **10000 times as long!** Data correlated from sources. [32] [38] [39] [40] [41]

3.4 Conclusion

Disclaimer

Everything included in this chapter (Conclusion) is merely an opinion by the authors and is not supported by hard evidence or direct quotations of relevant literature. The opinion is based on the research documented in the previous chapter.

The current status of NIST standardization progress is highly promising to produce a first draft of a standard by the beginning of 2022, with a definitive standard by the end of 2024. This perfectly fits in with the communicated NIST goals for the PQC process. There will probably no quantum computer by 2024 which can break classical cryptography. But, and this is a big variable in this process, the process of changing the worldwide asymmetric cryptography merely starts with the selection of a standard. This process can take from a few years up to a decade or more, depending on many factors, such as the age of the system, the urgency for change and the financial power of the key players, to name a few. While the urgency is as high as it has ever been, due to the fact that quantum computers are expected to break RSA with a 50% chance by 2030, we need to ask ourselves what happens if the quantum threat does not materialize as early as expected. Will the changes still be made if the development of quantum machines should plateau in the next couple of years? Or will the transition come to a standstill?

Only time will tell, as we will need to wait how the industry will react to the new standard. Big companies with highly crucial data, such as financial institutes or companies with valuable trade secrets in transit will be first adapters, together with government agencies and intelligence agencies. But how fast will the rest of the technological landscape follow their example?

The NIST has done everything in its power, with a well-structured and coherent process that has produced a small but strong selection of future algorithms for public use. The last contenders are a good choice, with a mix of variety and reliability. A mix of known processes and new endeavors. We are excited to see what NIST will propose at the beginning of 2021. The same goes for the possible round four, which could already start in late 2022 and produce even more possible candidates to make sure asymmetric cryptography is secure for the foreseeable future.

3.5 Look ahead

- End of third round: end of 2021 or early 2022
- First draft of standards: early 2022
- Round 4 for backup candidates: late 2022-2024
- Final standard: end of 2024

[16]

3.6 NIST candidates documentation

If you want to know more about the specific NIST candidates, we highly advise you to read the documentation which has been released by NIST. It contains a short overview over all current finalists and all backup candidates. It goes into detail how their basic underlying functionality works and says why NIST considers them a strong contender in the field of PQC You can find the reports regarding the standardization of PQC on NIST page for the status update concluding the second round.

Furthermore, you can find a detailed overview for the ongoing process of the third round on the page for the announcement for the third round.

And last but not least, they provide an overview over all the updated submission for the third round on the page for the third standardization conference.

4 Certificates

4.1 Introduction

In this section, we introduce digital certificates and their usage in today's technology. After the introductory paragraph, we present hybrid certificates, composite certificates and parallel hierachie, which help transition to post-quantum cryptography.

4.1.1 Definition

In this section, we are going to define the terms hybrid certificate. First, we will determine what the words hybrid and certificate are.

Hybrid

In biology, this term means a plant or animal has been produced from two different types of plants or animals, especially to get better characteristics.[42] The term is often used for improvements or adding functionality to a product in technology. A traditional certificate has extra fields for an alternative certificate. Later more on that.

Certifacte

A certificate refers to an official document stating that thing is true. In society, we often do this to provide truth for different things, such as birth, share or skills. Here we are talking about digital certificates used for providing integrity and confidentiality, and especially we are talking about the X509 standard. [43]

Composite

It means that something is made up of several parts or elements.[44] In the context of certificates, we are not using one algorithm, but several of it.

Public-Key cryptography

It is a generic term for asymmetric encryption that uses two keys, also referred to as pairs of keys for encryption. The particular property in this system is the one-way mathematical function. A message can be encrypted with one part of the keys but not decrypted. The public key is available for everyone, used to encrypt the message. The private key is to decrypt the received message. [45]

Hybrid cryptosystems

This definition clarifies the difference between hybrid certificates and hybrid cryptosystems. Besides the terms, symmetric-key cryptosystems and asymmetric-key cryptosystems are also called hybrid cryptosystems. By only using asymmetric or asymmetric cryptosystems, we are facing different problems. The asymmetric approach sometimes has the problem of performance, and the asymmetric is the main problem of key exchange.

To understand the concept, we are going to make an example. As always, we have lovey bob, Alice and eve for showing demonstrations.

Alice wants to send a confidential message to bob over a single unsecured channel. In this example, we are not determining bobs public key is from him. Eve has, of course, the possibility to send a fake certificate and make a man-in-the-middle attack. [46]

1. Alice asks Bob for his public key and make a verification of the key with PKI (She could also ask a third party, e.g. PKI for the public key)
2. Alice generates an encryption key for the message.
3. Alice encrypts the message with the encryption/symmetric key with a symmetric algorithm, e.g. AES-128.
4. Alice encrypts the key-encryption key with the public key. She knows artefacts, the ciphertext and the encrypted encryption/symmetric key.
5. Alice send the artefacts to bob.
6. Bob uses his private key to decrypt the encrypted encryption/symmetric key (At this point, Bob was the only one able to decrypt the message because of the public-private key approach).
7. Bob decrypts the message with the decrypted encryption/symmetric key.

This approach includes **key encapsulation**, which is the public-key cryptosystem and **data encapsulation**, which is the symmetric key cryptosystem.

In terms of hybrid certificates, the improvement is in the key encapsulation.

Cross-signed certificate

The concept of cross-signed certificates refers to verifying a certificate on different paths along the PKI system [47]. For example, Let's Encrypt certificates have different trust paths for verification. From the application point of view, it makes no difference which path for verification is used, as shown in the figure 4.1. If you like to read more about it, take a look at the blog post from Scott Helm.

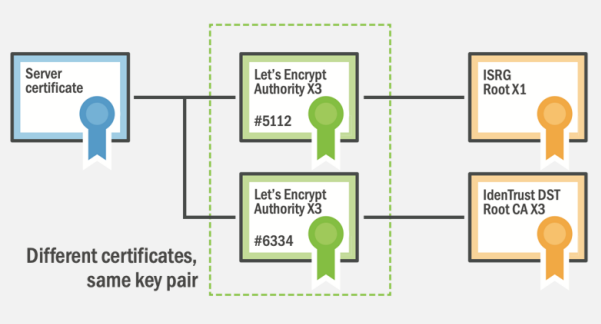

Figure 4.1: Let's Encrypt Cross-Signed Certitifcate [48]

PKI and PKIX

Refer to two different things. The term PKI is used for the technology Public Key Infrastructure and is defined RFC5280 PKIX is the IETF working group and establishes the standard and the worked-out standards are often referred to as "PKIX standards".

4.2 X.509

This section will learn how an X.509 certificate is structured. To fully understand the following sections, it's helpful to understand ASN.1 and make it easier to compare the actual structure of version 3 with the approaches.

There is no standard defined how to solve the problem of the migration to new PQC Certificates.

4.2.1 History

For your interest, we give you a short overview of the history of X.509 certificates [49].

It was the first issue on 3 July 1988 associated with the X.500 standard.

X.500 is a communication protocol and first approved in 1988 and enhanced in 1993. It's used in directories services and specifies a client-server architecture. The protocol for directory services is better known as Lightweight Directory Access Protocol (LDAP). Security is defined under the X.509v1 standard and nowadays deprecated. The first usages were a hierarchical structure for accessing resources using asymmetric encryption.

Version 2 introduced the concept of subject and issuer unique identifiers for the reuse of the certificate using the subject and/or issuer.

Version 3 was approved in 1996 and included the concept of extensions. Some extensions are defined in the RFC, and others can be user-defined.

4.2.2 ASN.1

The X.509 Standard is written in ASN.1 annotation because the standard is from the year 1988, and ASN.1 was the way to go. So we first need to understand ASN.1 for the following parts. For simplification, we will give you the code snippets in JSON. It should help you to understand it fully.

Let's begin with describing Abstract Syntax Notation One.

Abstract Syntax Notation number One is a standard that defines a formalism for the specification of abstract data types.

It gives a language to describe the content, but it does not provide effective encoding for transferring data. The encoding we are talking about is, as an example, JSON or XML. So ASN.1 is more a mother of encoding and provides a way to describe the content universally and encode the content in the following schemas. Basic Encoding Rules (BER), Packed Encoding Rules (PER), XML Encoding Rules (XER), JSON Encoding Rules (JER), and others. [50]

If you like to understand ASN.1 further, we can recommend this video.

4.2.2.1 Example

This example should make clear what ASN.1 is [51]. We like to send the phone number of John to other people, but we only know which encoding is used and know the schema.

Listing 4.1: Exmaple definition phone and name

```
1 Contact ::= SEQUENCE {
2     name VisibleString,
3     phone NumericString
4 }
```

We defined the schema, but we have different machines using a different encoding. There we need to use different encodings.

Basic Encoding Rules (BER)

Listing 4.2: ASN.1 Encoded in BER

```
1 30 19  80 0A 4A6F686E20536D697468  81 0B 39383720363534333231 30
```

Packed Encoding Rules (PER)

Listing 4.3: ASN.1 Encoding in PER

```
1 0A 4A 6F 68 6E 20 53 6D 69 74 68 0B A9 80 76 54 32 10
```

XML Encoding Rules (XER)

Listing 4.4: ASN.1 Encoded in XML

```
<?xml version="1.0" encoding="UTF-8"?> <Contact>
<name>John Smith</name> <phone>987 6543210</phone> </Contact>
```

XML Encoding Rules (XER) in HEX

Listing 4.5: ASN.1 Hex Representation of a XML

```
3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31 2e 30 22
20 65 6e 63 6f 64 69 6e 67 3d 22 55 54 46 2d 38 22 3f 3e
20 3c 43 6f 6e 74 61 63 74 3e 20 0a 3c 6e 61 6d 65 3e 4a
6f 68 6e 20 53 6d 69 74 68 3c 2f 6e 61 6d 65 3e 20 3c 70
68 6f 6e 65 3e 39 38 37 20 36 35 34 33 32 31 30 3c 2f 70
68 6f 6e 65 3e 20 3c 2f 43 6f 6e 74 61 63 74 3e
```

JSON Encoding Rules (JER)

Listing 4.6: ASN.1 Complex Example

```
{ "name" : "John Smith", "phone" : "987 6543210" }
```

Listing 4.7: ASN.1 Hex Representation of a JSON

```
7b 20 22 6e 61 6d 65 22 20 3a 20 22 4a 6f 68 6e 20 53 6d
69 74 68 22 2c 20 22 70 68 6f 6e 65 22 20 3a 20 22 39 38
37 20 36 35 34 33 32 31 30 22 20 7d
```

We see above the most human-readable encoding is JSON and the XML. This different encoding makes it independent from programming languages, protocols and operating systems. The opportunity makes it possible to choose the correct encoding for an application. For example, you usually need a communication protocol with a few bits for real-time applications. Therefore a BER or PER encoding fits it.

The ASN.1 schema definition in the examples shows how to encapsulate information for transportation.

4.2.2.2 Basic structure

This section shows what a basic definition looks like and how to interpret it. We will begin with a simple example and explain how to understand it.

The underlying protocol we communicate between the two parties doesn't matter; they only have to know the schema definition. An example is a simple bank transaction between two customers. A customer can make multiple transactions with one person. We send all transactions in a list for efficiency and to reduce overhead.

Listing 4.8: ASN.1 Complex Example

```
SimpleBankTransaction DEFINITIONS AUTOMATIC TAGS ::= BEGIN

```

```
3  BankTransaction ::= SEQUENCE {
4      dateOfTransaction   DATE,
5      from                CustomerInfo,
6      to                  CustomerInfo,
7      transactions        ListOfTransactions
8  }
9
10 CustomerInfo ::= SEQUENCE {
11     firstName           VisibleString (SIZE (3..50)),
12     lastName            VisibleString (SIZE (3..50)),
13     address             Address,
14     contactPhone        NumericString (SIZE (7..12)),
15     iban                VisibleString (SIZE (3..50))
16 }
17
18 Address::= SEQUENCE {
19     street    VisibleString (SIZE (5 .. 50)) OPTIONAL,
20     city      VisibleString (SIZE (2..30)),
21     state     VisibleString (SIZE(2) ^ FROM ("A".."Z")),
22     zipCode   NumericString (SIZE(4))
23 }
24
25 ListOfTransactions ::= SEQUENCE (SIZE (1..100))
26                                 OF Transaction
27
28 Transaction ::= SEQUENCE {
29     transactionId    INTEGER (1..99999),
30     amount           REAL (0.00 .. 9999.00),
31     currency         VisibleString ("CHF" | "EUR" | "USD"),
32     exchangeRate     REAL
33 }
34 END
```

At first, it seems a bit confusing, but let's break it down.

From a higher view, we have modules that contain all definitions and begins with BEGIN and ends with END.

4.2.2.3 Types

The next thing we see are the definitions `BankTransaction`, `CustomerInfo`, `Address`, `ListOfTransactions` and `Transaction`. These are used inside other definitions and used as types.
Types are the most basic representation of information. If you have experience with java, it's comparable to primitive types such as bool, short or char.

The following table gives you an overview of some types.

INTEGER

This is a type that the value is either true or false. It's comparable with a switch that has the state on or off."

Listing 4.9: ASN.1 Exmaple of INTEGER

```
1 doorOpen BOOLEAN ::= TRUE
```

BOOLEAN

An integer is a decimal negative or positive number with variable length. There are usually limitations caused by hardware or software, but ASN.1 theoretically defined no limit.

Listing 4.10: ASN.1 Exmaple of BOOLEAN

```
1 speed INTEGER (0..60) ::= 40
```

BIT STRING

They use it for naming the bits and not bytes and means each bit can have a meaning like the boolean type. The representation is either binary (`1101000100011010'B`), in hex (`82DA'H`) or named (`admin(2)`, the number 2 is the position in the bit array).

Listing 4.11: ASN.1 Exmaple of BIT STRING

```
1 "Sensors ::= BIT STRING {
2     doorOpen(0),
3     windowOpen(1),
4     engineOn(2)
5 }
6 myStatus Sensors ::= {windowOpen, engineOn}"
```

OCTET STRING

An octet (=byte) string is a sequence of bytes (= 8 bits). It has two meanings, either as a state or as a number. It's very similar to a bit string. However, it only has a length of 8 bits and not as in a bit string a length of one bit.

Listing 4.12: ASN.1 Example of OCTET String

```
1 ipV4Address ::= OCTET STRING SIZE(4)
```

DATE It is used when you need to represent a date. Values have the form `YYYY-MM-DD`. It's transmitted as UTF-8 in BER, CER and DER encoding.

Listing 4.13: ASN.1 Example of DATE

```
1 harvardEstablished DATE ::= "1636-09-18"
```

TIME-OF-DAY

Represents a time in `HH:MM:SS`. It is transmitted as UTF-8 in BER, CER and DER encoding.

Listing 4.14: ASN.1 Example of TIME-OF-DAY

```
1 callTime TIME-OF-DAY ::= "18:30:23"
```

DATE-TIME

Represents a combination of date and time with a delimiter T in the form `YYYY-MM-DDTHH:MM:SS`. It's transmitted as UTF-8 in BER, CER and DER encoding.

Listing 4.15: ASN.1 Example of DATE-TIME

```
1 callTime DATE-TIME ::= "2000-11-22T18:30:23"
```

REAL

This is a number with a comma or point between two numbers (ex. 3.14). As you should know, a computer doesn't know the concept of complex numbers, but the numbers have the possibility to be represented as $mantissa*base^{exponent}$ ($314*10^{-2} = 3.14$). This "conversion" of a complex number is also used in the REAL type. Have a look in the explanation from Yury Strozhevsky.

Listing 4.16: ASN.1 Example of REAL

```
1 Total ::= REAL
```

ENUMERATED

It is used to identify items by name rather than by a number from a given choice list. Note that the names of values always begin with a lowercase letter. It's similar to `INTEGER`. The only difference is that an `INTEGER` can have a state of infinity (Discussion.

Listing 4.17: ASN.1 Example of ENUMERATED

```
1 CarColors ::= ENUMERATED {black, red, white}
2 myCar CarColors ::= white
```

OBJECT IDENTIFIER

It is used when you need a globally unique identifier for something. A typical example of this is a digital certificate. An object identifier value is a list of arcs from the root to a node in the object identifier tree. Object identifiers are usually abbreviated with OID and used in comments or code.
The usage of the OID in the context of cryptography is to identify the algorithm and its use. There are many OID for many objects, for example, with RSA The RSA Inc. has a collection for asymmetric cryptography called Public Key Cryptography

Standard or better known as PKCS. These are organised in different OIDs under the 1.2.840.113549.1.

- `1.2.840.113549.1.1` - PKCS-1
- `1.2.840.113549.1.3` - PKCS#3
- `1.2.840.113549.1.5` - RSA PKCS 5
- `1.2.840.113549.1.7` - PKCS-7
- `1.2.840.113549.1.9` - PKCS-9 - Signatures
- `1.2.840.113549.1.10` - PKCS#10 - Certification Request Syntax
- `1.2.840.113549.1.12` - pkcs-12
- `1.2.840.113549.1.15` - PKCS#15 Applicatian Identifier

If we search for the RSA Encryption OID we have to have a look in PKCS#1 OID and the unique identifier `1.2.840.113549.1.1` make clear what the usage is. The description is also given and can be found on the OID infopage.

If you like to explore the tree structure of OID, we recommend the OID tree from oid-info.

Listing 4.18: ASN.1 Example of OBJECT IDENTIFIER

```
1 {joint-iso-itu-t(2) country(16) us(840) organization(1)}
```

SEQUENCE

It is used when you have a collection of items to group together.

Listing 4.19: ASN.1 Exmaple of SEQUENCE

```
1 Contact ::= SEQUENCE {
2     name VisibleString,
3     phone NumericString
4 }
5 driver Contact ::= {name ""J.Smith"", phone ""7325555555""}"
```

SEQUENCE OF

It is used when you have a list or array of a repeated item.

Listing 4.20: ASN.1 Example of SEQUENCE OF

```
1 breakTimes SEQUENCE OF TIME-OF-DAY ::= {
2     "10:00:00",
3     "12:00:00",
4     "14:45:00"
5 }
```

CHOICE

Used when you have a collection of items for which only one of the items can be present at a time. This isn't depending on a single type.

Listing 4.21: ASN.1 Example of CHOICE

```
Location ::= CHOICE {
    streetAddress Address,
    intersection Intersection,
    landmark LandMarkName,
    gpsCoordinates GpsInfo
}
meetAt Location ::= landmark: "Statue of Liberty"
```

IA5String

It is used when you need to use ASCII, including control characters.

Listing 4.22: ASN.1 Example of IA5String

```
TextWithLayout ::= IA5String
```

VisibleString

It is used when you need to use the subset of ASCII that does not include control characters.

Listing 4.23: ASN.1 Example of VisibleString

```
LineOfText ::= VisibleString
```

NumericString

It is used when you need to use only digits and spaces.

Listing 4.24: ASN.1 Example of NumericString

```
    LineOfNumbers ::= NumericString
```

UTF8String Used when you need to handle Unicode characters.

Listing 4.25: ASN.1 Exmaple of UTF8String

```
TextInAnyLanguage ::= UTF8String
```

NULL It is used when you need a placeholder for which there is no value. Further, it is often used as an alternative to the `CHOICE` type or as an optional component of a `SEQUENCE` type.

4.2.2.4 Contraints

Know we know some types usually used for defining a schema. Know there is a possibility to make more constraints to types. This is usually used if you need to restrict the values further.

Permitted Alphabet If you like to restrict using certain characters, it is possible to determine specific numbers and letters. Then this makes it possible to make a restriction. The example only allows `8BI10OD5S`.

Listing 4.26: ASN.1 Exmaple of UTF8String

```
1 HardToReadChars ::= IA5String (FROM("8BI10OD5S"))
```

Pattern Usually, there are well-known patterns for different usages. This usually makes sense, for example, sense for bank account numbers. The example only allows `1234-AB`.

Listing 4.27: ASN.1 Exmaple of IA5String with Pattern

```
1 LicensePlate ::= IA5String (PATTERN "[0-9]#4(-[A-Z]#2)?")
```

Value Size Very often used is the restriction which size of characters, items or other types should be inserted. `SIZE` can be either a fixed number or a range. Fixed-size is a single number like `SIZE(2)`, and the range is read by a minimum and maximum value `SIZE(3..9)`.

Listing 4.28: ASN.1 Exmaple of SEQUENCE with SIZE

```
1 LicensePlate ::= IA5String (SIZE (4..7))
2 CarPark ::= SEQUENCE SIZE (1..25) OF LicensePlate
```

Value Range A number can also be restricted, like the Value Size. The only thing that must be inserted after `INTEGER` is `(number1..number2)`.

Listing 4.29: ASN.1 Exmaple of INTEGER with value range

```
1 CarSpeed ::= INTEGER (0..200)
```

Single Value There is also a possibility to restrict to specific values. It's comparable to an enum.

Listing 4.30: ASN.1 Exmaple of single value

```
1 WarningColors ::= UTF8String ("Red" | "Yellow")
2 InfoColors ::= UTF8String ("Blue" | "White")
3 CitySpeedLimit ::= INTEGER (25 | 30 | 40)
4 HighwaySpeedLimit ::= INTEGER (40 | 50 | 60 | 70)
```

Contained Subtype This is a more advanced restriction. Imagine we have to types with different colours, the first one contains is an `InfoColors` and the second `WarningColors`. Now we want to merge these two colours. This is achieved by using `UNION`.

Another possibility is to combine two quantities, and only the same values are used.

Listing 4.31: ASN.1 Exmaple of Contained Subtype

```
SignColors ::= UTF8String (InfoColors UNION WarningColors)
RuralSpeedLimit ::= INTEGER (
            CitySpeedLimit INTERSECTION HighwaySpeedLimit)
```

Containing/Encoded By

This applies to `OCTET STRING` and is usually used for OID to set the prefix or to set a fixed OID.

Listing 4.32: ASN.1 Exmaple of CONTAINING and ENCODED BY

```
PerInside ::= OCTET STRING (
    CONTAINING Doc
        ENCODED BY { joint-iso-itu-t asn1(1)
                     packed-encoding(3) basic(0) unaligned(1)})

pdf OBJECT IDENTIFIER ::= { iso(1)
    member-body(2)
    us(840)
    adobe(113583)
    acrobat(1)}

Doc ::= OCTET STRING (ENCODED BY pdf)
```

In this section, we are going to have a look at the current published X.509 standard to show you the recent properties. An later, the last chapter shows you the proposed Draft for an extension for hybrid certificates.

4.2.3 Structure

Currently, we are on the third version of the X.509 standard. In this section, we are discussing the actual standard and why there are missing fields for the transition to PQC.

First of all, we need to understand the used fields of the certificate. Please note we are not going through all fields and only covering the relevant parts. For an exact explanation and further explanation, consult the RFC5280.

To make it simplier to understand the structure of the certificate we made a simplified version without primitives types or strcutre from ASN.1. [49]

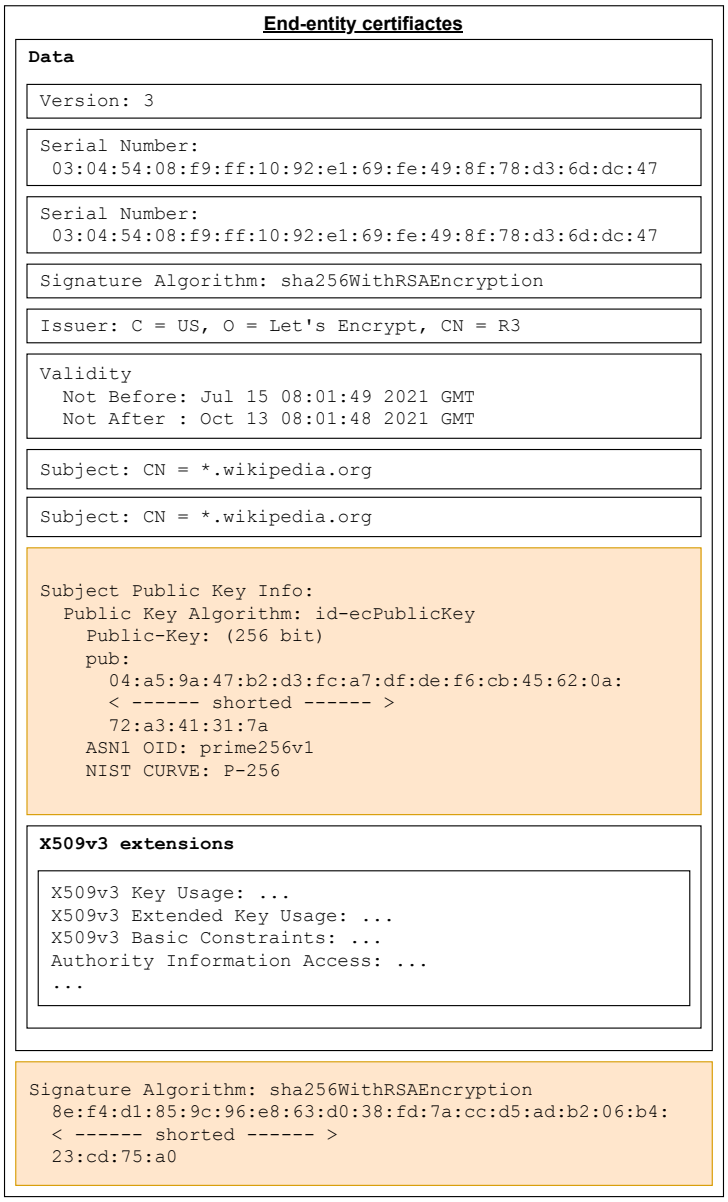

Figure 4.2: X509 End-Entity Certifacte. It's usually used for web servers. Made with Github and draw.io

4.2.4 Fields

This section gives you a brief introduction to the relevant fields to get a basic understanding.

Serial Number

Serials are used to identify a certificate with a unique number. This way, a certificate easy identified by the given serial number. The RFC also says the serial number has to be a positive number, and a system must have the capability to handle values above 20 octets.

Subject Public Key Info

This field is of interest. The actual public key is stored in this section and is relevant for the migration to the algorithm.

Listing 4.33: X.509 Subject Public Key Info as ASN.1 Notation

```
AlgorithmIdentifier  ::=  SEQUENCE  {
    algorithm               OBJECT IDENTIFIER,
    parameters              ANY DEFINED BY algorithm OPTIONAL
}
```

The OIDs for the algorithms are specified in the RFC3279, RFC4055, and RFC4491.

To fully understand the concept behind this field, we are examining the figure 4.2. This should also help you how the design of an approach could be.

In the figure 4.2 the algorithm is ECDSA and can be identified by `id-ecPublicKey`. If we take a closer look in the specification RFC3279 Section 2.3.5 for the OID, it is defined as.

Listing 4.34: X.509 - Subject Public Key Info with ECDSA OID represented as ASN.1

```
ansi-X9-62 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) us(840) 10045 }
id-public-key-type OBJECT IDENTIFIER  ::= { ansi-X9.62 2 }
id-ecPublicKey OBJECT IDENTIFIER ::= { id-publicKeyType 1 }
```

And this is acually the OID 1.2.840.10045.2.1.

The standard also includes the values needed for a mathematical calculation for verification. But first, we are looking closer in the standard, what's defined for ECDSA.

Listing 4.35: X.509 Subject Public Key Info with ECDSA OID represented as ASN.1

```
EcpkParameters ::= CHOICE {
    ecParameters  ECParameters,
    namedCurve    OBJECT IDENTIFIER,
    implicitlyCA  NULL
}

ECParameters ::= SEQUENCE {
    version   ECPVer,    -- version is always 1
    fieldID   FieldID,   -- identifies the finite field over
                         -- which the curve is defined
    curve     Curve,     -- coefficients a and b of the
                         -- elliptic curve
    base      ECPoint,   -- specifies the base point P
                         -- on the elliptic curve
    order     INTEGER,   -- the order n of the base point
    cofactor  INTEGER OPTIONAL -- The integer h = #E(Fq)/n
    }
```

```
18
19 ECPVer ::= INTEGER {ecpVer1(1)}
20
21 Curve ::= SEQUENCE {
22     a          FieldElement,
23     b          FieldElement,
24     seed       BIT STRING OPTIONAL }
25
26 FieldElement ::= OCTET STRING
27
28 ECPoint ::= OCTET STRING
```

The curve parameter are depending `CHOICE` and are described as follow.

Choice	Description
ecParameters	The parameters are user defined and can derivate from recommendations.
namedCurve	This are parameters indentified by a OID.
implicitlyCA	The parameters for the curved are inherted from the parent certificate.

Table 4.1: X.509 - Subject Public Key Info - Fields

Extensions

Extensions are only available in X.509 v3 and has extensions defined in the RFC5280 section 4.2.1. These are organised with OID in a `SEQUENCE`. Additionally, each extension is designed to be critical or non-critical. If a system using the X509v3 certificates encounters an extension, it must proc contain information. Critical extensions are mandatory to proceed. Otherwise, the certificate is rejected. A non-critical extension also has to be procced, but only if recognised by the system.

The standard also provides a list of defined extensions and have a prefix OID of 2.5.29. Here are some common extensions and full list is defined in the RFC5280 section 4.2.1 The table 4.2 shows some common extensions used in X.509 certificates.

Certificate Signature

Each certificate is signed by the "parent" certificate the CA. We now give you a short introduction to singing to understand the ingredients needed.

The first thing we need is a possibility to make a unique key of the certificate depending on the content. Therefore a Hashing-algorithm is a right choice.

Cryptographic hash function

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length. Values returned by a hash function are called message digest or simply hash values.

Extension	Description
Authority Key Identifier	Each certificate has a Subject Key Identifier and a certificate has to be signed by a authority private key. This field is used to identify this private key/certificate.
Subject Key Identifier	This Identifier is used as a unique key for each certificate. The value is genereates from the public key, the first method is a 160 bit SHA-1 of `BIT STRING` subjectPublicKey and the second method is composed of a four-bit type field with the value 0100 followed by the least significant 60 bits of the SHA1 hash of the value of the `BIT STRING` subjectPublicKey
Subject Alternative Name	This is usually used to bind the certificate to alternative resources such as Email, DNS, IP or URI.
Key Usage	This extension defines the purpose of the certificate, this could be certificate singining, digital singing, non repudation, data encipherment, key encipherment, key agreement, CRL signing, encipher only or decipher only. A detailed explanation can in RFC5280 section 4.2.1.3

Table 4.2: X.509 - Common Extensions

The follwing figure 4.3 show you how this mathematical function basically works. The text "It's a sunny day" has a unique hash and could be used as unqiue identifier.

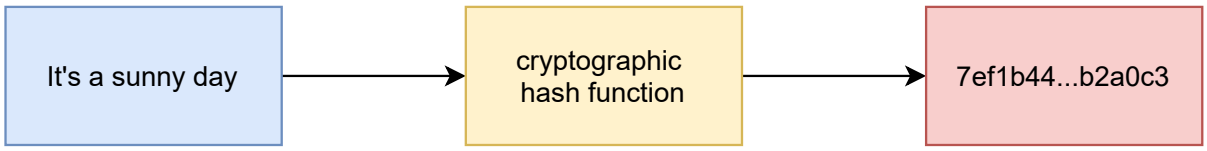

Figure 4.3: Simplification of a cryptographic hash algorithm

Currently, there are these cryptographic hash algorithms / One-Way Hash Functions referred by RFC5280. The algorithms are listed in the RFC3279 section 2.1, RFC4055 section 2.1 and RFC4491 section 2.1.

- MD2
- MD5
- SHA1
- SHA224
- SHA256
- SHA384
- SHA512
- GOST R 34.11-94

The next step is to make sure the CA has signed the digital signature. The CAs private key is used to encrypt the hash of the digital certificate. The client only hashes to decrypt the signature and compare the generated hash with the decrypted hash to validate the subject certificate.

The figure 4.4 shows how the text "It's a sunny day" is signed.

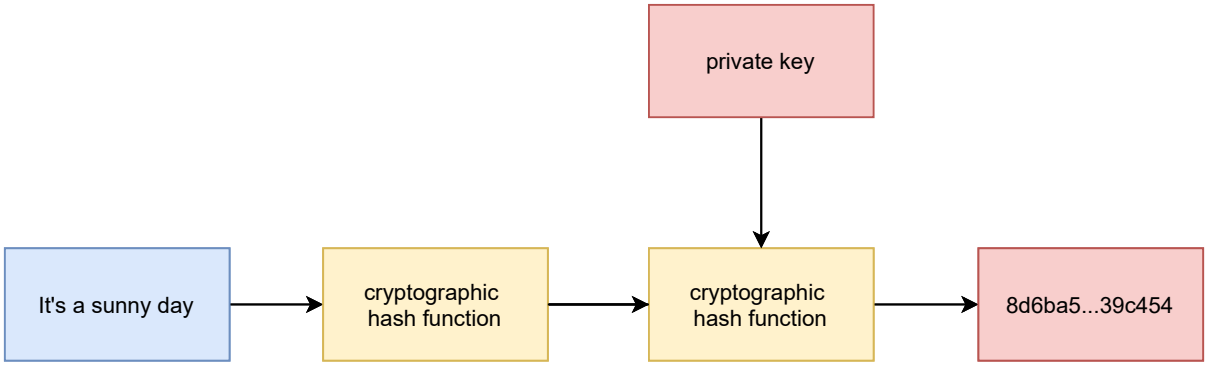

Figure 4.4: Simplification of a singing

The following singing algorithms are referad by RFC5280. The algorithms are listed in the RFC3279 section 2.1, RFC4055 section 2.1 and RFC4491 section 2.1.

- RSA
- DSA
- ECD Elliptic Curve Digital
- RSASSA-PSS
- GOST R 34.10-94
- GOST R 34.10-2001

Because of the combination of singing algorithms and hash algorithms, there is the field signatureAlgorithm. 4.36 shows that the algorithm is defined as a OID with additionally parameters.

Listing 4.36: ASN.1 - X.509 - signatureAlgorithm

```
1 AlgorithmIdentifier  ::=  SEQUENCE  {
2     algorithm  OBJECT IDENTIFIER,
3     parameters ANY DEFINED BY algorithm OPTIONAL
4 }
```

A example is OID is 1.2.840.113549.1.1.5 with SHA1 and RSA

The value of the signature calculation is stored in the Signature field. Figure 4.2 has the signature on the bottom.

4.3 Approaches

At the moment of writing, there is no standard for the transition of digital certificates. Therefore, we provide you with some guidance in the direction the transition could go and the approaches for transition to new quantum-safe certificates.

First, we give you an overview of the design requirements for the new certificates, and afterwards, we show you some approaches shown from different sources.

4.3.1 Design considerations

For developing the new certificates, we first have to think about the design requirements. Let's begin with the devices that use the certificates.

There are systems out there that are no longer supported by the manufacturer or potentially new devices without post-quantum algorithms. Therefore **Backwards-Compatibility** is one of the most important requirements when designing the new certificate.

There are three possible scenarios if we take a client-server architecture as an example. We are using the term "hybrid-aware" for denoting a client or server that can use a PQC algorithm. [52]

1. Hybrid-aware client, hybrid-aware server: These parties should negotiate and use hybrid modes.
2. Hybrid-aware client non-hybrid-aware server: These parties should negotiate a traditional (non-PQ) cypher suite (if the hybrid-aware client is willing to downgrade to traditional-only).
3. Non-hybrid-aware client, hybrid-aware server: These parties should establish a traditional (non-PQ) cypher suite (if the hybrid-aware server is willing to downgrade to traditional-only).

We also have the problem of a transition time, which means the point when a solution/-standard is published and approved until it's implemented and functional. That means there is a need for pre and post-quantum cryptographic algorithms. This time makes it's difficult for a traditional certificate (RFC5280) to be used because its design only supports one public/private key and one signature. It further means we need **two or more algorithms** for the transition time.

If we are talking about a hybrid certificate in general, there are some questions we are faced with. NIST has thankfully already made some considerations and came up with these questions. [52]

1. How to negotiate the use of hybridization in general and component algorithms and parameters specifically?
2. How many component algorithms can be combined?
3. How should cryptographic data from multiple algorithms (public keys/ciphertexts/signatures) be conveyed?
4. How should cryptographic data from multiple algorithms (e.g., shared secrets) be combined?

Negotiation
There different protocols that agree to a certain algorithm to use. If we take a TLS as a example the negotiation takes place by sending a list of supported algorithms. Also see figure 4.5.

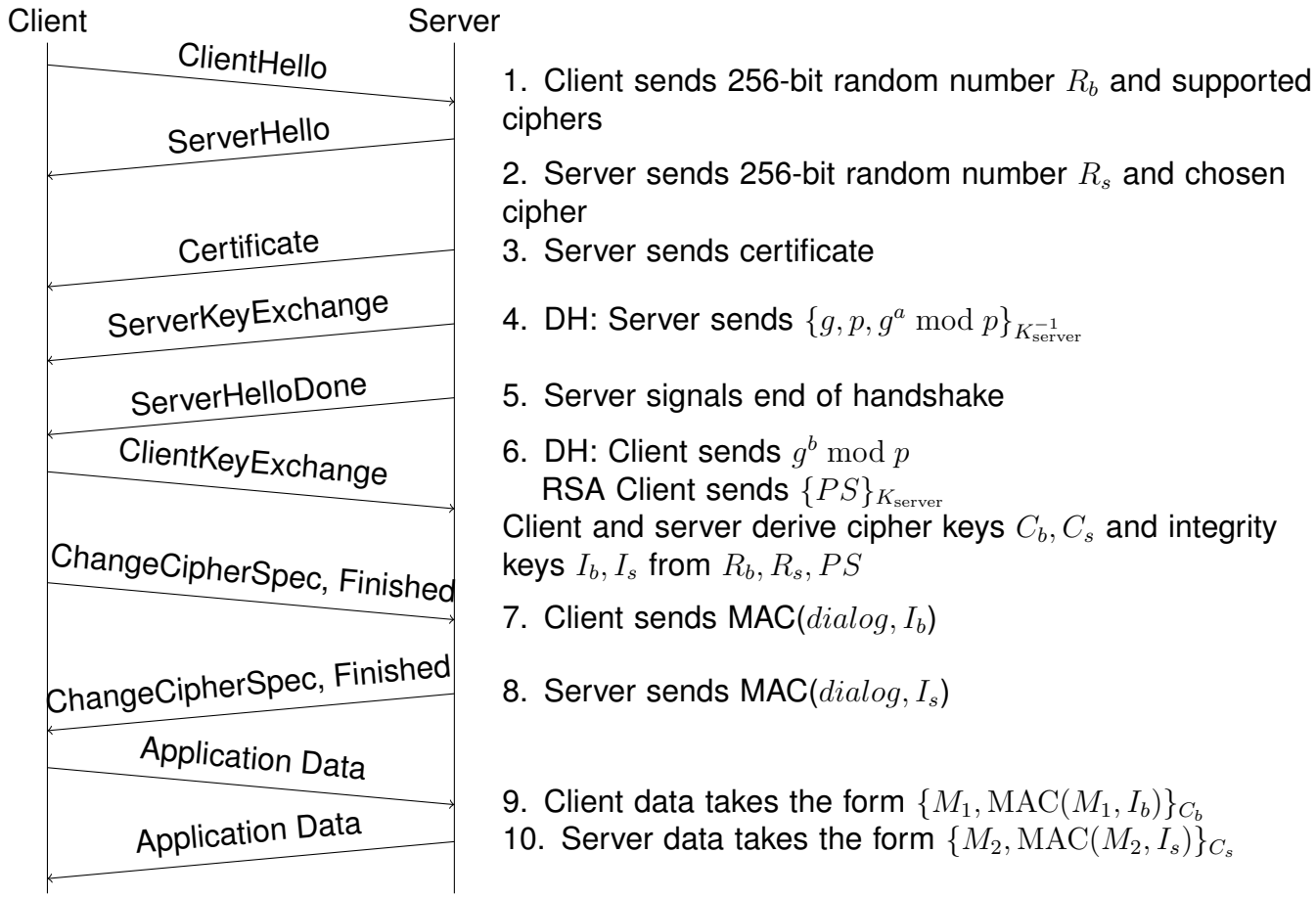

Figure 4.5: TLS 1.2 Handshake

Number of component algorithms
Currently, the X.509 standard only supports one algorithm and one signature; also see 4.2. For the transition, it is needed to have two or more algorithms supported.

Convey cryptographic data
For a system with multiple cryptographic techniques, the data must be sent on the protocol layer or inside a certificate. The first has to make changes in the existing protocols. For example, TLS supports extensions in the `ClientHello` and `ServerHello`, but other messages do not. Another option is to concatenate the multiple algorithms into the existing message structure. The second option requires no change in the protocol format or logic. Therefore it's simpler and has better backwards compatibility.

Combine cryptographic data
Besides the support for two or more algorithms, there also have to be a solution for encryption data using these algorithms. How should a secure connection be established?

- Choose one algorithm from the certificate and ignore the others.
- Combine the keys securely.

4.3.2 Hybrid Certificates

The approach is from A. Truskovsky and has the idea of embedding an alternative algorithm and signature to an existing certificate. [53]

We now go into a deeper level to understand the changes made based on the RFC5280.

4.3.2.1 General idea

As you can see from the 4.6 the certificate has two additional fields. The idea is to add three fields in the extensions. That way, a client that is a hybrid-aware client and server can establish a post-quantum safe connection. The elements are all already present in the RFC5280 and are reused with this approach.

In the next part, we will look closely at the new extensions of this approach.

Subject Alt Public Key Info Extension

This extension is equal to the Subject Public Key Info from the RFC5280, and the only change is the position in the structure.

Listing 4.37: Hybrid Certificates Subject Alt Public Key Info Extension

```
1 SubjectAltPublicKeyInfoExt  ::=  SEQUENCE  {
2     algorithm                    AlgorithmIdentifier,
3     subjectAltPublicKey          BIT STRING
4 }
```

Alt Signature Algorithm Extension

Also, the Signature Algorithm is embedded as an additional extension.

Listing 4.38: Hybrid Certificates Alt Signature Algorithm Extension

```
1 AltSignatureAlgorithmExt  ::=  AlgorithmIdentifier
```

Alt Signature Value Extension

Listing 4.39: Hybrid Certificates Alt Signature Value Extension

```
1 AltSignatureValueExt  ::=  BIT STRING
```

If the above extensions a new certificate structure could look like in the figure 4.6.

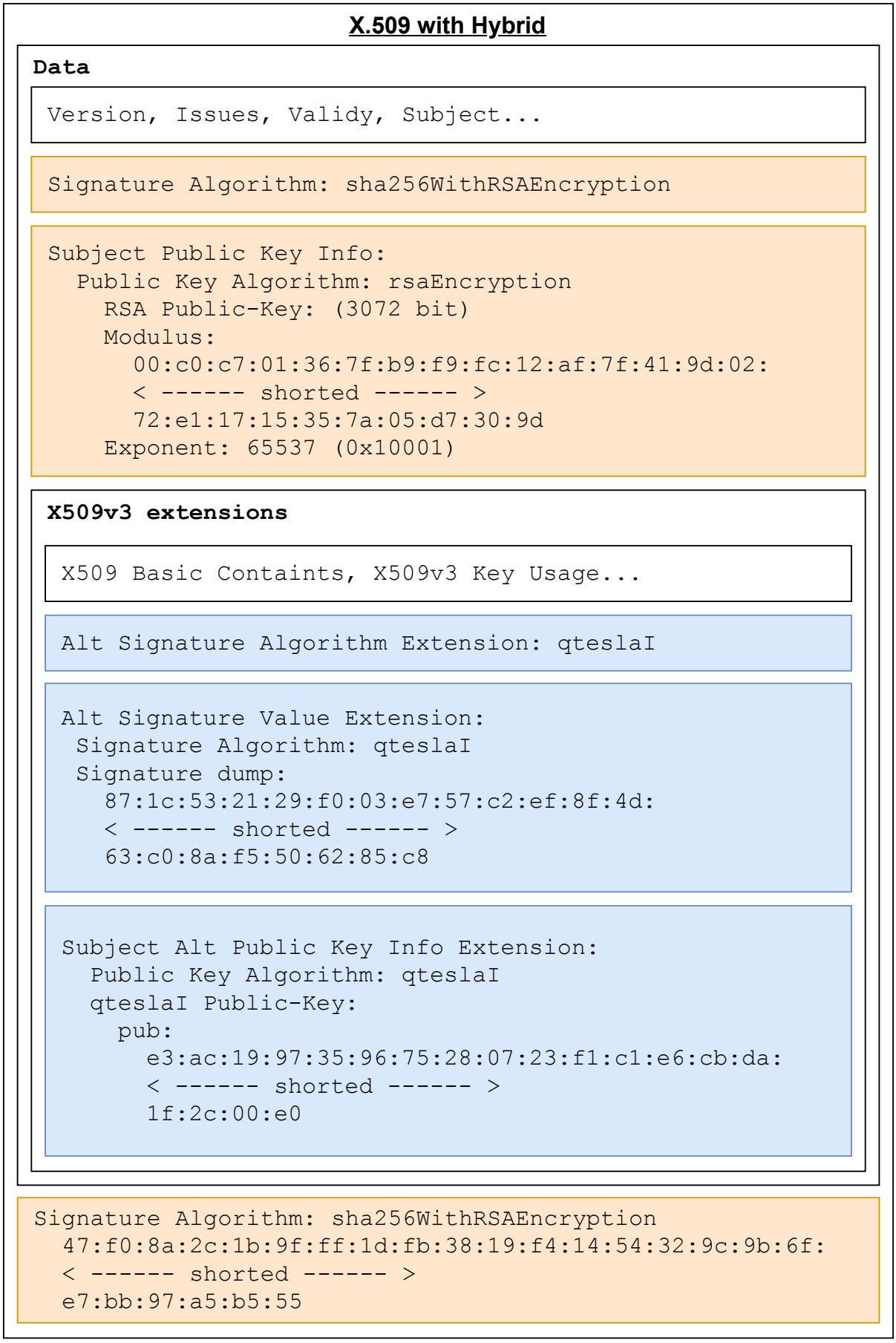

Figure 4.6: X509 Hybrid Certificate. Made with GitHub and diagrams.net

4.3.2.2 Generation

The generation of a certificate in general in two steps. First, a `PreTBSCertificate` is signed with the alternative signature algorithm and then added as an extension. The second step is to convert the `PreTBSCertificate` to a `TBSCertificate` with the alternative signature and signed with the traditional algorithm.

Figure 4.7 show how the certificate is generated.

The steps can be summaries with the following steps from the [53] draft.

1. Create a `PreTBSCertificate` object, which is populated with all the data to be signed by the alternative private key, including the `SubjectAltPublicKeyInfoExt` and `AltSignatureAlgorithmExt` extensions. (figure 4.7 left)
2. Calculate the alternative signature on the DER encoding of the `PreTBSCertificate`, using the Issuer's alternative private key with the algorithm specified in the `AltSignatureAlgorithmExt` extension. (figure 4.7 left)
3. Add the calculated alternative signature to the `PreTBSCertificate` object using the `AltSignatureValueExt` extension. (figure 4.7 middle)
4. Convert the `PreTBSCertificate` to a `TBSCertificate` by adding the signature field and populating it with the algorithm identifier of the conventional algorithm to be used to sign the certificate. (figure 4.7 middle)
5. As per [RFC5280], calculate the conventional signature using the conventional private key associated with the Issuer's certificate and create the certificate from the `tbsCertificate`, `signatureAlgorithm` and `signature`. (figure 4.7 right)

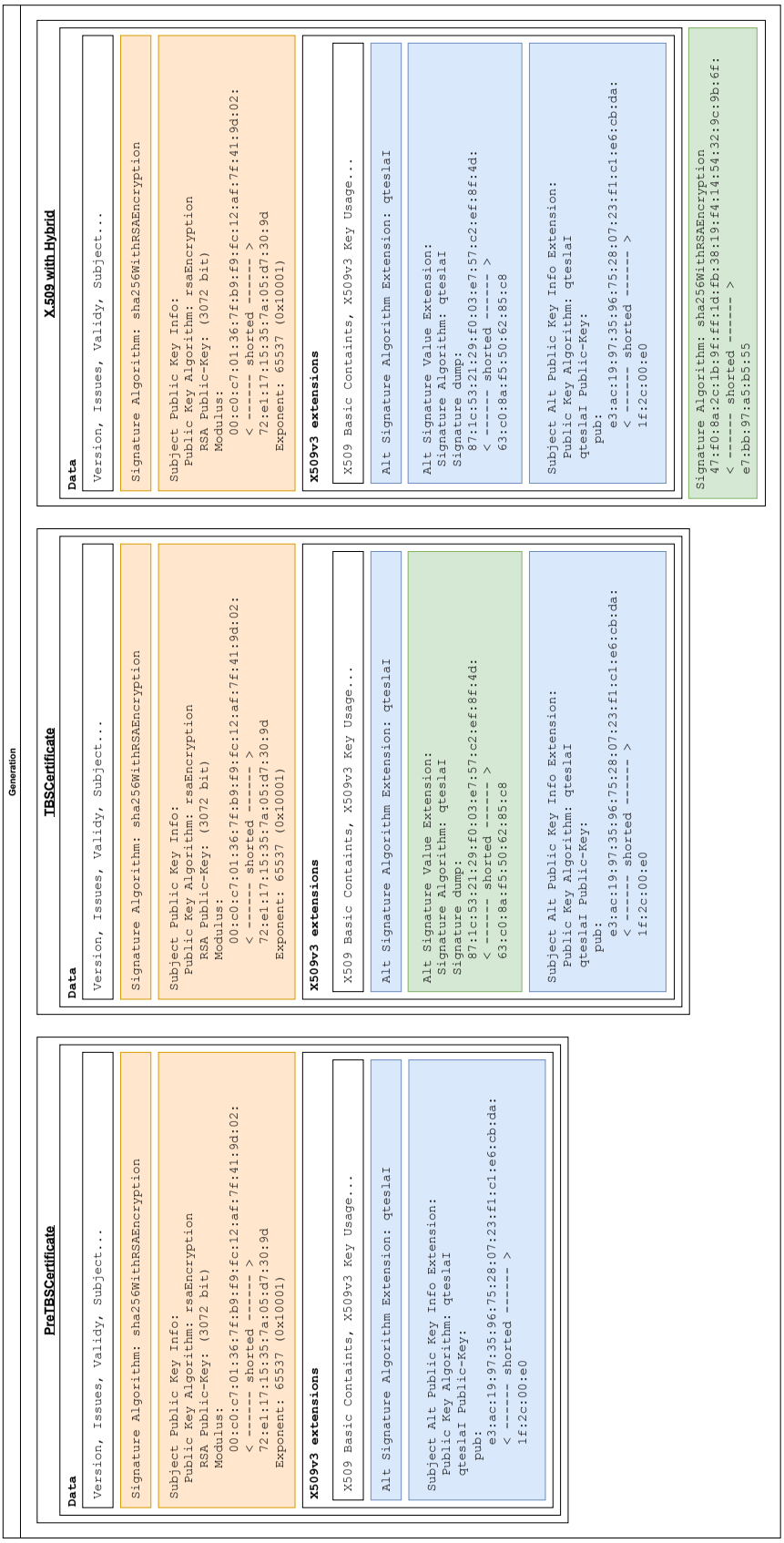

Figure 4.7: X509 Hybrid Certificate Generation. Made with GitHub and diagrams.net

4.3.2.3 Verificaiton

The verification of the certificate is an interesting part because this makes the backwards compatibility. A software verification component should scan the certificate for alternative algorithms. If so, it should verify the alternative also.

This extra step let's decide the software it the extra verification takes place.

The verification is presented in the figure 4.8.

The steps to verify a signature are as follow as present in the [53].

1. ASN.1 DER decode the `tbsCertificate` field of the certificate to get a `TBSCertificate` object. (figure 4.8 left)
2. Remove the `AltSignatureValueExt` extension from the `TBSCertificate` object and set aside the alternative signature. (figure 4.8 left)
3. Remove the signature field from the `TBSCertificate` object, converting it to a `PreTBSCertificate` object. (figure 4.8 middle)
4. ASN.1 DER encode the `PreTBSCertificate` object. (figure 4.8 middle)
5. Using the algorithm specified in the `AltSignatureAlgorithmExt` extension of the `PreTBSCertificate`, the alternative public key from the Issuer's `SubjectAltPublicKeyInfoExt` extension and the ASN.1 DER encoded `PreTBSCertificate`, verify the alternative signature from (2). (figure 4.8 right)

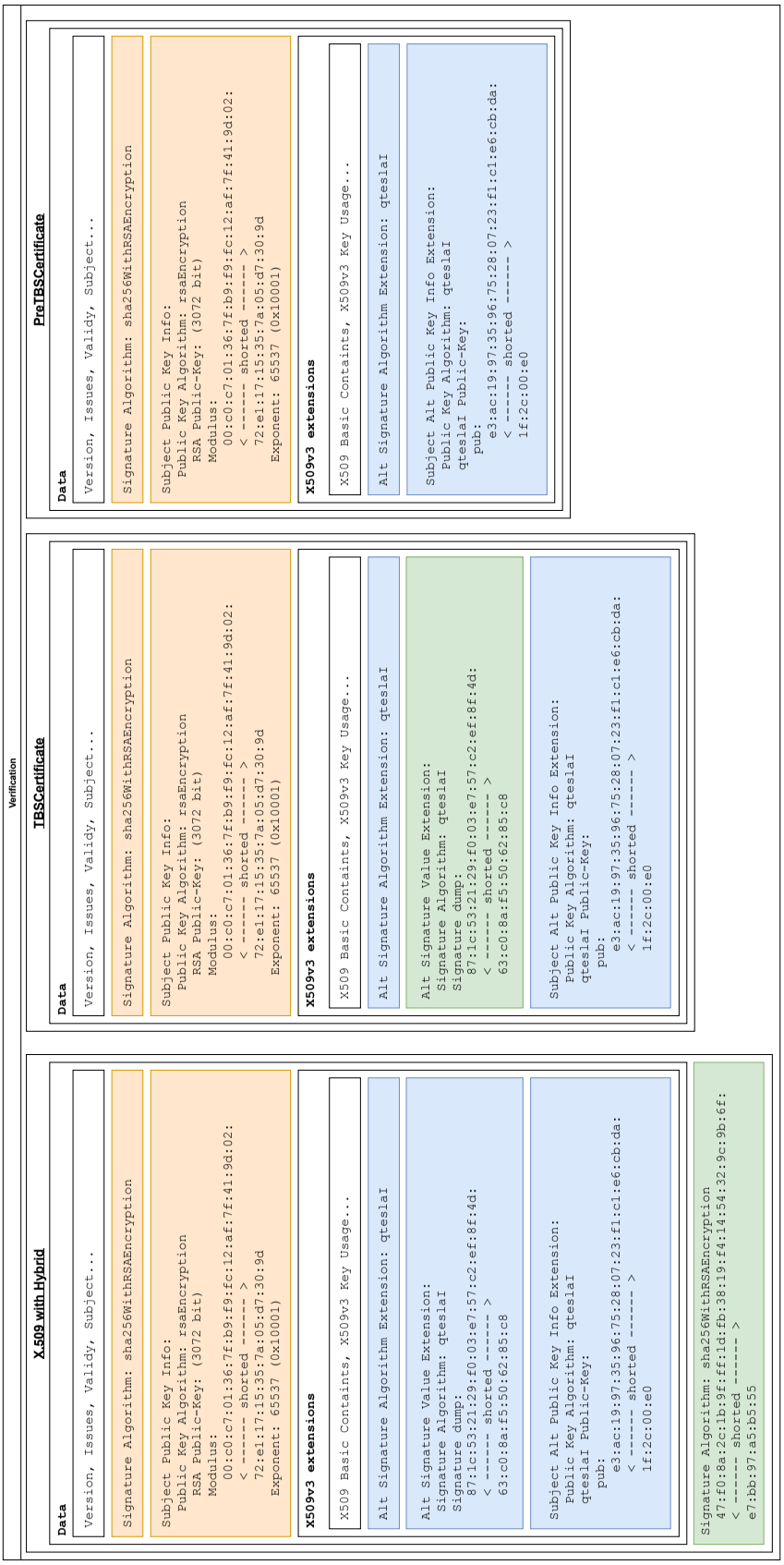

Figure 4.8: X509 Hybrid Certificate Verification. Made with GitHub and diagrams.net

4.3.2.4 Properties

In the following section, we are discussing how the crypto agility properties can be applied to this approach.

- **Extensibility** A certificate has a design limit of two certificates but can use more than only one algorithm.
- **Removeability** Existing algorithms can only be made invalid in a certificate. This is achieved by the expiration date or revoked with by a CLR.
- **Fungibility** This approach doesn't change the properties of the X.509 certificate. After the NIST candidate selection has been finished, a new algorithm ID or OID are created.
- **Interoperability** This approach doesn't change the properties of the X.509 certificate.
- **Updateability** A client can update to new algorithms if supported, but there is also the possibility to use the old algorithms.
- **Flexibility** The new algorithm is by design not relying on the traditional algorithms.
- **Compatibility** The certificate only can provide information about the algorithm and the parameters. The compatibility of the certificate depends on the software component that has eighter the ability to use a traditional certificate or hybrid certificate.
- **Reversibility** The structure of the certificate certainly gives this. A client that fails to use the alternative certificate can use the old algorithm.
- **Transition Mecahnisms** The software has to provide a new component for the extraction of the alternative cryptographic material. The hybrid certificate only provides de necessary parameters for the implemented algorithms on the client.
- **Backards Compability** Legacy systems or not yet upgraded systems can use the old way of verification. The change in the X.509 structure should have little effect on the verification components. Thus the client has to ignore the extensions and proceed with the known field.

4.3.3 Composite Certificates

The following possible solution is based on the draft from M. Ounsworth and subdivided in the definition for the public, and private key [54], and for the signature, [55]

4.3.3.1 Genereal idea

The idea of this draft is to have more than two algorithms in a certificate. For public key, private key and signature, this also means we need to identify the pairs.

Unfortunately, the draft doesn't include how to implement this in the existing certificate structure, but we could assume it is included as the hybrid certificate approach as an extension.

The [54] has the definition for the public and private keys. So by that, we will only look further in the public key.

For the Public key, Private key and signature, we need a structure for multiple fields. Therefore, we will use the `SEQUENCE OF` with a minimum of two algorithms to use.

The next question is verifying the signature with those algorithms included. The draft states that we could check only specific (Composite-OR Public Key) or all (Composite Public Key) algorithms.

The drafts show the following structure for composite certificates in figure 4.9.

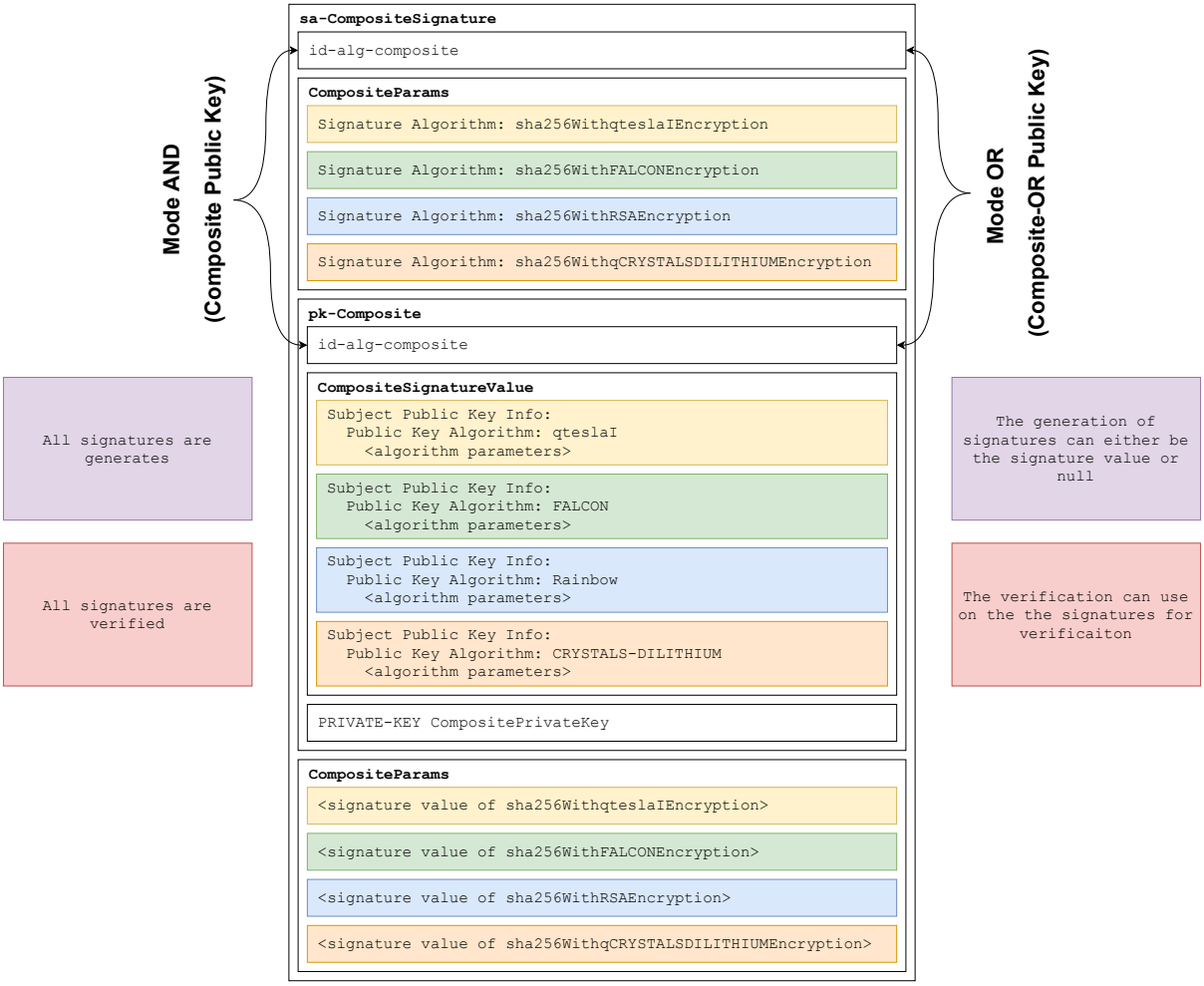

Figure 4.9: Composite Certificate. Made with GitHub and diagrams.net

Composite Keys

We first need a structure for the public and private keys. As mentioned above, this approach has two modes, the AND and OR. Here we give you more details about these Modes.

AND mode requires to use of all certificates, and devices must support all algorithms in the certificate. This could be very interesting for use cases with high confidentially or integrity requirements.

OR mode only uses a subset of algorithms in the certificate. Also, the processer of generating signatures doesn't require using all keys. This is more interesting for use cases in which devices only support fewer algorithms than present in the certificate. "The design intent of this mode is to support migration scenarios where an end entity has been issued keys on algorithms that either itself or the peer with which it is communicating do not (yet) support. This design allows for both the mode where the site signatures that it knows its peer cannot process in order to save bandwidth and performance and the mode where it includes all component signatures and allows the verifier to choose how many to verify." [55]

The defined structure in the draft has the id-alg-composite that defines the mode of the certificate and how generation and verification are done. CompositePublicKey and CompositePrivateKey are `SEQUENCE`s of a minimum of two algorithms used, As you see below, both keys are using the already defined structure (`SubjectPublicKeyInfo` and `OneAsymmetricKey`) from the RFC5280. The benefit of using the existing structure is, only the location of keys are changed, and old/legacy/alternative algorithms can be used.

It's also essential to have the proper order in `CompositePublicKey` and `CompositePrivateKey`. This means, for example, the first private key and public key have to be the same algorithm, The second private and public keys are from the same algorithms and so forth.

Figure 4.10 illustrates the process of generation and verification.

Listing 4.40: Composite Certificate Keys

```
1     pk-Composite PUBLIC-KEY ::= {
2         IDENTIFIER id-alg-composite
3         KEY CompositePublicKey
4         PARAMS ARE absent
5         PRIVATE-KEY CompositePrivateKey
6     }
7
8     CompositePublicKey ::= SEQUENCE SIZE (2..MAX)
9                                 OF SubjectPublicKeyInfo
10
11    CompositePrivateKey ::= SEQUENCE SIZE (2..MAX)
12                                OF OneAsymmetricKey
```

Composite Signature

For the generation of the signature is the public key needed of the parent certificate. The first is the mode; therefore, we need to define the `id-alg-composite`. The next

thing is the signatures values CompositeSignatureValue and signatures parameters CompositeParams. The order of the `SEQUENCE` depends on the algorithms defined in the public keys `CompositePublicKey`. Therefore we need to have the same order. Otherwise, the process of generation and verification fails. We also include `pk-Composite` with the key material.

Listing 4.41: Composite Certificate Signature

```
1     sa-CompositeSignature SIGNATURE-ALGORITHM ::= {
2         IDENTIFIER id-alg-composite
3         VALUE CompositeSignatureValue
4         PARAMS TYPE CompositeParams ARE required
5         PUBLIC-KEYS { pk-Composite }
6         SMIME-CAPS { IDENTIFIED BY id-alg-composite } }
7     }
8
9     CompositeParams ::= SEQUENCE SIZE (2..MAX)
10                                OF AlgorithmIdentifier
11
12    CompositeSignatureValue ::= SEQUENCE SIZE (2..MAX)
13                                OF BIT STRING
```

4.3.3.2 Generation

The generation of the keys depends on the mode is operating. As mentioned above, the algorithms can operate either in OR or AND mode. The only main difference between the two- modes in the generation of the signature is that OR can skip signatures, and AND has to generate all.

If we take a look at the figure 4.10 above, we can see in the process of generation of signatures.

As we know from a traditional generation, we apply an algorithm to generate signatures from the private key and message. We don't do it for one algorithm but several different ones. The pictures also show two methods of generation of the signature. The first is the AND mode, and the second is the OR mode. Both modes are working in the same way, and the difference is the OR mode outputs either null or a signature value.

The algorithms are processed as a queue, and orders in the `SEQUENCE` present.

For a more technical explanation, the draft provides an enhanced introduction and explanation of implementation in Section 3.1 and Section 3.2.

4.3.3.3 Verification

The verification of the signatures are present in the under part of the Figure 4.10 and has two modes. Depending on our mode, the first process AND is processed, otherwise OR.

The ingredients needed for verification are the public key gained from the signature of the parent certificates, signatures from the certificate to check and the algorithms used for verification. These are in both modes processed as queue and in the same order processed as present in the `SEQUENCE`.

The verification in the AND mode has to verify all actual signatures. Otherwise, it will be in an invalid state. In comparing the OR mode, only one or a few signatures have to pass the verification.

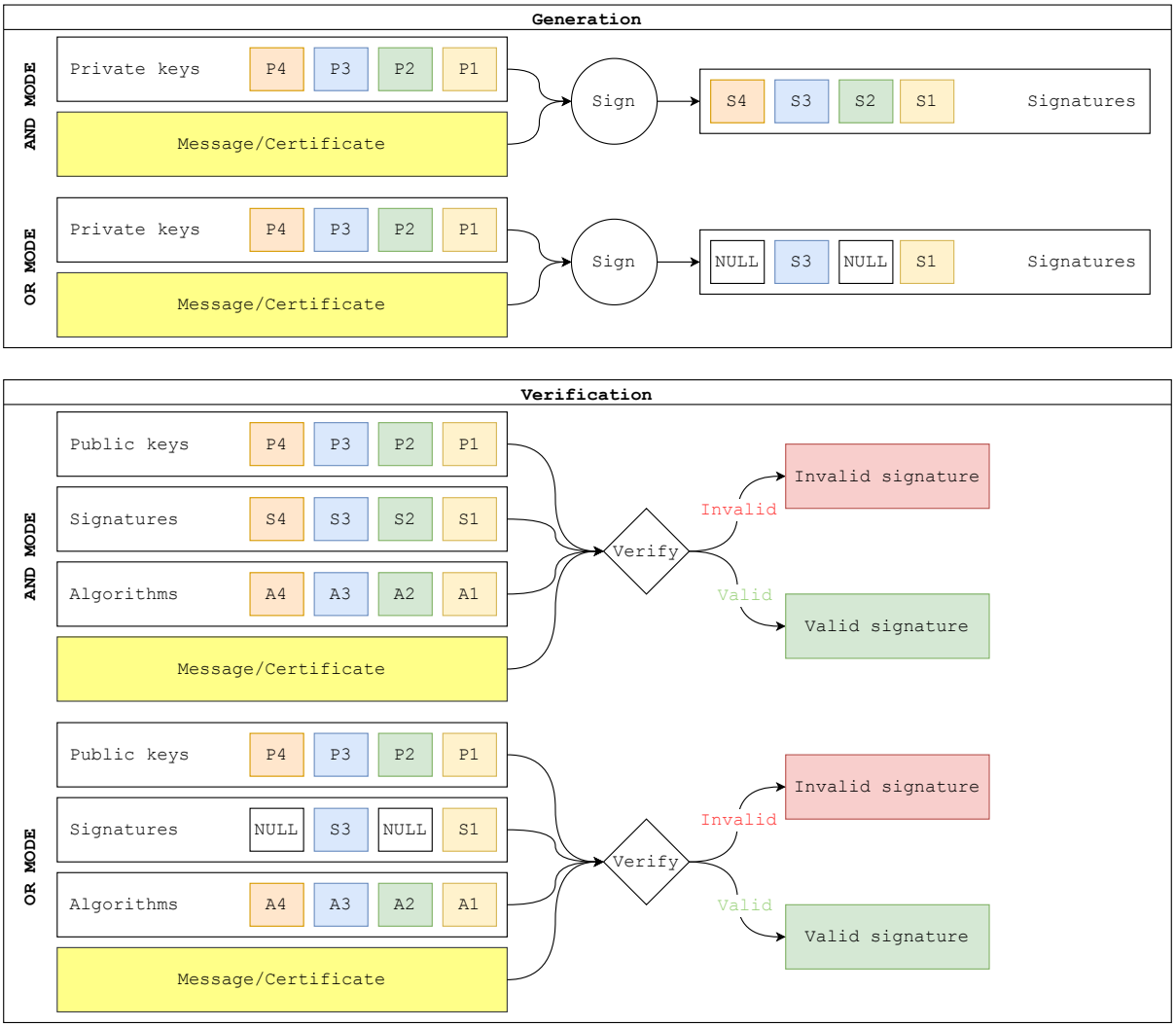

Figure 4.10: Composite Certificate. Made with GitHub and diagrams.net

4.3.3.4 Properties

In the following section, we are discussing how the properties from 5.2 can be applied to this approach.

- **Extensibility** The ability in the OR mode makes it possible to add more algorithms or to remove some. The AND mode has to process all algorithms. This could make it more complex.

- **Removeability** Some algorithms can be removed from the sequences or can be nulled in the OR mode.
- **Fungibility** The replacement of the algorithm could be performed by adding the new algorithm and nullifying the signature.
- **Interoperability** A software component can choose an algorithm in the OR mode for verification.
- **Updateability** The replacement of an algorithm can be achieved by using an adding the new algorithm and removing the old.
- **Compatibility** By running in the OR mode, the software can choose which algorithm have to be verified; this also depends on the policy that is the certificate.
- **Reversibility** Unfortunately, the draft don't provide any information on how it's implemented in the current X.509 certificate structure.
- **Transition Mecahnisms** The software has to provide a new component for the extraction of the alternative cryptographic material. An update of the software is certainly necessary.
- **Backards Compability** Currently, there is too little information available to make a clear statement of how it is implemented in the current X.509 certificate structure.

4.3.4 Parallel Hierachie

Another approach is to use two PKI systems with the currently standard RFC5280. The idea is that one PKI uses a traditional algorithm and another uses only post-quantum safe algorithms. This means that each end-entity has two certificates, one for each hierarchy.

The idea is illustrated in Figure 4.11. The traditional certificate can be run until it is broken. The Post-quantum certificate to this point is optionally verified.

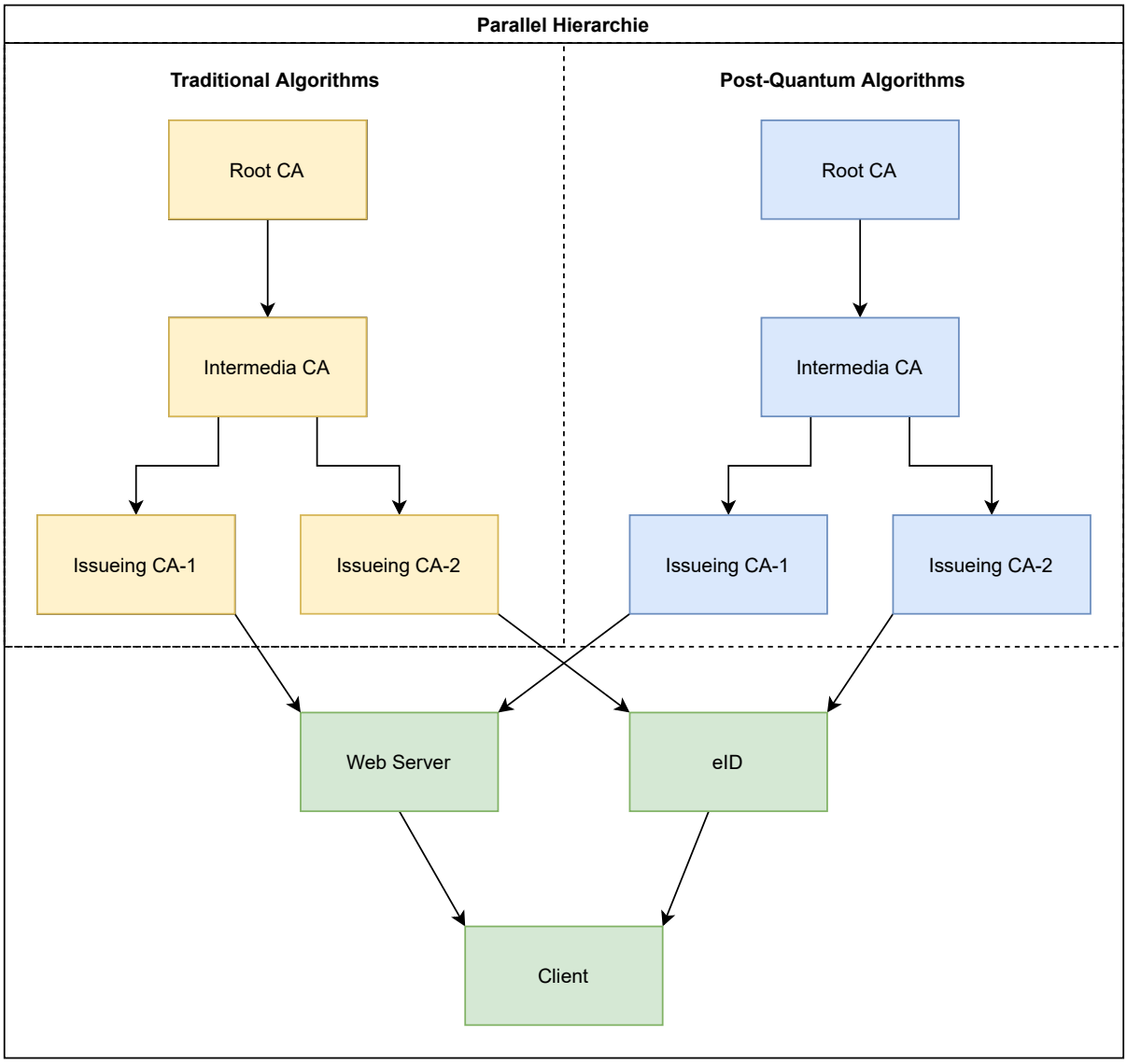

Figure 4.11: Parallel PKI Hierarchie. Made with GitHub and diagrams.net

4.4 Conlusion

The structure of the certificates is a crucial factor for migration to new algorithms and transition time. Some systems adapt fast to new changes because of companies' updated strategies, and there are, of course, legacy systems. And one of the widely used structures is X.509 certificates, which is used almost in every modern system or in protocols, such as HTTPS, S/MIME or eIDs. This means that after the quantum computer time comes, the algorithms behind are broken, and an alternative way has to be already to make a fast switch to the post-quantum safe word. This is where the design of the X.509 is limited and has to be changed with different presented approaches.

The first section of this chapter gives you an introduction to X.509 and makes a deep dive into ASN.1 to fully understand it. This extensive tutorial into ASN.1 gives you more information than needed for the following chapters cause of the lack of experience with it and to understand future standards. There is an expectation of the certificates from different sources with multiple algorithms and signatures, but there is also the possibility of parallel PKI hierarchies. The first approach presented is using exactly two algorithms in a certificate. This is easily implemented by using the extension field from the X.509 version 3. Therefore this has highly backwards compatibility for legacy systems. The second approach is a composite certificate containing two or more algorithms. Fortunately, the draft doesn't provide any information on how to implement in the existing structure of a X.509, but we can expect that either the design of the official standard changes or the extension fields are used. The least is a combination with the first approach, with multiple algorithms being used. Last but not least, the parallel hierarchies have the idea of using two PKI systems and providing end-entities with two independent certificates. Each of these approaches has its advantages and disadvantages. is shown in the table 4.3. The table 4.3 was taken from this source [56] and supplemented with our information.

The presented approaches show ways how to address the problem for multiple certificates. We can clearly say there has to be the adoption in the used libraries and software used in today's software. It's also to be expected that a change is made in the structure of the X.509 version 3, because of the importance of backwards compatibility.

Table 4.3: Comparision of approaches

Approach	Advantages	Disadvantages
Quantum-safe certificates	• Only few changes of standards and applications/devices[56] • Only moderate increase of certificate size[56] • Hardly any changes need to be made to the system, only the algorithms need to be supported.	• Abrupt migration for all applications at the same time[56] • No fall back in case security or implementation issues are discovered for quantum-safe algorithms in the future [56] • All systems have to support the algorithm immeditaly • Should there be another incident in the future, the same thing will have to be migrated again.
Hybrid certificates	• Smooth transition to quantum-safe certificates[56] • Combines security of pre- and post-quantum algorithms[56] • Legacy systems have still the possibility to support traditional algorithms • The changes in the software is can take place in later or erleay phase, if the extensions field are used.	• Needs changes of standards (e.g. RFC 5280) to store and verify two signatures and two public keys in a certificate[56] • Size of certificates increases[56], this is depending on the algorithms beeing used. • The complexity in verification and generation has the risk of more bugs, depending on the developer

Continued on next page

Table 4.3 – continued from previous page

Approach	Advantages	Disadvantages
Composite Certificates	• Combines security of pre- and post-quantum algorithms[56] • The use of multuple algorithms has the abbility to deliver different algorithms and signature, depending on the type of the device (IoT) • The mode AND could make a connection more secure, because of the multiple vertification of the signatures.	• Abrupt migration for all applications at the same time[56] • Needs changes of standards (e.g. RFC 5280) for two signatures and two public keys in a certificate[56] • Size of certificates increases the most[56] • Implemetation is very complex
Parallel hierarchies	• Only few changes of standards and applications/devices[56] • Smooth transition to quantum-safe certificates[56] • Only moderate increase of certificate size[56]	• PKI software needs to be changed to manage parallel hierarchies[56]

5 Crypto Agility

As mentioned above, NIST is currently doing a challenge for new algorithms, for a replacement for RSA and ECC based Algorithms. This transition of new algorithms has always been a challenge, for example, DES to AES.

In 1970 the Nation Bureau of Standards (NBS), now says NIST, needed a standardized way for secure and confidential interagency communication. After discussing if the NSA and two tender DES was finally published in the Federal Register in 1975 and became the official standard in 1976. After almost 20 years in operation, the algorithms were broken in June 1997 in the DESHALL project and July 1998 deep crack. [57]

Therefore a new algorithm was needed, and the U.S. Department of Commerce wrote out a search for a new successor, algorithms and the final algorithm was released 12. In September 1997, the algorithm Rijndael was selected and released as Advanced Encryption Algorithm (AES). [58]

The problem now is that software vendors have to adapt their software to be safe again. Imagine an ERP system that is used in different companies in different versions. The newest version of the software has the AES implemented, but also DES to be able to support older versions of the system. The **Backward Compatibility** makes the software able to communicate with older components. Software is, of course, also faced with other problems. We show this in the following sections.

5.1 Definition

"Crypto-agility, or cryptographic agility, is the capacity for an information security system to adopt an alternative to the original encryption method or cryptographic primitive without significant change to system infrastructure." [59]

This also means a system is capable of dealing with new algorithms and has the possibility to change cryptography in a fast way.

5.2 Properties

Properties such as confidentiality, integrity, availability, etc. are assigned to cryptographic algorithms. The same way [60] proposes the following properties to cryptop agility.

- **Extensibility** It should be possible to extend a protocol or software with a new algorithm. It is hard to implement other algorithms and ensure the protocol or software can operate normally.

  Example The X.509 can use other algorithms (Algorithm-ID)[61]

- **Removeability** The ability to remove cryptography systems that are known as insecure. If we look at the past, other algorithms such as RC4, MD5, DES, etc., are still in use today.

- **Fungibility** is the ability to replace an algorithm easily.

  Example As noted above, the X.509 digital certificate has an Algorithm-ID. Only that has to be changed to use another algorithm. [61]

- **Interoperability** means that some solutions must interoperate between independent implementations based purely on the information provided in the specification. Ideally, it is to select the same algorithms and suites for different protocols.

- **Updateability** The development of software components is often associated with software flaws and bugs. This also means software must be able to update itself soon. A patch or update is available. In cryptography, this means an unsecured or flawed algorithm can be fixed/replaced.

  This also means a capability to update must also ensure compatibility. "In other words, new software modules and patches should be able to operate on the same hardware as the replaced software."

  Example The OpenSSL Version 1.0.1 to 1.0.1f had a severe program error, also known as Heartbleed. The vulnerability manipulated the Heartbeat protocol. The result was accessed and read of the memory [62].

- **Flexibility** Means an implementation can flexibly change the underlying algorithm.

  Example For a hardware implementation, a FPGA is a possible solution, in contrast to an ASIC (Application-specific integrated circuit). [63]

- **Compatibility** Changes or replacements of an algorithm should run with the used components. Therefore it is recommended to test with different hardware and software components.

- **Reversibility** In the case of an unsuccessful update, the software has to fall in a specific state or be able to reverse to a previous working version. (Sidenote: Fall securely)

- **Transition Mecahnism** The transition to a new algorithm should be protected against integrity. A possible way to ensure it is using a hash of the algorithm and encrypting it with an asymmetric algorithm.
  A possible downgrade attack could be achieved if the implementation of the update algorithm is not securely taken place.

- **Backwards Compability** This property is considered for the transition period. Therefore a cryptosystem should support multiple cryptography algorithms for a smooth transition.

5.3 Do we need it?

The problem in today's cryptography is that these are based on a particular issue that is not yet solved or takes time to solve. Further, NIST made a definition of the algorithm.

- **Acceptable** is used when the algorithm and key length when "no security risk is currently known when used in accordance with any associated guidance." [64]
- **Deprecated** is used when the algorithm and key length is not recommended to use, but the usage of it has some security risks. [64]
- **Disallowed** the algorithm and key length is not allowed for further use [64]

The definition for the Status Approval by NIST lays out that the acceptance of algorithms and the key length is based on trust. However, this also means we have to make sure the used algorithm and key size are secure enough with crypto analysis and have an alternative if an algorithm is unsecured.

This also means there is a lifecycle for cryptography algorithms. For example DES has been published in 1975 and standardized under the "FIPS PUB 46", validated under the numbers FIPS46-1,FIPS46-2 and FIPS46-3 and NIST disallowed the usage beginning after 2023.

If you are interested in the lifetime of hash functions, we recommend looking at this table. [64]

5.4 General steps of a replacement

The NIST Publication [65] claims the follwoing steps for replacment algorithms.

1. Identifing
   It refers to as crypto inventory. The meaning of maintaining such an inventory of cryptographic algorithms is to look at the IT infrastructure and list all used algorithms. The NIST Publication also recommend the following steps for identification
   - legacy algorithms
   - Understand the data formats and interfaces of libraries
   - Acceleration mechanisms
   - Vurnebale communication devices
   - Cryptographic protocol dependencies

     Example The usage of TLS is also depended on the underlaying algorithms. In the OpenSSL Suite is the ECDHE-RSA AES256-SHA by using this suite you already use 4 algorithms.
2. The next step is the objects characteristics such as:

- latency and throughput
- key sizes
- Update mechanisms
- legal conditions
- interllecual property

A complete list is available in the official paper.

3. Choose a replacement algorithm
   There are undoubtedly no algorithms with the same characters, and it is also not a drop-in replacement, but the characters can help choose the right one.
4. Operational considerations
   After the algorithms are selected, the next is to make some operational considerations.
   - methods to validate the implementation
   - identify cases that need additional interims, such as hybrid or composite certificates
   - update related operational processes for developers, implementers and users
   - establish a communication plan for internal and external
   - plan the migration
   - plan the resources
   - update internal policies, standards and related documents
   - provide the necessary documentation
   - test and validate the new processes

6 HSM

As discussed before, it is of the highest importance for an enterprise to transition towards crypto agile in a post quantum world. One possible way of achieving this task is with the use of HSMs (Hardware Security Modules). The following chapter will look at the role that HSMs could play in such a transition. How they could play a vital role in the transition from current asymmetric encryption, to new quantum secure algorithms and how they can offer substantial advantages over software based solutions. For example in performance considerations, which matter in high throughput or low latency applications, such as payment systems. [66]

6.1 Introduction

6.1.1 What is an HSM?

Hardware security modules (HSM is the umbrella term which unites computing hardware, that was specifically built to perform security relevant tasks in a system. This includes, but is not limited to, tasks such as:

- Safeguarding and managing digital keys.
- Performing encryption and decryption.
- Performing strong authentication.
- Performing further cryptographic functions, such as entropy generation.

They come in various sizes and shapes, and use a variety of technologies to be integrated into existing systems. Such as having a USB interface that can directly be plugged into an existing system, being an embedded Peripheral Component Interconnect (PCI) card, or having Ethernet ports for LAN communication with the rest of the system. This allows HSMs to perform many security relevant tasks faster and more efficient than software running on all-purpose hardware, which makes them destined for applications such as mainframes or servers, which handle high throughput or low latency applications. These can usually not rely on cryptographic software to perform the tasks at hand, because the performance would not suffice. Another aspect is the security rating. Software based cryptography often does not meet the requirements for business applications, which are required by law or by their own internal policies to

employ FIPS[1](Federal Information Processing Standards) 140-3 or CC[2] (Common Criteria) rated modules. To achieve such a rating, the device must be protected against physical attacks. Those requirements arise as soon as the certification passes the first levels. This is usually required in critical infrastructure such as financial institutes, public transport digital infrastructure, government applications, etc...

This makes HSMs a viable solution for the coming age of agile cryptography. Companies will strive to implement so-called "Crypto Agility" workflows, which will allow them to change their currently employed cryptography infrastructure much easier than it was done up until now. Flexible HSM's, with this requirement in mind, can be a great asset for the future shift in the cryptographic landscape towards PQC and beyond, into a quantum agile future. The following chapters will take a closer look at the specific use cases for these applications of HSMs.

6.2 Challenges and solutions

As we face the challenge to adapt to PQC many companies will face two main problems. [67] One being that, they have no concise picture of their whole cryptographic infrastructure. This will create the need for them to perform a close inspection of their current topology, to determine where they use which cryptographic measures. While this might be a trivial task for a small enterprise, it can quickly grow with certain factors such as headcount, infrastructure, past infrastructure mergers, and many more. But this task is absolutely crucial, as cryptography can only be adapted if it is known where which assets are and how they operate together.

The second issue is the fact, that most company infrastructures are far more complex than initially thought. [67] This usually entails, that a big redesign is needed, if a company wants to change one aspect of their current security architecture. And while some might tend to simply adapt their current concept with some partial replacements or updates, it might be smart to use this challenge as an opportunity for a complete overhaul of the security architecture. Such endeavors take a substantial amount of time. Depending on the infrastructure size, this can vary from a year, up to a decade or more. If you think back to the transition from SHA1 to SHA2 or the transition from RSA to ECC, you might remember how long this took in your company. And while that change was severe, the upcoming transition will be on a larger magnitude than ever before, and past solutions for transitions might not apply here. It is always advisable to reevaluate the whole infrastructure when changes are made at such a basic layer of the architecture. This is due to the fact, that while applications and their implementations do not require a big change in architecture, the cryptographic implementation is usually the foundation of a company security pyramid, as can be seen in Figure 6.1.

[1]FIPS are public standards which are published by NIST and used to determine the security level of computer systems.

[2]The Common criteria are international standards for computer security certification.

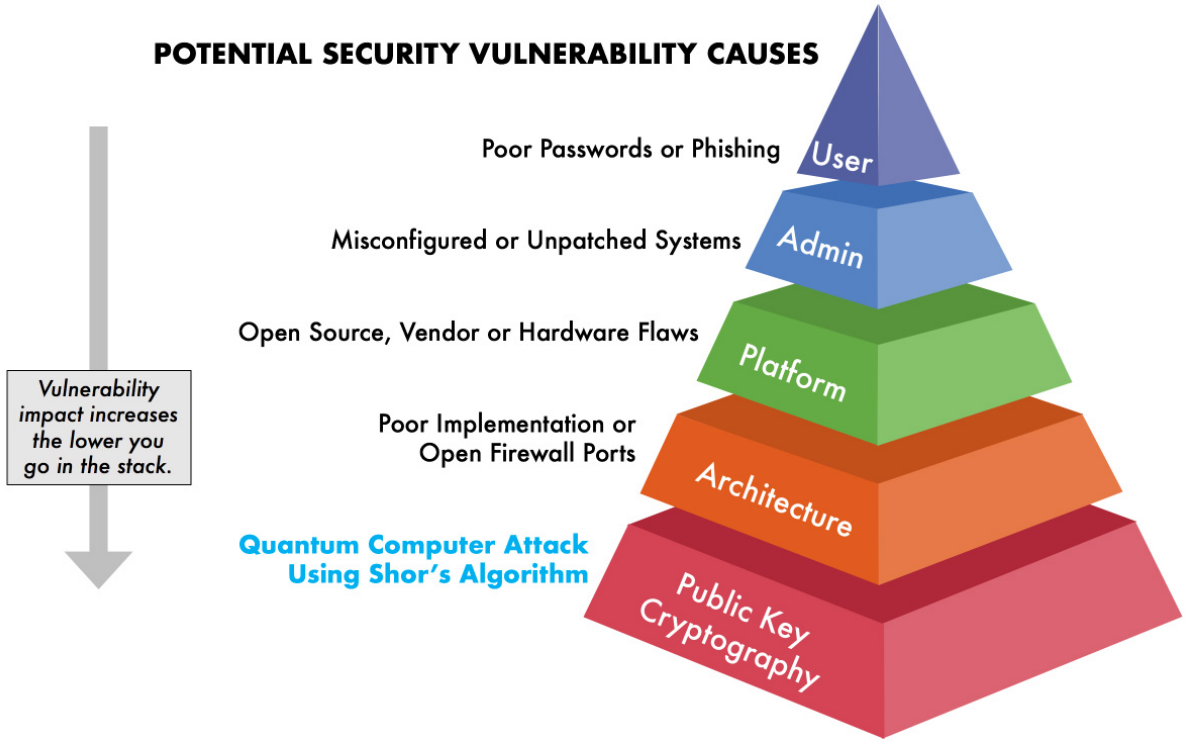

Figure 6.1: ISARA - Managing Cryptographic and Quantum Risk [67]

Furthermore, it is possible that the current architecture might simply not be ready to handle the increased workload of PQC solutions. As discussed in chapter 3, the new algorithms vary strongly in performance. Due to this, it could become necessary to provide additional performance to existing infrastructures. This can be solved by scaling vertically or horizontally[3]. But there always exists the issue, that some devices might not be capable to handle PQC or might not be suited for scaling, and thus need replacement. This is a problem that needs to be taken into consideration!

Cryptography used to be something static, and slow to change in the past. An "install and forget" kind of endeavor. But it must become an agile part of any companies IT architecture, if they want to move forward into a quantum ready future. The process of cryptography moving from something that was statically implemented and then forgotten about, to this dynamic asset in one's security architecture is called "Crypto Agility" and is described more in detail in

6.2.1 Legacy devices

As mentioned before, one of the biggest problems of the coming cryptographic shift, after the issues of cryptographic visibility and complexity, is the issue that some devices might simply not be able to adapt to the new requirements of post quantum

[3]Horizontal scaling is increasing the number of devices, while scaling vertically means to increase the power of the individual devices.

cryptography. But what if these devices are a crucial part of the infrastructure of your company? You could simply replace them with newer devices that are capable of dealing with PQC but this would require a big financial investment in the best case, and a complete redraw of your architecture in the worst case (But keep in mind to see this as an opportunity to switch to a quantum agile architecture). It could (but it absolutely shouldn't) be the case that the device is irreplaceable in the architecture, and needs to stay, without the possibility of getting it ready for PQC We could argue, that this is simply a case of a bad architecture and cannot be tolerated, but we all know, that the reality can necessitate such cases.

This is a further task, that an HSM could fulfil. If the device in question can't be upgraded, then it, and it's surrounding devices would need to be placed in an enclosed network. This should preferably be in a physically protected location, to make sure the data flow can't be hijacked and read out by an attacker. An HSM is then placed at the gateway of this network, which takes up the task of encrypting and decrypting all traffic that goes in and out of this network. The advantages are clear, as the non PQC ready pieces of hardware traffic is encrypted when sent out of the network in a quantum safe manner. But the drawbacks include the aspect that all traffic inside this network is not quantum secure and can possibly be intercepted and decrypted by a possible attacker with quantum resources. This would require a breach of the secured perimeter, be it physical or digitally.

6.2.2 High throughput/low latency environments

One of the main applications, in which HSMs already are used today in the security architecture, is environments in which low latency or high throughput are crucial. These HSMs are usually tailored to suit the exact use case they are intended for, with specific hardware implementations. This makes them unable to adapt to a new cypher suite. A further hindrance is that these devices often come with preinstalled firmware, which cannot be changed, modified, or updated to accommodate new cyphers or cryptographic processes. Microsoft noted this in their picnic implementation paper:

> Often the firmware on these devices (HSMs) is fixed by the manufacturer, and prototyping new algorithms is not possible. [68, p. 47]

This is a clear drawback for a customer looking for quantum agile solutions. Manufacturers are thus preparing for this new challenge. While it usually sufficed to build well tailored solutions, which solved one task especially well, often due to the help of hardware implementations. It has become increasingly important to have a product which can fit in the new crypto agile landscape. This can be achieved through various means. But due to the fact that HSMs typically heavily rely on the use of hardware acceleration, the manufacturers need to use FPGAs to enable their customers the best of both worlds: Agility and performance. [68, p. 47] While some manufacturers already implemented this in their current product lines, others are slower to adapt. But more on this in the following sections.

6.3 Quantum ready HSMs

As we have seen before, HSMs are already a basic part of many modern architectures. This won't change with the migration to post quantum cryptography, it will probably even increase, as they can tackle some challenges provided by PQC better than their software counterparts. But what exactly does it mean when we speak about "Quantum ready HSMs"? Which criteria should they fulfill to be considered a strong contender in the field? There isn't really a specification what these modules should provide to be considered post quantum ready, but a whitepaper of MTG makes a good point. They specify three main points which their HSMs aim to fulfil to perform with PQC These are:

- Strong protection and use of PQC keys
- Customized hardware solution
- Compliance with industry standards

[69]

So I used these three points to paint a picture of PQC ready HSMs.

6.3.1 Strong protection and use of PQC keys

Key security always was HSMs main goal. It's one of the areas where they are clearly superior to software based solutions, due to the fact that they can offer physical protection in a way a software product simply can't. This fact hasn't changed with PQC but has simply remained an important part of an HSM Another aspect of this is the secure use of the keys with the new algorithms. This can be done in a similar fashion as before, but needs to be adapted to the current PQC algorithms.

6.3.2 Customized hardware solution

Since current HSMs only strive to implement a very narrow group or even just one specific algorithm, they can allow themselves to be rigid in their hardware implementation. This allows them to offer just the main building parts of these algorithms within their hardware platform. But with PQC and especially the move towards crypto agility, it becomes increasingly important for HSMs to have an adaptable hardware implementation. It would not be a feasible approach to implement all current PQC candidates on the hardware, since there would be too many uncertainties, and it would overcomplicate the internal architecture of the device. What can be done instead is the integration of FPGAs with a new firmware, that allows for a change in algorithms during it's operation and thus allows for an agile HSM preparing it for the post quantum future.

6.3.3 Compliance with industry standards

This is something that is directly influenced by the previous part. While the hardware and software has become more agile, it also has become more prone to attacks. The firmware now needs to be updated for new algorithms and the FPGAs need to be reconfigured by this firmware. And while this was simply impossible before, offering the HSMs a great way to increase their security, it is now a necessary part of a quantum ready HSM This offers additional attack surface. Some industry standards, such as FIPS and CC need to implement this in their new regulations and need to evaluate the vendors solutions to combat this new risk, so that the HSMs can be certified as before. While some vendors have already started this process with their quantum ready HSMs, such as Crypto4A [70], it remains to be seen if they can obtain a FIPS certification, and how this process will develop in the future.. The NIST itself is currently in the process of making sure that these developments which ocurred in the PQC process are implemented in future standards for certification. They state that some PQC aspects are simply out of scope for FIPS [71]. When checked with the FIPS140-3 standard [72], it is clear that it was not designed to incorporate PQC and thus is in need of a future change to address this issue. The NIST has not released an official statement when this will be the case, but they did state:

> Additionally, NIST plans to incorporate a cleaner, and therefore preferable, hybrid key establishment construction in a future revision of SP 800-56C [71]

SP 800-56C isn't a full-fledged certification standard like FIPS140-3, but a recommendation by NIST, with guidelines for "key-derivation methods in key-establishment schemes". This is an important building block of modern HSMs and in turn has an influence in the FIPS certification process. So while it isn't a FIPS140-3 change, or a new 140-4 proposal, it seems as if they are aiming to implement guidelines on certain PQC schemes, which will propagate to FIPS standards in the future. This will probably occur once the standardization of the PQC algorithms is completed.

6.4 Manufacturers

Disclaimer

Everything included in this chapter is based on marketing material of hardware manufacturers. We are well aware that this isn't the same grade of information quality as scientific evidence. This chapter still tries to give an unbiased opinion of current possible PQC HSMs manufacturers. All marketing material referenced in this section has been cataloged and cited for your convenience.

The manufacturers are not ranked in any form and are simply arranged in alphanumeric order. Also, the list is in no way complete. This is not the stated aim of this chapter. It simply aims to provide an overview on a few manufacturers who have published their current approach to the quantum threat.

6.4.1 Crypto 4A

Crypto 4A is a Canadian manufacturer of cryptographic solutions, including state of the art HSMs. Their production for the fourth generation of their HSMs recently started. This generation uses top-notch FPGAs to enable an agile and quantum ready HSM They have a clear focus to provide the HSM solutions of the future and strive to have a complete product palette for any post quantum security needs. For which, they are currently in the process of obtaining a FIPS140-3 certification. [70]

Crypto4A, is a great example for a manufacturer who wants to be an early adapter in this field. They saw the need for agile HSMs in the future and acted on it, showing how early PQC HSMs could look like.

6.4.2 ISARA

ISARA is not a manufacturer of HSMs but rather the software and firmware with which HSMs can function in an architecture. And while they don't provide security hardware, they are an important mention in this field, due to their HSM compatible software suites. They strive to enable a quantum ready future for security products, be it their own software, or their software suites in combination with Thales hardware. This is achieved by a strong collaboration with Thales to enable an optimal product portfolio. [73] [67]

ISARA is an interesting mention, since it will become increasingly important to have a good firmware, that is capable of adapting to customer needs. As we discussed previously, an agile HSM can only function with a good firmware suite running on it. This helps it to unlock it's agile potential.

6.4.3 MTG

> MTG offers a comprehensive portfolio of state-of-the-art quantum-safe security products and services [69]

They are, just like ISARA, a software provider and not a hardware manufacturer. They offer services such as key management interoperability for different HSMs and Certification services, etc... Their HSM partner is Ultimaco, with which they follow a similar goal such as the ISARA & Thales cooperation. The main aim being to provide a complete crypto agile solution for the coming PQC infrastructures.

MTG is another great example of manufacturers realizing the importance of agile firmware, in combination with powerful hardware.

6.4.4 Securosys

Securosys is a Swiss company which specialized in the production of HSMs. While securosys aknowledges the quantum threat, they haven't officially stated any intentions to start develop a new series of HSMs specifically for it. With their current stance being, that they simply don't see any urgency to develop new HSMs specifically for PQC at the moment. Though they offer the possibility to implement PQC on their current devices. This more timid approach can make sense, since we do not know which algorithms will be standardized and in what time frame. It remains to be seen which products they will provide specifically for it, once they will start their development. [74]

The approach of Securosys, while different, is still valid. They might not have the new cutting edge quantum agile products at the ready in the next few years (Or maybe they do, but simply are keeping quiet about it). But since they wager that their current lineup can handle PQC they are nevertheless prepared for the initial PQC rollout. They might not be as agile or performant as the competition, but it remains to be seen how highly this will be valued by customers to begin with.

6.4.5 Thales

Thales is following a similar tactic like Securosys. But they are heavily advertising for their new post quantum solutions. Their current line, the "Thales Luna HSM" which they claim to be able to implement post quantum technologies as they emerge. [75] This is a great promise, but we will see if it holds true in the future, and for how long the current generation will be able to keep up with the new standards in this sector. [66] If they want to keep the edge over their competition, they will need to release new hardware, which can bring the fight to companies like Ultimaco and Crypto 4A.

As said before, this is a valid approach. If your current product line is strong enough to handle the increased workload of PQC But it will surely fall short in performance to newer HSMs with integrated FPGAs. The possibility remains to implement the new NIST algorithms once they are standardized with specific hardware. This isn't the spirit of crypto agility, but it would offer substantial performance boosts. And maybe, Thales is aiming towards that possibility.

6.4.6 Ultimaco

Ultimaco is probably the current leader when it comes to the development of post quantum HSMs. According to their own claims, they were the first manufacturer to release a PQC firmware extension for their existing HSMs. [76] They have partnered up with companies such as MTG and Microsoft to create the cryptographic product variety of tomorrow. Their products were even used for the first fully quantum secure VPN connection. [77] This, combined with their clear road map, shows how determined a company can be to provide a strong HSM lineup for possible PQC solutions. They

achieve this by combining FPGAs and a strong software suite, which was created in collaboration with MTG, to provide the crypto agile solutions of tomorrow. [78]

Ultimaco, just like Crypto4A, shows a good picture of everything that a post quantum agile HSM strives to be. Most importantly they seem to have a complete vision with strong industrial partners for firmware and field testing. It remains interesting to see where this cutting-edge development will lead them.

6.5 Conclusion

Disclaimer

Everything included in this chapter is merely an opinion by the authors and is not supported by hard evidence or direct quotations of relevant literature. The opinion is based on the research documented in the previous chapter 6.

Cryptography is the foundation of a company's security architecture. And it is quite fitting to visualize the whole topology as a pyramid, like Figure 6.1 If the foundation cracks, the whole pyramid fails. HSMs are important cornerstones in today's large scale architectures, but while some already saw them loose relevance in certain use cases, due to the adaptability of software based solutions, it seems as if they will become greatly important over the next few years. The rise of quantum computing will enable the use of HSMs in nearly every scenario to improve a company's crypto agility, and thus enabling it to prepare for the quantum age.

While they shouldn't be treated as an universal solution for everything, they can prove to be highly useful. In the right application with the right surrounding system an HSM can become a powerful tool to implement PQC in an enterprise's perimeter and enable it to tackle the various challenges of todays IT infrastructures with a powerful tool at their disposal. Manufacturers around the world have acknowledged the imminent threat of quantum computing, and are thus preparing for it in their own way. Some identified the need for a new generation of HSMs to fulfill this task. They have started to build HSMs which will enable enterprises to confidently advance into a crypto agile future. Others remain with their current solutions, and will adapt them along the way.

All smaller enterprises which haven't implemented HSMs so far should reconsider their stance on them, and see where they could profit from them. While larger companies, which usually already use them, will see that they will be able to redesign their security architecture with HSMs at their core. They will be able to find new use cases for them, and to further develop current use cases.

7 QKD

While HSMs are great for high throughput applications, there are other solutions to achieve similar tasks in certain environments. The technology discussed in this chapter, QKD(Quantum Key Distribution), is the only one in this essay, that actually uses quantum states to encrypt information. But while this sounds ever so promising, it also has it's limitations, which we will discuss later on.

The following chapter will give an overview over QKD Please note that in some articles, you will also read the word quantum key exchange (QKD), which is a synonym for the same process. While current cryptographic system rely on classical computers to perform safe operations for key generation and exchange, QKD does this with the laws of quantum physics. This brings the advantage that they can leverage quantum properties such as the spin and entanglement to create nearly unbreakable cryptographic algorithms.

Since it is theoretically impossible to clone quantum states [79, p. -37], due to the no cloning theorem, it is impossible to physically circumvent this kind of cryptography. However implementing these systems always brings the chance of error with them, and thus they need adequate protection against possible eavesdroppers.

This chapter will take a look at the current technical development of this technology, the advancements made in recent years, and give an outlook for future challenges and opportunities to come. It will provide an estimation of relevance and feasibility for different business types in which QKD could be used. To understand this technology, we first have to take a little look at basic quantum mechanics

7.1 What is QKD? (Exemplified with BB84

Quantum key distribution is the process of using cryptographic protocols that leverage quantum physics for their key generation. That means they use quantum particles sent over a channel to exchange key material with the participants.

The BB84(Named after his Inventors, Charles H. Bennett and Gilles Brassard in 1984), was the firs quantum cryptography protocol. It was defined in theory in 1984 and tested in a lab in 1989.

The algorithm works as follows (For a better understanding you can look at Figure 7.1):

1. Two participants, let's call them Alice and Bob want to perform a secure key exchange over an insecure channel.
2. So alice will provide Bob a key. She needs a source of single quantum particles. BB84uses spinning photons[1].
3. Alice will now pass photons through one of the two filters and send them to Bob.
4. Bob measures them with a random filter of his choice and notes if the receives a logical 1 or 0.
5. After they have exchanged a sufficient number of photons for the key exchange [2] they will compare their filter settings.
6. If both choose the same filter they keep the value, if they choose different filter the measurement is not reliable and thus discarded.
7. They perform error correction.
8. Now they both have the same key which could not have been intercepted by anyone.

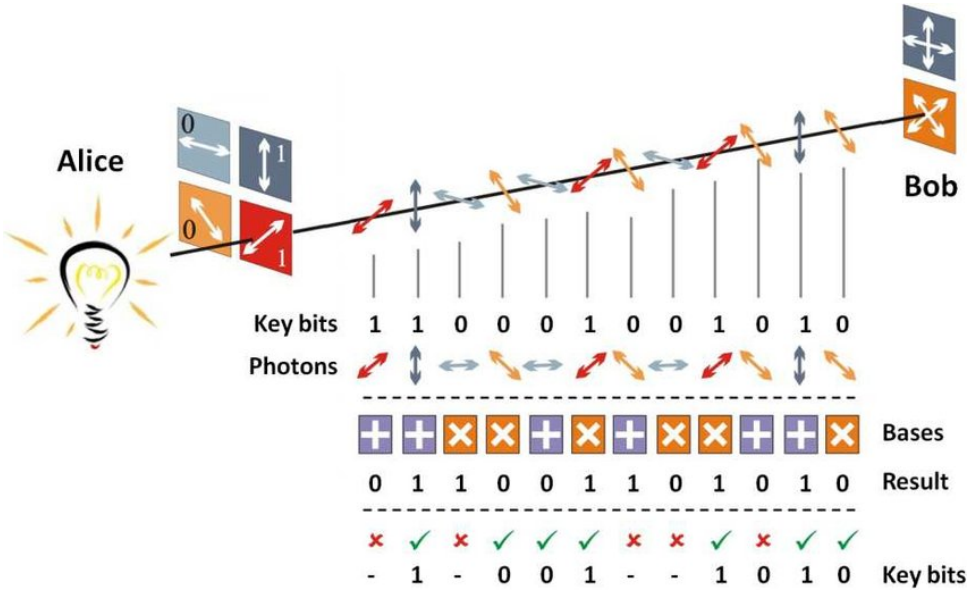

Figure 7.1: BB84protocol basic scheme

The possible attack on this algorithm:

[1] Photons are quantum particles with a spin. This spin can be measured by passing them through a filter. Normal filter alignments are -45°/45°(diagonal) and 0°/90° rectilinear. Each of the two filters has a 0 and 1 position depending on the spin of the passing photon. If a photon encounters a filter which is approximately similar to their spin they pass through without being absorbed. The closer the filter they encounter is to a 90° shift from their current spin, the more likely it is they will be absorbed (and thus not generate a signal at all). In real life photons can spin any possible way, but since in QKD we generate our own photons, we can generate them with a certain spin.

[2] Usually at least double the key size to account for Bobs random filter choice, and an additional amount to overcome the photon loss due to background noise in the fiber.

1. Let's imagine someone wants to attack this key exchange, let's call that person Eve.
2. Eve can insert herself into the channel between the two participants.
3. As she doesn't know which filter to use herself, she also has to choose at random.
4. This leaves her with a 50/50 chance of picking the wrong filter, and subsequently destroying the quantum particle during the measurement.
5. It will thus not travel to Bob and will be discarded.
6. Even if Eve manages to pick the correct filters these cases can arise:
7.
    - All three choose the same filter: The bit is kept and Eve has learned the key information.
    - Alice and Eve choose the same filter, but Bob didn't: Eve learned the key information, but it will be discarded since Bob didn't measure it correctly.
    - Alice and Bob picked the same filter, but Eve didn't: Eve measures (or even destroys) the photon, but the measurement is with the wrong filter and provides her no information about the key. If she destroys it, the measurement is not recorded by Bob, it gets discarded by Alice.
    - Eve and Bob picked the same filter, but Alice didn't: Alice and Bob did not pick the same filter, they will discard the measurement.
8. So as you can see, there is no chance for Eve to measure the key correctly. Unless she could clone the particles, which is not possible due to quantum physical laws. Or if she would choose the correct filter every time, but that chance diminishes exponentially. Were she to try this with a AES 256 bit key, and we assume the participants exchange only 600 bits which would be a close call regarding chance of correct filter choice and noise in the channel, her chances of picking the correct filter each time would be $1 : 4.15e^{180}$ which is simply impossible.

Hopefully this very brief overview over the BB84algorithm gave you an introduction on how it works. If you are still in need of additional material, we highly recommend watching this explanatory video on the topic. You can also find countless detailed write-ups of the topic, such as this wikipedia article.

7.2 History of QKD

The idea of quantum cryptography is nothing new. The first ideas for this type of cryptography emerged nearly 40 years ago.

But development hasn't stood at a standstill during all those years. Here is a timeline of the most notable events in quantum cryptography:

So as you can see, QKD came a long way in the last 40 years. From barely managing to pass 30 cm of fiber, to a couple of hundred kilometers with a reliable service.

Year	Topic
approx. 1970	Work on quantum algorithms starts. Stephen Wiesner and Gilles Brassard start to research the topic.
1984	BB84Algorithm is published. The first quantum cryptography algorithm. See section 7.1
1989	BB84is experimentally proven in a lab environment.
1991	The E91 algorithm is published by Arthur Ekert. It is similar to BB84, but uses entangled quantum particles.
2004	A quantum key is exchange in a lab environment over 360 m with entangled photons.
2006	Two students of the MIT manage to eavesdrop a BB84message, rendering it insecure. Showing the need for a possible mitigation to attackers
2007	Elections results in Switzerland are being sent over a channel with QKD The key is established from Geneva to Bern (the capital) over a 100 km long fiber.
2016	A Chinese research group launches a QKD capable satellite into orbit and performs QKD over a period of two years. The keys are transmitted over 1200 km of space
2016-2021	The research in QKD is intensifying in different aspects. See section 7.3

Table 7.1: The history of QKD

During this time, new ways of performing QKD have emerged, new algorithms have been formed and developed. But up until now it was just a secure way of sharing keys, without many practical use cases, since our current crypto algorithms were already a secure and mainly cheaper alternative. The only real use cases were few and far between, mainly directed towards research institutions and government agencies, which opted for the highly secure cryptography solution, since they could pay the higher costs. The market was pretty niche, with only a handful of companies offering hardware for QKD The swiss company ID Quantique probably being the most well know. But now, with the threat of quantum computer looming at the horizon, QKD has become a viable contender for PQC In the recent years (2016-2021) many new papers have been published regarding this area, as the investments of private companies and government funded research agencies have increased. We will take a look at the most important areas of current research and future developments in the following chapters.

7.3 Current technical advances

7.3.1 Development fields

To better understand, why the development is still ongoing, it needs to be noted that QKD still has many flaws that need fixing before it can become a good contender for PQC These drawbacks currently prevent if from being used in certain situations. They mainly entail, but are not limited to:

- Cost
  - The equipment for creating pulses with single quantum particles need to be very precise, which makes it expensive
  - QKD needs purpose-built fiber channels to work and cannot be run on conventional infrastructure which increases cost
- Robustness
  - If cheaper equipment is used, such as prebuilt fibre infrastructure, the reliability drops substantially
  - Quantum particles are highly fragile and due to that, prone to loss in a fiber, even under optimal conditions
- Distance
  - Due to being prone to loss in a fiber, the distance is limited, capping it's effective usable distance at around 200 km [3]
  - Solutions above that become increasingly complex
- Key Rate
  - The key rates drop quickly with increased distances (Due to increasing loss)
  - Lower key rates make key exchanges last longer, which reduces usability

So the research teams who aim to improve QKD clearly have their work cut out for them. We will look at a few focus points of current research and which advances have been made so far.

[3] It is highly debatable where this "barrier" exactly lies. As many research papers claim different numbers. So this isn't to be considered a set number, but more of a range.

7.4 Increasing the distance

One of, if not the biggest, drawbacks that QKD still has is the limited range. Single photons can only be sent over such a long distance before they are absorbed by an imperfection of the fiber, or the reflective coating itself. This happens due to quantum scattering, which is not important for this essay, you should simply know that it is incredibly hard to send single photons over long fibers without them getting absorbed at some point.

So this shows why the first experiment was done over such a short distance, and all subsequent experiments failed to reliable crack the 400 km [80, p. 4] mark at all.

It is thus not surprising that a lot of the published research papers from recent years, are about overcoming this barrier. Be it with the use of new approaches to already existing protocols, new protocols, or even new transmission techniques and technologies.

You might ask yourself how it comes that sending quantum particles is much more complex than sending conventional data via copper cables or fiber. There are many factors:

- Quantum particles are usually sent by transmitting **single** photons, compared to a couple of thousands(if not millions) in conventional cases.
- Conventional signals can simply be read out and repeated, or even cloned. Reading them and repeating them is not as easy for quantum particles, while cloning is physically impossible.
- Even the tiniest background noise can destroy a quantum particle, while it takes substantially more background noise to disrupt a conventional signal.

Currently, there are a few different approaches on how to fix this:

- We can produce ultra lossless fibers (ULF) which are incredibly pure, but also highly expensive.
- We could use quantum repeaters to increase the signal strength. But the technology for quantum repeaters isn't fully ready yet.
- We can send the quantum particles over channels that have less background noise compared to classical optical fibers, such as clean air, or space.
- We can adapt new protocols that are more tolerant to background noise.

7.4.1 Quantum repeaters

Now the solution with quantum repeaters seems pretty easy to implement. We could just use quantum repeaters to solve the problem of distance, just as we did with the conventional internet, to increase distance for transmission. This concept is proven and one of the easiest ways to overcome physical boundaries. Repeaters are used everywhere in our daily life. Radio transmissions, satellite communication, Wi-Fi range increase and the list goes on and on.

Firstly we need to distinguish what "quantum repeating" is all about. Because there are two ways of repeating quantum information. For one, we can use a device that actually entangles[4] quantum objects to repeat their properties, this guarantees us that the information is always in a quantum state along the way, and thus tamper resistant. Or we can read it out, and create a new quantum state with the use of conventional hardware. While this second approach is way easier, it also mitigates the advantages of QKD since we interrupt the pure quantum channel. We offer an attack surface. So please keep in mind, that we are mainly talking about option one, true quantum repeaters, in this section.

Repeating quantum particles is not that easy. Because we can't simply clone quantum states, due to the no cloning theorem. This theorem helps us to stay safe from an eavesdropper, but it also makes it so much harder to use a repeater for our quantum particles.

There is a whole field of study dedicated to quantum repeaters. The current approach that is being taken, is to entangle the quantum particles in intervals [81]. It thus increases the distance travelled by the quantum particles with each repeating process. This allows to preserve the quantum particles entanglement over increasing distances with the help of additional particles. As this process continues, it can be performed with as many quantum repeaters as needed along the way, until we have 2 quantum particles that are far enough apart for our desired connection distance. Every particle sent over the channel needs to go through this repeater process.

You see this is no easy task, and relies on technology that has not been fine-tuned, or even developed. So one ought to find other solutions for this problem. Why not turn to the second option, classical repeaters?

7.4.2 Classical repeaters

We could measure each quantum particle, destroying it in the process, and then recreate them. This would make the whole process a lot easier, and we could use existing hardware to perform it. Conventional networks perform this task in a similar fashion on

[4] Quantum entanglement, is the quantum mechanical linking of two quantum particles. This is such as that when measuring one, the properties of the other one are known without measuring it.

switches or bridges on layer two.

And this is exactly what is done in China's first quantum Network, spanning from Beijing to Shanghai. They built relay stations roughly every 65 km [82] which measure the signal, destroying it in the process, and then recreate it. While this is not ideal for security, making the whole quantum link vulnerable at these locations, it narrows the possible attack locations down to a couple of repeater nodes. The solution is far from optimal, but since there are currently no better alternatives, this might just be how it's going to be done in the near future until a better technology enters service. Please note that this technique is substantially different quantum repeaters. Quantum repeaters aim to duplicate the state of the quantum particle without destroying the original.

7.4.3 Increasing the distance without repeaters

Another approach is to try to increase the transmitting distance without the need for repeaters. Because if we riddle our network with repeater nodes (be it quantum or not) it would mitigate a lot of the advantages that QKD or even a possible quantum network (more in section 7.5), could provide. Nodes slow it down (and introduce possible failure points), make it more vulnerable to threats from the outside, and also harder and more expensive to maintain.

What if we can never achieve reliable quantum repeaters? As with a lot of things in the quantum physical field of studies, it holds a high uncertainty if we can reliably achieve it in the next couple of years. Regardless of this possibility, the scientific community is looking into ways to increase the transmission distance of QKD without repeaters, to broaden its use. Because if quantum repeaters never become viable, they will have a solution ready for it, and if they will, it still stands to the debate if they will be reliable and don't introduce unwanted drawbacks. One of the most promising topics, and the current record holder when it comes to distance, is the so called twin field quantum key distribution algorithm (TF-QKD.

7.4.4 Twin Field Quantum Key Distribution

The approach of TF-QKD has a couple of differences to classical QKD It does not rely on controlled receivers, but instead simply uses an untrustworthy node which contains the receiving equipment. This was inspired by the measurement-device-independent quantum key distribution MDI-QKD another protocol, which aimed to increase QKD transmission safety and distance. MDI-QKD is not subject of this essay, if you want to know more about it, you can read this paper.

In TF-QKD as opposed to BB84, both participants send their photon pulses to the same destination (an untrustworthy recipient), which then measures the state and informs the participants if they have achieved a match or not. Since the untrustworthy node only sees if the participants achieved a match and not the individual phases of the individual signals, it cannot determine the key. And in a further deviation from the base idea of BB84, TF-QKD uses phase modulation to alter the photon state, leveraging the effect that a photon isn't just a particle, but also a wave at the same time. While this increases the complexity of the protocol, it greatly helps to achieve further distances and increases security. This can be seen by the phase modulators (PM) on each participants side in Figure 7.2. Because if an eavesdropper would measure the signal along the fiber channel, he could not determine which phase the particle is in, making the measurement void for him and even risking destroying the particle in the process, alerting the participants of a possible eavesdropper.

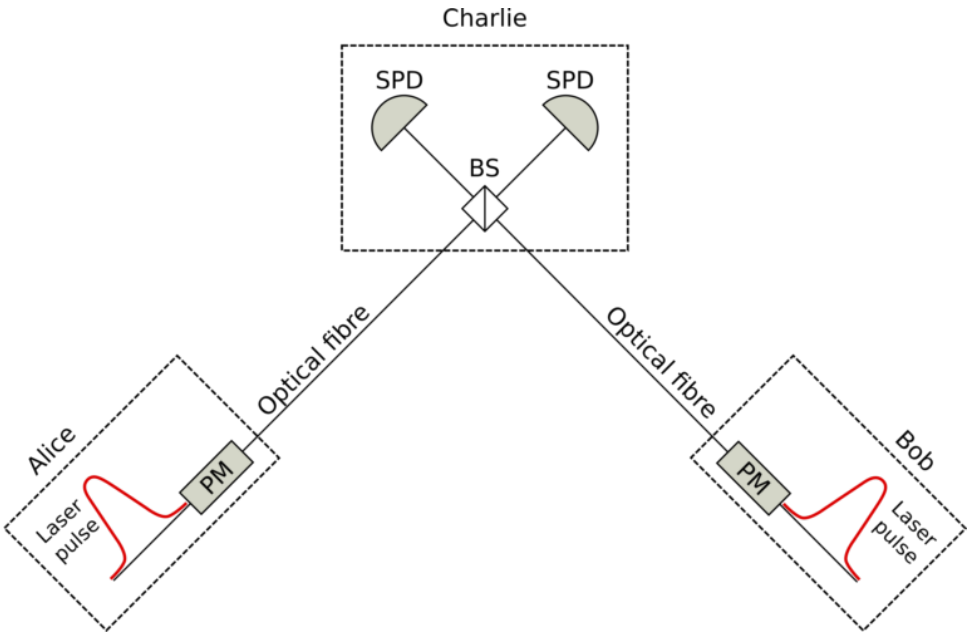

Figure 7.2: The usual setup of TF-QKD

This technology has been experimentally tested under lab conditions by Toshiba Europe research. While they tested the implementation, they did not do this over an actual 500 km of fiber, but rather over less than 100 m, with artificial attenuation added to the fiber channels. These learnings concluded in a paper [83] over their specific implementation, which provides insight in how they managed to achieve high key rates over such huge simulated distances. The paper [83] furthermore looked at ways of eradicating possible pitfalls for real world implementations of the protocol, as it acknowledged that their laboratory implementation had certain advantages which could not be counted on in the real world, such as low background noise and no phase shift due to reduced cable length.

These learnings from Toshiba Europe were picked up by a Chinese research group in 2020, and used to test an implementation of the TW-QKD in a real-world environment. The research group managed to establish a link between the Chinese cities of Jinan

and Qingdao, which are over 400 km apart. This proved the effectiveness of TF-QKD in a practical use case. They stated that his proves a possibility for TF-QKD to be operated on distances of over 500 km. The measured key rate is around thrice as good as previous QKD field tests with similar lengths. [84, p. 2] This is nearly double the distance of comparable implementations of QKD and proves that QKD is ready to be implemented in nationwide networks sooner than later.

7.4.5 Satellite-to-ground QKD

7.4.5.1 Idea

A whole new approach to overcoming the rate-distance barrier of QKD is the use of satellites. The basic principle of QKD stays untouched. We have two entities that want to generate a key by leveraging quantum physics as a means of provable security. Now instead of having one of them operate as the sender and the other as the receiver of the key, they both act as receiver. The sender in this case is the satellite, which aims a laser at a ground station below. It then transmits a beam of entangled photons, which are directed at a ground station with a photon receiver connected to a large-scale telescope. This telescope gathers the photons sent by the satellite and channels them into the quantum receiver, where they are measured and converted into a bit stream. The satellite then continues his orbit and waits until it is above the second receiver, where it can send the entangled photons to the second ground station. The two ground stations can now negotiate a key, which they derived from the received bits from the satellite.

7.4.5.2 Chinese research results

Chinese researches tested this technology with a satellite launched in 2016, carrying the necessary equipment to establish QKD with ground stations. During the mission duration of two years, they conducted a variety of experiments in correspondence with three ground stations in China and one in Austria. The goal of this experiment was to be a proof of concept for QKD and quantum communication in general over such a long distance. To achieve QKD with the ground, the satellite had to send the signal over a distance of 1200 km.

So how could this technology help to increase the usability of QKD? It provides the possibility for a key exchange between two ground stations which can be anywhere on an orbit around earth. This is done without the need for a complex infrastructure in between these two locations, needed to transport quantum signals over a fiber or the disruptive atmosphere. The only infrastructure needed for this is a ground station each at the receiving locations and the satellite itself.

In the beginning of the large-scale use of QKD as solution for PQC large scale quantum networks will be few and far in between. Only the drivers of the technology will be able to afford the construction and maintenance of large scale quantum networks. It

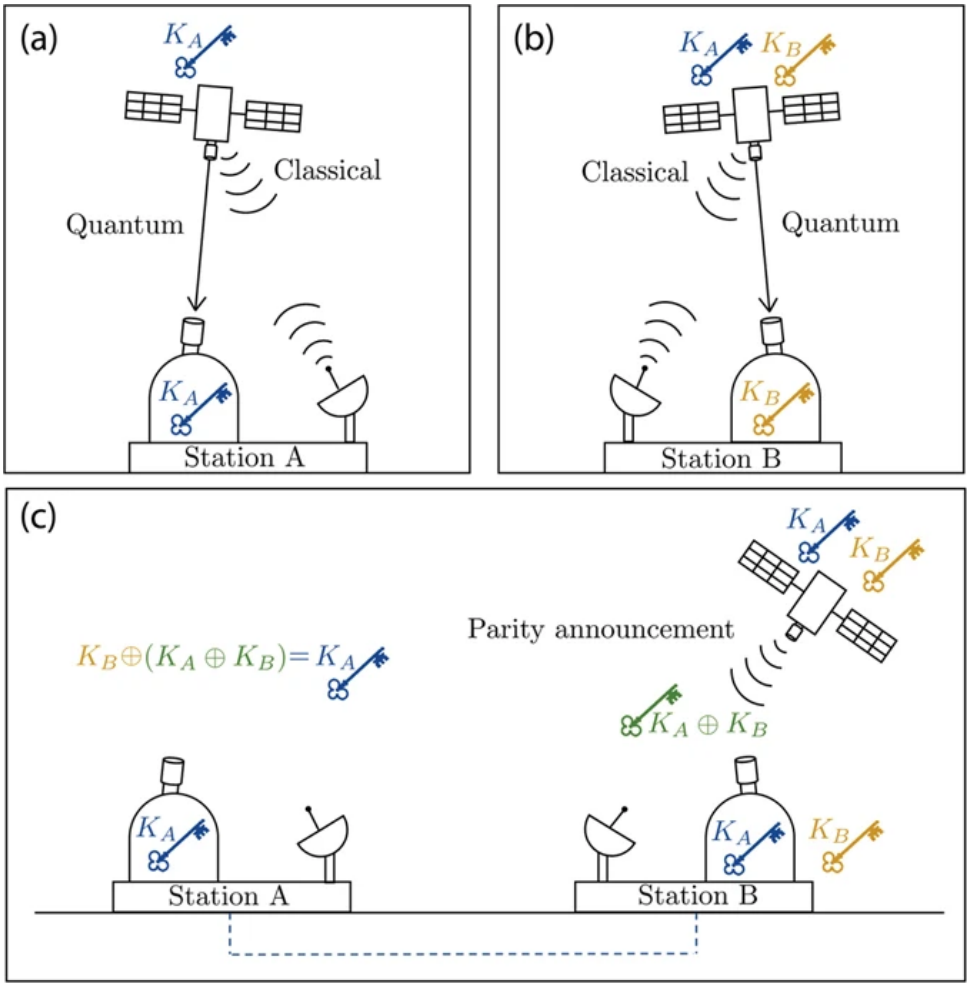

Figure 7.3: Flying trusted-node QKD scheme, a special protocol of satellite QKD

could thus serve as a solution to provide QKD between local quantum networks that only span cities, regions or at best nations.

As QKD becomes more widely adapted, and we will be able to deploy nation- or even continent wide quantum networks, there still remains the challenge of providing QKD across continental borders with no land connection. Specifically between Eurasia/Africa, the Americas and Australia. While the current internet uses submarine optical fiber cables for this task, it is currently uncertain if this will ever be possible for quantum networks. Current QKD technologies cannot operate on such fibers, and the necessary infrastructure for a long quantum link is not capable of being built on the ocean floor. So it could be a feasible solution to perform this task via satellite QKD

As of now, there are still a couple shortcomings of this technology for it to be considered a viable staple in the future of post quantum cryptography, such as, but not limited to:

- The satellite's key distribution rate is reliant on the weather conditions.
- There needs to be a big satellite network for it to span the whole globe (Think GPS but with a finer mesh.)
- Currently, only big telescopes with an aperture of roughly 1 m are able to reliable receive the signal.
- The infrastructure costs for receivers and the satellite are quite high at the moment
- Long transmission time for a key pair (Due to orbits and number of satellites)

These issues will need to be addressed before this technology can become an everyday use item, like e.g. GPS But the research team working on satellite QKD is well aware of this and has already stated a couple of improvements to their technology for future tests. For example, they plan to deploy a whole constellation of satellites, which would thus speed up the process and enable quicker key exchanges between far apart ground stations. A further measure to increase the key distribution properties and the availability for the intercontinental key exchange is the increase of the orbit height. With a higher orbit, the satellites would be more stable, allowing for more precise QKD with the ground stations. This would allow for cheaper receivers, better transmission rates and a bigger cone which could effectively be serviced below the satellite. [85, p. 10]

They make it clear that their main goal is not to provide QKD for every individual institution. This is simply not possible in the coming years with the current status of the technology. But the technology holds great promise for a previously mentioned use case, the interconnection of metropolitan areas:

> The satellite-based QKD can be linked to metropolitan quantum networks where fibers are sufficient and convenient to connect numerous users within a city at 100 km scale. [85, p. 10]

7.4.5.3 Further key developers

An indication that this technology is seen as a promising solution to the rate/distance problem can be seen in the fact that other nations have started to invest into satellite QKD as well. The European communication infrastructure giant SES is already working on this technology. They recently teamed up with the European Space Agency (ESA) to develop a system for the creation and use of QKD via satellite. [86] Currently, they do not make any concrete statement when they will be ready with a fully integrated system for customer use. But they have recently announced the creation of a consortium of 10 Partners to support the so called "QUARTZ" initiative which has the consumer market as the main priority. They stated in a press release:

> QUARTZ applications will address the needs of users such as telecommunication operators, financial organizations, infrastructure providers, institutions and governmental organizations. [87]

Which would make them a promising future provider for QKD on a consumer level.

Another promising up and comer is the UK based company Arqit. According to their own statements, they will soon be ready to launch their own QKD satellite, and have already developed most of the technology needed, including a completely new QKD protocol tailored to the satellite's needs. They are planning to launch their first two satellites by 2023 from the newly created spaceport in Cornwall. They have strong collaboration partners as well for this project, such as the UK telecommunications company BT and the U.S defense company Northrop Grumman. Their ambitious aim is to launch a QKD service by 2023, which they are confident about :

> But Williams (who is also a founder and former CEO of communication satellite operator Avanti) told Space.com that Arqit is "far ahead of the world in launching a commercial [quantum key distribution] service." [88]

So we can expect them to provide usable results in the very near future for consumer QKD solutions.

7.5 A possible quantum internet

Why are we undertaking all this effort to send quantum particles over longer and longer distances? Wouldn't it just suffice to send them over 200 km? We could simply use QKD for what it was originally designed. A short-distance key exchange that is provable secure. But researches are working towards a way bigger goal that just a little bit of key exchange. The quantum internet. What they are trying to do is to create a big network, comparable to the internet, which only leverages quantum states to communicate. This would bring two key advantages with it:

1. Speed: A quantum internet would have quicker response times
2. Security: A quantum internet would be provable secure against eavesdroppers, because we can leverage quantum physical processes to ensure this property

This new internet would allow many real-world applications to be run substantially faster. An interesting aspect for applications such as:

- Sensor Networks
- Upscaling Quantum Computing
- Secure Quantum Communication

While these applications hold great value for the scientific community, they do not provide much of an impact for the industry. But the aspect of having a worldwide network that leverages quantum states for transmission could be a possible solution for post quantum cryptography. If we transmit data over a classical link, we need to protect the nodes of the networks and make sure that no one attacks the links between them. The quantum internet would fully mitigate the need for link security, as data in transit is now being sent with quantum particles, which makes it impossible to eavesdrop on it. It could also help to increase node security, with technologies discussed above (Such as quantum repeaters). Since with the use of quantum repeaters, even the nodes would leverage quantum states. This would make attacking the nodes harder, if not impossible, without being detected and destroying the data itself, due to quantum physical laws.

7.6 Quantum conference key agreement

We talked a lot about QKD in the previous sections and the achievement of building large scale quantum networks, even up to a global quantum internet. But all the solutions that we discussed so far for QKD have one thing in common. They were all just peer-to-peer. So were we to connect a group of users together with the security of QKD we would need to perform QKD between each pair of peers. This does not scale well. In a network with n peers, this would mean we would need to perform $N(N-1)/2$ key exchanges. As you see this scales exponentially with the number N which means it starts to increase very quickly as N grows even slightly. Researches encountered this problem early on as they started to build the first inter campus networks that spanned more than just 2 peers. So they set out to create new protocols and implementation for QKD which they all united under the term (quantum) conference key agreement (Q)CKA.

CKA protocols aim to take new approaches to QKD in networks that have more than 2 peers. They want to increase speed, minimize number of keys and key exchanges and still keep security at a high standard.

7.6.1 Bipartite CKA

While we just stated peer-to-peer QKD to be inefficient, some CKA protocols actually follow the idea of simply using QKD to establish peer-to-peer connections between all nodes in the network. These, now secure channels, can be used to distribute a classical, non-quantum key, to all the participants. While this is inefficient, it was the only solution to this problem for quite some time. The only way this kind of protocols were able to achieve a slight speedup is that they are not reliant on a full mesh topology. When the protocol simply uses a dedicated host in the network to perform the QKD process with each peer once and then distributes the secure key for the CKA to them, the needed connections decrease from exponential $(N(N-1)/2)$ to just linear $(N-1)$. This is already a substantial decrease. But the host in charge needs to be capable to quickly and efficiently perform many QKD sessions in a row for this approach to reliably work.

7.6.2 Multipartite CKA

Another approach that seems promising for the future is multipartite CKA. Here, special quantum states (Such as the GHZ[5] state) are used, to perform a key agreement between N amount of peers simultaneously. So basically QKD is performed with N nodes at the same time.

There already are a few protocols that use different quantum properties to perform CKA. The one we will use as an example is an experimental implementation of CKA by using the GHZ states to negotiate keys between the peers. Leveraging existing technologies, the protocol in use is simply called NBB84, since its functionality is exactly the same as the BB84protocol, but it is performed between N peers with entangled quantum particles. This essay will only highlight the differences from the BB84protocol here, if you want to read more about how BB84works, please refer to section 7.1.

The differences of NBB84compared to standard BB84are as follows:

- The photon source does not sit at a participant's end, but on a shared "quantum server".
- The protocol is triggered by a request from one of the participants to this server.
- The server then distributes the key via entangled GHZ states to all participants at the same time.
- They derive the keys and then communicate their misses to each other.
- Now everybody knows which bits can be used in the key generation.

[5]The GHZ state is a special quantum state that entangles quantum particles in a way that all entangled quantum particles have the same outcome, should they be measured with the same polarization filter.

[89, p. 3]

After this is completed, all participants of this round of the protocol are in possession of a key, which they now can use to safely communicate with each other. The protocol successfully authenticated a full network without the need for peer-to-peer QKD which makes it more lightweight and thus faster. This technology will be important in medium scale quantum network that contain N nodes, for example within metropolitan areas in which many participants want to talk to each other, while classical peer-to-peer QKD remains useful for the interconnection of these medium scale network with each other.

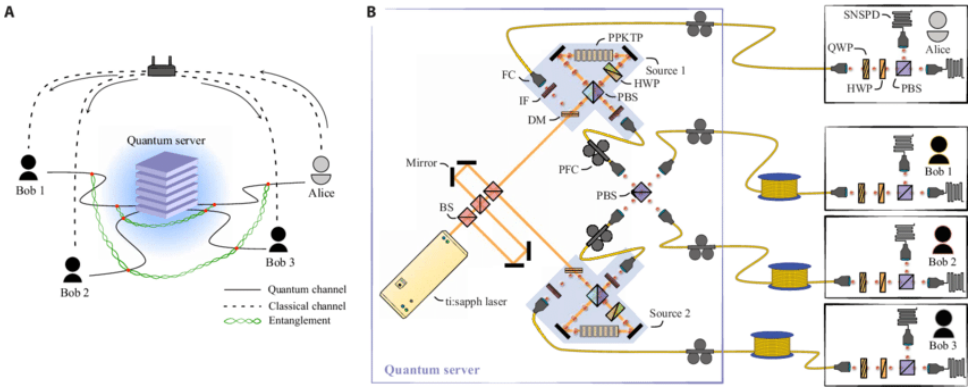

Figure 7.4: A typical setup for CKA with the use of NBB84[89, p. 2]

7.7 Conclusion

Disclaimer

Everything included in this chapter is merely an opinion by the authors and is not supported by hard evidence or direct quotations of relevant literature. The opinion is based on the research documented in the previous chapter.

The previous chapter provided an in depth look at the current state of QKD Technology is steadily advancing as time's arrow marches forward, but what exactly can this technology provide in the regard of PQC for current businesses?

As you can see in the Table 7.2, QKD does not seem to hold up to its promises of being a solution that can be ready in time to act as a serious contender for PQC That is mainly due to its drawbacks, which keep it from being an easy adaptable solution for the masses. QKD is more like a highly technological solution that has its niche applications in certain areas of cryptography where it's amazing properties can be fully utilized. While a lot of the technological advancements discussed in the previous chapters seem to have the potential to make it into a usable solution for **some** use cases, it simply cannot achieve the daunting requirement of being usable by the masses, and thus fails as PQC solution for the near future. The main reason for its shortcoming are its **costs**, and its **distance limitations**. This combined with increasing unreliability over distance and being prone to an easy denial of service, simply makes this form of cryptography unattractive for global connections that need to be reliable and cheap.

The technology is undoubtedly one of, if not the most secure, way of exchanging keys over an insecure network. But even the best technology sees no widespread use if its drawbacks hold it back. QKD can almost be compared to the Concorde [6]. It's a big leap in technology, undoubtedly the most sophisticated and fastest in its field, but its costs and unreliability simply hold it back and prevent it from achieving its full full potential.

But, the scientific community is working on the shortcomings of QKD While some call this the turning point for QKD others are skeptical that this technology can be improved to a level in which it would be suitable for the masses. Solutions to distance problems, like satellite transmissions and twin field encoding would greatly improve QKD but it would also increase the costs further. There are other approaches, such as using new protocols, to utilize the existing infrastructure to a better extent, such as CKA. But even these ambitions fall short when the underlying technology isn't ready for the consumer market in it's current state. So while QKD was trying to become more accessible, and thus more widespread in it's use, it seems to look like it's going back towards its roots. Being a highly sophisticated solution for a few niche applications, such as highly sensitive data transmissions and scientific applications such as clock synchronization and telescope intercommunication. For them, the possibility of a quantum internet sure is a great scientific endeavour that might materialize itself in the coming years or decades, to advance technology even further.

But even in its most specific field of work, QKD seems to be slowed down. The transmission of highly sensitive data for governments and global agencies. Different government organizations have announced that the use of QKD is not ideal for their use cases. While NCSC (National Cyber Security Centre[7]) has recanted their statement and has allowed the use of QKD again [90], the NSA continues to keep their stance that QKD is not the way to go. They state on their webpage:

> In summary, NSA views quantum-resistant (or post-quantum) cryptography as a more cost effective and easily maintained solution than quantum key distribution. For all of these reasons, NSA does not support the usage of QKD or QC to protect communications in National Security Systems, and does not anticipate certifying or approving any QKD or QC security products for usage by NSS customers unless these limitations are overcome. [91]

This is a potential deathblow to the commercial potential of QKD in the United States. If a technology is not endorsed by the NSA it will be tough to justify any investments in pursuing this as a PQC solution. While the US is not the only country on this planet to develop IT solutions, it's a crucial market that could fade away due to this. This would make QKD less viable for vendors, meaning the technological advances could slow down. But other nations are currently working on projects to further develop QKD especially the UK, the EU and China. It remains to be seen what can be achieved in the near future, and maybe the solution will become viable after all, but right now, it isn't.

[6] A highly sophisticated supersonic passenger airliner from 1969, ultimately brought down by its flaws and high cost.Wikipedia

[7] The NSCS of the united Kingdom is their official national cyber security guidance institution.

Audience	+ advantages – disadvantages
General comments	+ QKD is provable secure + Thus, impeccable against eavesdroppers – High infrastructure costs – Has severe distance limitations – Lower reliability compared to other PQC
Users	+ QKD Providers could make this technology feasible for very secure key agreement – QKD in general is not viable for the consumer at the current stage
Small/Medium businesses	+ Could be bought from a provider as a service for secure key agreement with certain government institutions – No real use for QKD internally
Global enterprises	+ Could be a great solution for key exchange between locations across the globe + Possibility for small scale quantum infrastructure between viable facilitys in close vicinity – Operating infrastructure across the globe is unviable from a financial standpoint – So would still be reliant on providers, which need to be trusted
Governments	+ Could operate nationwide QKD networks and provide secure key exchange for highly sensitive data + Could license service providers to build quantum networks for use by public for QKD needs – Nationwide infrastructure is highly expensive – Needed collaboration with other nations for global network – Global network currently only achievable with satellites, making nations dependent on contractors like SpaceX or Boeing

Table 7.2: QKD conclusion overview

8 Conclusion

Disclaimer

This chapter reflects the personal opinion of the writers. It is based on all previously mentioned research.

We touched on many things in this essay. Be it new technological developments, improvements of existing technologies, or already existing solutions, repurposed for their new use. We painted a vibrant picture of many technologies which will all play their assigned part in the complex dance that will be post quantum cryptography. But what do we take away from all of this?

Quantum computers have made a huge leap in the recent years. While they seemed impossible for many decades, it was finally accomplished to build reliable quantum circuits which are more than just a lab experiment. They work, and they provide real value. It is certain, that this technology will grow in the future and become more and more powerful. And while this brings exciting new possibilities in countless areas of research and technology with it, it also poses a threat for our current cryptography. A threat like we never faced it before.

But we are not as unprepared as some might think. The cryptographic community has already spend nearly two decades, debating, collaborating and researching for possible solutions that can help us to reshape cryptography, and to prepare it for a post quantum age. This cannot be undertaken without any guidance, and while many smaller committees have longed this task, the NIST has taken the torch and united all other institutions and researchers behind them. They have started a remarkable process, in which they aim to validate the cryptography of tomorrow, step by step, line by line, and even bit by bit. They are currently concluding the third round of this process, and we are certain that the end of this round will bring a first standard to light. The first ever standard which will include at least one KEM and one signature algorithm, which are quantum proof and ready to be put to good use. Which algorithm they will choose is incredibly hard to predict. But when factoring in all the different aspects it seems likely that the "Crystal" family has a very strong position, due to their great strength, adaptability and performance. Furthermore it is also likely that picnic and saber could be standardized. We think that it is unlikely that NTRU or Classical McEliece will be standardized, due to their poorer performance and increased key sizes. But our guess is as good as any.

Hardware security modules are an interesting topic that seems to gain traction in the coming decade. Post quantum cryptography brings new challenges with it, and hardware security modules seem a good fit to tackle some of them. These specifically are

scenarios of high throughput and low latency, such as payment operations. But also scenarios on which current architectures need adaptation to the newer cryptographic algorithms, without a big overhaul of the whole architecture. We believe that hardware security modules can play a key role in the transition, even in combination with hybrid certification processes. And they have great characteristics to come away from their current reputation, as being too rigid, and moving towards a future in which they will be one of the most agile parts of a cryptographic architecture in an enterprise. Thanks to new developments with state of the art technologies.

Sadly the same thing cannot be said about QKD. While it is a highly interesting technology, we do not believe it will have wide spread use, at least for the coming years. The technology is simply too complex and brittle to be used in everyday use cases. But we think, thank QKD can truly shine when put in the right spot. QKD is an amazing technology, when used in areas where it shines. That is provable secure key generation over short distances for highly confidentail data, such as election results, trade secrets and data of national interest. QKD has a future, but not as a contender for post quantum cryptography.

A alternative to QKD are hybrid or composite certificates for a faster transition to new algorithms. The presented designs enable multiple algorithms and high backwards-combability for legacy systems. It enables PKI systems to serve new algorithms to applications and avoids the installation of a second PKI infrastructure. Our research showed that combining hybrid or composite certificates with FPGA makes the transition to new algorithms faster.

List of Figures

List of Tables

Listings

Bibliography

[1] Hugh Collins, “Ibm unveils breakthrough 127-qubit quantum processor,” Nov 16, 2021. [Online]. Available: https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor

[2] D. Moody, G. Alagic, D. C. Apon, D. A. Cooper, Q. H. Dang, J. M. Kelsey, Y.-K. Liu, C. A. Miller, R. C. Peralta, R. A. Perlner, A. Y. Robinson, D. C. Smith-Tone, and J. Alperin-Sheriff, “Status report on the second round of the nist post-quantum cryptography standardization process,” Gaithersburg, MD.

[3] Wikiedpia, “Quantum computing,” 2021.

[4] “quantum-vs-classical-bit-pinson-1,” July 16, 2020.

[5] Jay Gambetta, “Ibm’s roadmap for scaling quantum technology,” 2020.

[6] “Quantum computers may be destroyed by high-energy particles from space,” 26 August 2020.

[7] Greg Nichols, “Ibm’s goldeneye: Behind the scenes at the world’s largest dilution refrigerator,” February 2, 2021.

[8] J.-P. Aumasson and M. D. Green, *Serious cryptography: A practical introduction to modern encryption*. San Francisco: No Starch Press, 2018.

[9] Andreas Baumhof, “Are quantum computers really a threat to cryptography?” 2019.

[10] D. J. Bernstein, J. Buchmann, and E. Dahmén, *Post-quantum cryptography*. Berlin: Springer, 2009.

[11] PQCrypto, “Post-quantum cryptography: Conferences,” 2021.

[12] Wikipedia, “Post-quanten-kryptographie,” 2021.

[13] Viet Ba Dang, Farnoud Farahmand, Michal Andrzejczak, Kamyar Mohajerani, Duc Tri Nguyen, and Kris Gaj, “Implementation and benchmarking of round 2 candidates in the nist port-quantum cryptography standardization process using hardware and software/hardware co-design approaches.”

[14] Moody, Dustin, “The 2nd round of the nist pqc standardization process,” June 7, 2021.

[15] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone, “Status report on the first round of the nist post-quantum cryptography standardization process,” Gaithersburg, MD.

[16] D. Moody, “Nist status update on the 3rd round,” June 7, 2021.

[17] “Candidates updates for kem’s,” June 7, 2021.

[18] D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds., *Post-Quantum Cryptography*, 1st ed., ser. Lecture notes in computer science Post-quantum cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg and Imprint: Springer, 2009.

[19] “Example “shortest vector problem” lattice problem from - closest vector problem lattice clipart.”

[20] “Candidate updates for signatures,” June 7, 2021.

[21] Ding Jintai and Petzoldt Albrecht, “Current state of multivariate cryptography.”

[22] “Sharcs ’09: Special-purpose hardware for attacking cryptographic systems.”

[23] “Reduced instruction set computer.”

[24] “Arm cortex-m.” [Online]. Available: https://en.wikipedia.org/wiki/ARM_Cortex-M#Cortex-M4

[25] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen, “pqm4: Testing and benchmarking nist pqc on arm cortex-m4.”

[26] S. Zakrajsek, “Performance analysis of nist round 2 post-quantum cryptography public-key encryption and key-establishment algorithms on armv8 iot devices using supercop.”

[27] Kanad Basu, Deepraj Soni, Mohammed Nabeel, and Ramesh Karri, “Nist post-quantum cryptography: A hardware evaluation study.”

[28] J. Ding and J.-P. Tillich, *Post-Quantum Cryptography*. Cham: Springer International Publishing, 2020, vol. 12100.

[29] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, “Post-quantum authentication in tls 1.3: A performance study.”

[30] D. Moody, “The 2nd round of the nist pqc standardization process,” August 22, 2019.

[31] “ebacs: Ecrypt benchmarking of cryptographic systems: Measurements of key-encapsulation mechanisms, indexed by machine.” [Online]. Available: https://bench.cr.yp.to/results-kem.html

[32] “ebacs: Ecrypt benchmarking of cryptographic systems: Measurements of public-key signature systems, indexed by machine.” [Online]. Available: https://bench.cr.yp.to/results-sign.html

[33] “Saber webpage.” [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

[34] Sujoy Sinha Roy, “Saberx4: High-throughput software implementationof saber key encapsulation mechanism,” 12 Nov 2019.

[35] Cong Chen, Oussama Danba, Jerey Hostein, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, and Zhenfei Zhang, “Ntru: Algorithm specications and supporting documentation.”

[36] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé, “Crystals-kyber: Algorithm specifications and supporting documentation.”

[37] Martin R., Tung Chou, Carlos Cid, Tanja Lange, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Edoardo Persichetti, Peter Schwabe, Jakub Szefer, Cen Jung Tjhai, and Martin Tomlinson, “Classic mceliece: conservative code-based cryptography.”

[38] Shi Bai, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé, “Crystals-dilithium-round 3.”

[39] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang, “Falcon: Fast-fourier lattice-based compact signatures over ntru.”

[40] “ebacs: Ecrypt benchmarking of cryptographic systems.” [Online]. Available: https://bench.cr.yp.to/primitives-sign.html

[41] Ming-shing Chen, Jintai Ding, Matthias Kannwischer, Jacques Patarin, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang, “Rainbow signature: One of the three nist post-quantum signature finalists.” [Online]. Available: https://www.pqcrainbow.org

[42] “Hybrid (biology).”

[43] “Public key certificate,” 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Public_key_certificate&oldid=1058310669

[44] “Definition of composite,” 23.12.2021. [Online]. Available: https://www.merriam-webster.com/dictionary/composite

[45] Digital Guardian, “What is public key cryptography?” 2015. [Online]. Available: https://digitalguardian.com/blog/what-public-key-cryptography

[46] “Hybrid cryptosystem,” 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hybrid_cryptosystem&oldid=1000086473

[47] SSLTrust, “Understanding certificate cross-signing | ssltrust,” 23.12.2021. [Online]. Available: https://www.ssltrust.com.au/blog/understanding-certificate-cross-signing

[48] S. Helme, “Cross-signing and alternate trust paths; how they work,” *Scott Helme*, 22.06.2020. [Online]. Available: https://scotthelme.co.uk/cross-signing-alternate-trust-paths-how-they-work/

[49] “X.509.” [Online]. Available: https://en.wikipedia.org/wiki/X.509

[50] ITU, “Introduction to asn.1,” 23.12.2021. [Online]. Available: https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx

[51] “Asn.1 made simple - introduction,” 26.05.2021. [Online]. Available: https://www.oss.com/asn1/resources/asn1-made-simple/introduction.html

[52] Eric Crockett, Christian Paquin, and Douglas Stebila, “Prototyping post-quantum and hybrid key exchange and authentication in tls and ssh.”

[53] “Hybrid key exchange in tls 1.3,” 13.07.2021. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-tls-hybrid-design-03.html

[54] Mike Ounsworth and Massimiliano Pala, “Composite public and private keys for use in internet pki.”

[55] ——, “Composite signatures for use in internet pki.” [Online]. Available: https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/

[56] H. Roßnagel, C. H. Schunck, and S. Mödersheim, “How quantum computers threat security of pkis and thus eids,” Open Identity Summit 2021.

[57] Data Encryption Standard.

[58] Advanced Encryption Standard.

[59] What is Crypto-Agility? [Online]. Available: https://www.cryptomathic.com/news-events/blog/what-is-crypto-agility

[60] The Crypto-Agility Properties, “Ha536vg.”

[61] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet x.509 public key infrastructure certiticate and certificate revocation list (crl) profile.”

[62] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey, and J. A. Halderman, “The matter of heartbleed,” in *IMC ’14 : proceedings of the 2014 ACM Internet Measurement Conference : November 5-7, 2014, Vancouver, BC, Canada*, C. Williamson, Ed. Place of publication not identified: ACM, 2014, pp. 475–488.

[63] C. M. Maxfield, “Fpga vs. asic designs,” in *Maxfield (Hg.) 2008 - FPGAs*, pp. 61–73.

[64] E. Barker and A. Roginsky, “Transitioning the use of cryptographic algorithms and key lengths,” Gaithersburg, MD.

[65] D. C. William Barker, “Migration to post-quantum cryptography.”

[66] Michael Suby, “An anchor of trust in a digital world: Risk management strategies for digital processes.”

[67] “Managing cryptographic and quantum risk: A non-technical and hype-free explanation of what’s at risk, what you can do, and why you should act now.”

[68] “The picnic signature scheme.”

[69] “Mtg post-quantum cryptography,” 2021.

[70] “Hsm security - cryptographic key management - qxedge,” 2021. [Online]. Available: https://crypto4a.com/products/hybrid-security-platform/

[71] “Post-quantum cryptography: Faqs,” January 03, 2017.

[72] “Security requirements for cryptographic modules,” Gaithersburg, MD.

[73] “Using thales luna hsms with quantum-safe security to protect iot,” May 2020.

[74] “Quantencomputer: Eine bedrohung für pki,” 2021.

[75] Simon Piff, “Quantum-safe crypto key management: Why now!”

[76] “Apply post-quantum cryptography today with utimaco q-safe 1.0,” 2021. [Online]. Available: https://hsm.utimaco.com/products-hardware-security-modules/general-purpose-hsm/quantum-safe/

[77] “Post-quantum cryptography vpn - microsoft research,” 2021. [Online]. Available: https://www.microsoft.com/en-us/research/project/post-quantum-crypto-vpn/

[78] “Post-quantum cryptography: Secure encryption for the quantum age.”

[79] F. Grasselli, *Quantum Cryptography*. Cham: Springer International Publishing, 2021.

[80] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, “Practical challenges in quantum key distribution.”

[81] N. Sangouard, C. Simon, H. d. Riedmatten, and N. Gisin, “Quantum repeaters based on atomic ensembles and linear optics,” *Rev. Mod. Phys; 83*, 2011. [Online]. Available: http://arxiv.org/pdf/0906.2699v2

[82] RACHEL COURTLAND, “China’s 2,000-km quantum link is almost complete,” *IEEE Spectrum*, 26 Oct 2016.

[83] M. Lucamarini, Z. Yuan, J. F. Dynes, and A. J. Shields, “Overcoming the rate-distance barrier of quantum key distribution without using quantum repeaters,” *Nature*, vol. 557, no. 7705, pp. 400–403, 2018. [Online]. Available: http://arxiv.org/pdf/1811.06826v1

[84] J.-P. Chen, C. Zhang, Y. Liu, C. Jiang, W.-J. Zhang, Z.-Y. Han, S.-Z. Ma, X.-L. Hu, Y.-H. Li, H. Liu, F. Zhou, H.-F. Jiang, T.-Y. Chen, H. Li, L.-X. You, Z. Wang, X.-B. Wang, Q. Zhang, and J.-W. Pan, “Twin-field quantum key distribution over 511 km optical fiber linking two distant metropolitans.”

[85] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, F.-Z. Li, X.-W. Chen, L.-H. Sun, J.-J. Jia, J.-C. Wu, X.-J. Jiang, J.-F. Wang, Y.-M. Huang, Q. Wang, Y.-L. Zhou, L. Deng, T. Xi, L. Ma, T. Hu, Q. Zhang, Y.-A. Chen, N.-L. Liu, X.-B. Wang, Z.-C. Zhu, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan, “Satellite-to-ground quantum key distribution,” *Nature*, vol. 549, no. 7670, pp. 43–47, 2017. [Online]. Available: http://arxiv.org/pdf/1707.00542v1

[86] Markus Payer, “Esa und ses-geführtes konsortium entwickeln satellitenbasiertes cyber-sicherheitssystem,” 03 May 2018.

[87] ——, “Ses announces 10 project partners in quartz satellite cybersecurity consortium,” 07 Jun 2018.

[88] Tereza Pultarova, “U.k. company to start sending secret quantum keys with satellites in 2023,” *space.com*, June 23, 2021. [Online]. Available: https://www.space.com/arqit-quantum-key-distribution-space

[89] M. Proietti, J. Ho, F. Grasselli, P. Barrow, M. Malik, and A. Fedrizzi, “Experimental quantum conference key agreement,” *Science Advances*, vol. 7, no. 23, p. eabe0395, 2021. [Online]. Available: http://arxiv.org/pdf/2002.01491v2

[90] Tom De Nert, “Community response to the ncsc 2020 quantum security technologies white paper,” 27th May 2020.

[91] “Quantum key distribution (qkd): and quantum cryptography (qc)quantum cryptography qc,” 2021.

Glossary

CA A certificate authority (CA) is a trusted entity that issues certificates 47, 49

CLR In cryptography, a certificate revocation list is a list of digital certificates that have been revoked by the issuing certificate authority (CA) before their scheduled expiration date and should no longer be trusted. 59

SHA-1 In cryptography, Secure Hash Algorithm 1 is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest 48

X.509 An X.509 certificate binds an identity to a public key using a digital signature . . 44, 47, 66

Acronyms

AES Advanced Encryption Algorithm 18, 24, 25, 26, 34, 69, 84

ASN.1 Abstract Syntax Notation One 36, 66

BB84 QKD scheme developed by Charles Bennet and Gilles Brassard in 1984 . . 82, 83, 84, 85, 96, 102

CC Common Criteria, a specification for security requirements. 25, 69, 78

CKA Conference Key Agreement 95, 96, 97, 102

CVP Closes Vector Problem 21

DNS Domain Name System 48

eBACS ECRYPT Benchmarking of Cryptographic Systems 25

ECDSA Elliptic Curve Digital Signature Algorithm 46, 104

ESA European Space Agency 94

FIPS Federal Information Processing Standards, a specification for security requirements. 71, 74, 78, 79

FPGA Field Programmable Gate Array 70

GHZ Greenberger-Horne-Zeilinger state 96

GPS Global Positioning System 93

HSM Hardware Security Module, purpose built security device. . . . 73, 76, 77, 78, 79, 80, 81

IBM International Business Machines Corporation, a US It company. 13, 14, 102

IP Internet Protocol 48

LAN Local Area Network 73

LDAP Lightweight Directory Access Protocol 35

MQ Multivariate Quadratic Equations 22