



**CZECH TECHNICAL  
UNIVERSITY  
IN PRAGUE**

**F3**

**Faculty of Electrical Engineering  
Department of Computer Science**

**Master's Thesis**

# **Securing Quantum Key Distribution: Architecture, Threats, and Deployment Strategies**

**Sebastian Štefko**  
Cybersecurity

**May 2024**

**Supervisor: doc. Ing. Leoš Boháč, Ph.D.**



## I. Personal and study details

Student's name: **Štefko Sebastian**

Personal ID number: **492294**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Cyber Security**

## II. Master's thesis details

Master's thesis title in English:

**Securing Quantum Key Distribution: Architecture, Threats, and Deployment Strategies**

Master's thesis title in Czech:

**Architektura systému pro zabezpečení kvantového přenosu klíče: hrozby a strategie nasazení**

Guidelines:

The thesis will focus on the creation and security of quantum key distribution (QKD) systems, emphasizing the development of an optimized architecture for an efficient and secure QKD system. It will explore the theoretical and practical aspects of security, identify key security elements, analyze potential vulnerabilities and attack vectors, and create a methodology for the development, deployment, and operation of QKD systems. The work will combine a literature review, analysis of real-world cases, and practical experiments, providing a comprehensive view of the development, security, and deployment in the field of QKD, which is essential for progress in quantum communication and security.

Outputs of the thesis:

- Design and description of an optimized QKD system architecture.
- Detailed analysis of security threats and vulnerabilities.
- Developed methodology for the development, deployment, and operation of QKD systems.
- Suggestions for improving existing security practices in the field of QKD.

Bibliography / sources:

- [1] H. Wang, Y. Zhao, A. Nag, X. Yu, X. He and J. Zhang, "End-to-End Quantum Key Distribution (QKD) from Metro to Access Networks," 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, Milan, Italy, 2020, pp. 1-5,
- [2] P. Burdiak, L. Kapišák, L. Michalek, E. Dervisevic, M. Mehic and M. Vozák, "Demonstration of QKD Integration into 5G Campus Network," 2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2023, pp. 1-4,
- [3] Y. Zhao, Y. Cao, X. Yu and J. Zhang, "Software defined optical networks secured by quantum key distribution (QKD)," 2017 IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, China, 2017, pp. 1-4, doi: 10.1109/ICCCChina.2017.8330367.

Name and workplace of master's thesis supervisor:

**doc. Ing. Leoš Boháč, Ph.D. Department of Telecommunications Engineering FEE**

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **09.02.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

\_\_\_\_\_  
doc. Ing. Leoš Boháč, Ph.D.  
Supervisor's signature

\_\_\_\_\_  
Head of department's signature

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
Dean's signature

### III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

\_\_\_\_\_  
Date of assignment receipt

\_\_\_\_\_  
Student's signature

## Acknowledgement / Declaration

I would like to thank my supervisor, doc. Ing. Leoš Boháč, Ph.D., for his valuable suggestions and comments during the guidance of this thesis. I would also like to acknowledge Ing. Jiří Weiss for his advice during the consultations. Lastly, I would like to express my gratitude to my family, who have supported me throughout my studies.

Thank you.

I declare that I elaborated this thesis on my own and that I mentioned all the information sources that have been used in accordance with the Guideline for adhering to ethical principles in the course of elaborating an academic final thesis.

Prague, 24th May 2024

.....

## Abstrakt / Abstract

Pokroky v kvantovém počítání nás nutí zkoumat a implementovat nové metody ke klasické kryptografii s veřejným klíčem. Jednou z alternativ je kromě postkvantové kryptografie takzvaná kvantová distribuce klíče. Tato technika je spojena s budováním speciálních sítí schopných distribuovat sdílený klíč mezi libovolné dvě komunikující strany.

Nejprve vysvětlíme bezpečnostní důsledky, které kvantové počítače představují pro dnes široce používané algoritmy asymetrické kryptografie. Jako řešení, kterým se budeme dále zabývat, jsou sítě pro kvantovou distribuci klíče. Je zmíněno několik reálných implementací a následně detailně popsána architektura s důrazem na bezpečnost. Je vysvětlena podstata informační bezpečnosti a provedena bezpečnostní analýza kvantových sítí. Její součástí je navržený bezpečnostní profil pro systém pro správu klíčů v kvantových sítích, napsaný podle specifikace Common Criteria. Dále navrhuje techniky pro zvýšení bezpečnosti v oblastech: (i) doručení klíče k uživateli, (ii) posílení bezpečnosti klíče a (iii) propojení sítí. Nakonec vyvineme metodologii pro vývoj, implementaci a provozování kvantových sítí tak, aby mohla být aplikována na tyto nové sítě.

**Klíčová slova:** kvantová distribuce klíče, síť pro kvantovou distribuci klíče, QKD, bezpečnost QKDN, metodologie QKDN

Advances in the field of quantum computing are compelling us to explore and implement novel methods to classical public-key cryptography. One of the alternatives besides post-quantum cryptography is quantum key distribution (QKD). This technique involves building specialised QKD networks capable of distributing a shared secret between two parties.

We commence by illustrating the enormous security implications of quantum computing on currently used asymmetric cryptographic schemes. QKD networks (QKDN) are determined as a solution. Several real-world implementations are mentioned, and network architecture design is described in detail, emphasising security. We explain the essence of information security and conduct a security analysis of a QKDN. We contribute to increasing the security of QKDNs by writing a protection profile for a key management system for a QKDN according to the Common Criteria specification. Next, we suggest techniques for improving the security of the following: (i) key delivery to users, (ii) strengthening of key security and (iii) interconnection of networks. We develop a methodology for creating, implementing, and managing QKDNs that can be readily applied to upcoming networks.

**Keywords:** quantum key distribution, quantum key distribution network, QKD, QKDN security, QKDN methodology

# Contents /

<b>1 Introduction</b>	<b>1</b>		
1.1 Asymmetric cryptography . . . . .	1		
1.1.1 Integer factorisation problem . . . . .	2		
1.1.2 Discrete logarithm problem . . . . .	2		
1.1.3 Applications of asymmetric cryptography . . . . .	3		
1.1.4 Shortcomings of asymmetric cryptography . . . . .	4		
1.2 Transport layer security handshake analysis . . . . .	4		
1.3 Quantum computing . . . . .	7		
1.3.1 Shor's algorithm . . . . .	8		
1.4 Solutions . . . . .	8		
1.4.1 Post-quantum cryptography . . . . .	8		
1.4.2 Quantum key distribution . . . . .	9		
<b>2 Architecture of QKD Networks</b>	<b>11</b>		
2.1 Historical review of QKDNs . . . . .	11		
2.1.1 DARPA Quantum Network . . . . .	11		
2.1.2 SECOQC . . . . .	12		
2.1.3 Tokyo QKD Network . . . . .	12		
2.2 Standardisation of QKDNs . . . . .	12		
2.3 QKDN topology . . . . .	14		
2.3.1 Point-to-point network . . . . .	14		
2.3.2 Multipoint network . . . . .	16		
2.4 Functional layers . . . . .	17		
2.4.1 Quantum layer . . . . .	17		
2.4.2 Key management layer . . . . .	18		
2.4.3 Control layer . . . . .	20		
2.4.4 Management layer . . . . .	20		
<b>3 Security Analysis</b>	<b>21</b>		
3.1 Information security . . . . .	21		
3.1.1 Objectives . . . . .	22		
3.1.2 Entities and relationships . . . . .	22		
3.2 Risk and risk management . . . . .	24		
3.3 General model . . . . .	25		
3.4 Users of the QKDN . . . . .	28		
3.5 Threat identification in QKDNs . . . . .	28		
3.6 Attack surfaces in QKDNs . . . . .	30		
3.6.1 Quantum hacking . . . . .	31		
3.7 Protection profile: Key management system for QKDN . . . . .	31		
3.7.1 PP introduction . . . . .	32		
3.7.2 Security problem definition . . . . .	32		
3.7.3 Security objectives . . . . .	34		
3.7.4 Security requirements . . . . .	35		
<b>4 Design Towards Optimised QKDN Architecture</b>	<b>41</b>		
4.1 Key delivery to cryptographic applications . . . . .	41		
4.1.1 Security profiles — application inside the security perimeter . . . . .	43		
4.1.2 Security profiles — application outside the security perimeter . . . . .	43		
4.2 Methods of enhancing key security . . . . .	45		
4.2.1 Key combination . . . . .	45		
4.2.2 Multi-path key delivery . . . . .	46		
4.3 Interworking of the QKDN networks . . . . .	47		
4.3.1 Threat identification . . . . .	48		
<b>5 Methodology for Development, Deployment and Operation of QKDNs</b>	<b>51</b>		
5.1 Development . . . . .	51		
5.2 Deployment . . . . .	53		
5.2.1 Backbone network . . . . .	54		
5.2.2 Metropolitan and access network . . . . .	55		
5.3 Operation . . . . .	57		
5.4 EuroQCI . . . . .	58		
<b>6 Conclusion</b>	<b>59</b>		
<b>References</b>	<b>61</b>		
<b>A Abbreviations</b>	<b>69</b>		

## Tables / Figures

<b>1.1</b> TLS Cipher Suites .....	6	<b>1.1</b> TLS Handshake .....	5
<b>1.2</b> NIST selected PQC algorithms ..	9	<b>2.1</b> SECOQC QKDN Topology....	12
<b>3.1</b> Security objective rationale ....	35	<b>2.2</b> Tokyo QKDN Topology.....	13
<b>3.2</b> Security functional require- ment rationale.....	40	<b>2.3</b> Point-to-point QKDN topol- ogy .....	15
		<b>2.4</b> Multipoint QKDN topology ...	16
		<b>2.5</b> Key management functional architecture.....	18
		<b>2.6</b> Key relaying .....	19
		<b>3.1</b> Security relationships dia- gram .....	23
		<b>3.2</b> Protection Profile relation- ships diagram.....	26
		<b>3.3</b> Common Criteria relation- ships diagram.....	27
		<b>3.4</b> Attack surfaces in QKDN .....	30
		<b>3.5</b> TOE identification .....	31
		<b>4.1</b> The idealised QKDN.....	42
		<b>4.2</b> Cryptographic application location .....	43
		<b>4.3</b> Key combination .....	45
		<b>4.4</b> Scheme of interworking node ..	49
		<b>5.1</b> Key rate on distance depen- dency .....	54
		<b>5.2</b> Metro and access network .....	56



# Chapter 1

## Introduction

Secure communication over public telecommunication and data networks is not just a standard but a necessity in our modern world. As these networks become a part of our day-to-day lives, it is almost impossible to imagine a situation without being able to, for example, access Internet banking and perform banking operations or securely communicate with other people. Although security was not a primary concern in the early days of network communication between computer systems, almost all network connections are now encrypted. Such a trend is evidenced by the transition to secure alternatives of protocols used on the Internet or vast usage of virtual private networks. Algorithms and methods for achieving secure communication typically rely on distributing a shared secret between communicating parties if a pre-shared key is not present and then using this secret to create encryption keys, which are used in symmetrical encryption algorithms. The distribution of a shared secret is one of the domains of asymmetric cryptography.

In this thesis, we will briefly analyse the current situation in the field of asymmetric cryptography, specifically the quantum non-resistant algorithms, and explain its shortcomings for future use. Then, we will describe an alternative method for distributing a shared secret using the principles of quantum physics called the Quantum key distribution (QKD). Because this method requires constructing a complex infrastructure, a quantum key distribution network (QKDN), we will describe its architecture in detail, review current standards for the optimised network and provide a security analysis of the network and the key management system. Lastly, we will propose a methodology for implementing and operating the QKDN.

### 1.1 Asymmetric cryptography

Asymmetric cryptography, also known as public-key cryptography, is characterised by generating a pair of keys, one of which is typically used for encrypting and the other for decrypting. Theoretically, it does not matter in which order the keys are used. However, a message encrypted by one key can be decrypted only by the other. This allows us to safely publish one of the keys (public key) and keep the other private (private key). These keys are linked due to the mathematical problem a particular cryptographic algorithm relies on. Such problems are computationally very difficult to solve without additional knowledge, which only the communicating sides have or can arbitrarily choose.

In this section, we are concerned about algorithms widely used today, which are based on the integer factorisation problem (IFP) or the discrete logarithm problem (DLP).

Asymmetric cryptography can be used in three different areas. First, it is employed for asymmetric encryption, enabling the classical encryption and decryption of whole messages; second, for a digital signature; and lastly, for a key exchange, which will concern us the most. Not all public-key cryptosystems can realise all three areas mentioned above at once. From this point of view, an RSA [1] algorithm based on the IFP is the most universal one. On the other hand, DLP-based algorithms are tailored for a specific area. The ElGamal [2] encryption system falls under the category of asymmetric encryption. Another system, called DSA and its variant on elliptic curve (ECDSA), is used for digital signatures. Lastly, the Diffie–Hellman [3] key exchange algorithm is used for exchanging a secret over a public channel. This system also has a variant on the elliptic curve.

### ■ 1.1.1 Integer factorisation problem

According to the fundamental theorem of arithmetic, any positive integer greater than 1 is either a prime number or can be represented as a product of prime numbers. The representation is unique. The IFP is based on the search for such a product of primes. For classical computers, there is no known efficient algorithm for factorisation that would find a solution in a polynomial time for all integers. Nor was it proven that such an algorithm does not exist. The decision problem of factoring is defined as: Given numbers  $N, L, U$  decide whether  $N$  has a factor  $M$ , such that  $L \leq M \leq U$  [4]. It is believed that the decision problem of factoring is not in the **P** nor the **NP**-complete class [4]; thus, it resides somewhere in **NP**, more precisely in  $\mathbf{NP} \cap \mathbf{coNP}$ . That is because verifying a given factorisation for a number in polynomial time is possible. So far, the best algorithm in terms of complexity for classical computers is the general number field sieve [5], which runs in sub-exponential time. Cryptosystems relying on the ILP are, for example, the following:

- RSA
- Rabin Cryptosystem
- Blum–Goldwasser Cryptosystem

### ■ 1.1.2 Discrete logarithm problem

The discrete logarithm problem is another computationally difficult problem some cryptosystems rely on, formulated as follows. Let  $G$  be a cyclic group of order  $n$  generated by  $a$  ( $G = \langle a \rangle$ ). Every element  $b \in G$  can be written as  $a^x = b, x \in \mathbb{Z}_n$ . The DLP is the problem of finding  $x$ , written out as  $\text{dlog}_a b = x$ . The number  $x$  is called the discrete logarithm of  $b$  to the base  $a$  [6]. Regarding the computational complexity of the DLP, no known polynomial algorithm for classical computers is known. There is, however, also a sub-exponential algorithm for DLP [7]. Cryptosystems relying on the DLP are, for example, the following:

- ElGamal encryption system
- Digital Signature Algorithm
- Edwards-curve Digital Signature Algorithm
- Elliptic-curve Digital Signature Algorithm
- Diffie–Hellman key exchange
- Elliptic-curve Diffie–Hellman
- Schnorr Signature

### ■ 1.1.3 Applications of asymmetric cryptography

To better understand the immense benefits of asymmetric cryptography, a list of applications and real-world use cases is provided. Although not a comprehensive enumeration, it aims to raise awareness about the direct consequences of breaking cryptosystems listed in the previous paragraphs. The applications are, for example:

1. Secure communication protocols
  - SSL/TLS for secure web communication (HTTPS)
  - SSH for accessing remote shell
  - IPSec for securing IP communication
2. Virtual private networks
  - OpenVPN
  - WireGuard
3. Secure file transfer
  - FTPS and SFTP
  - PGP and GPG for encrypting and signing of messages
4. Secure email communication
  - S/MIME
5. Secure Voice over IP
6. Digital signatures and certificates
  - Public Key Infrastructure for establishing a hierarchy of trust
7. Secure Operating Systems
  - Secure boot
  - Software/driver signing
8. Blockchain
9. Secure Transactions and Payments
10. Digital Rights Management
11. DNSSEC

## 1.1.4 Shortcomings of asymmetric cryptography

The difficulty of the mathematical problems on which current asymmetric cryptography stands is always relative to the size of the problem and the available computational power. To successfully break RSA, factorising a 2048-bit long number, which is currently the most commonly used modulus, would be necessary. So far, the longest number publicly known to be factorised is an 829-bit long modulus.<sup>1</sup> With current computational power, it would be unfeasible to try to break RSA. However, it is hard to predict how the situation will be in the future in terms of classical computing. Also, there is still the possibility of a polynomial time algorithm being discovered.

Encrypting data using public key cryptography is computationally very demanding. Therefore, it is used primarily to distribute the encryption keys for symmetrical cryptosystems. The key exchange happens at the beginning of communication when a secure channel is established. So, all security falls to the public key algorithm. If someone intercepted the initial communication with the encrypted data stream, they could try to brute-force find the decryption keys. That would allow the attacker to obtain symmetrical encryption keys and decipher all the stored communication. Such an attack is known as a *store now, decrypt later*. It is a genuine concern for subjects who wish to keep the communication confidential for an extensive time.

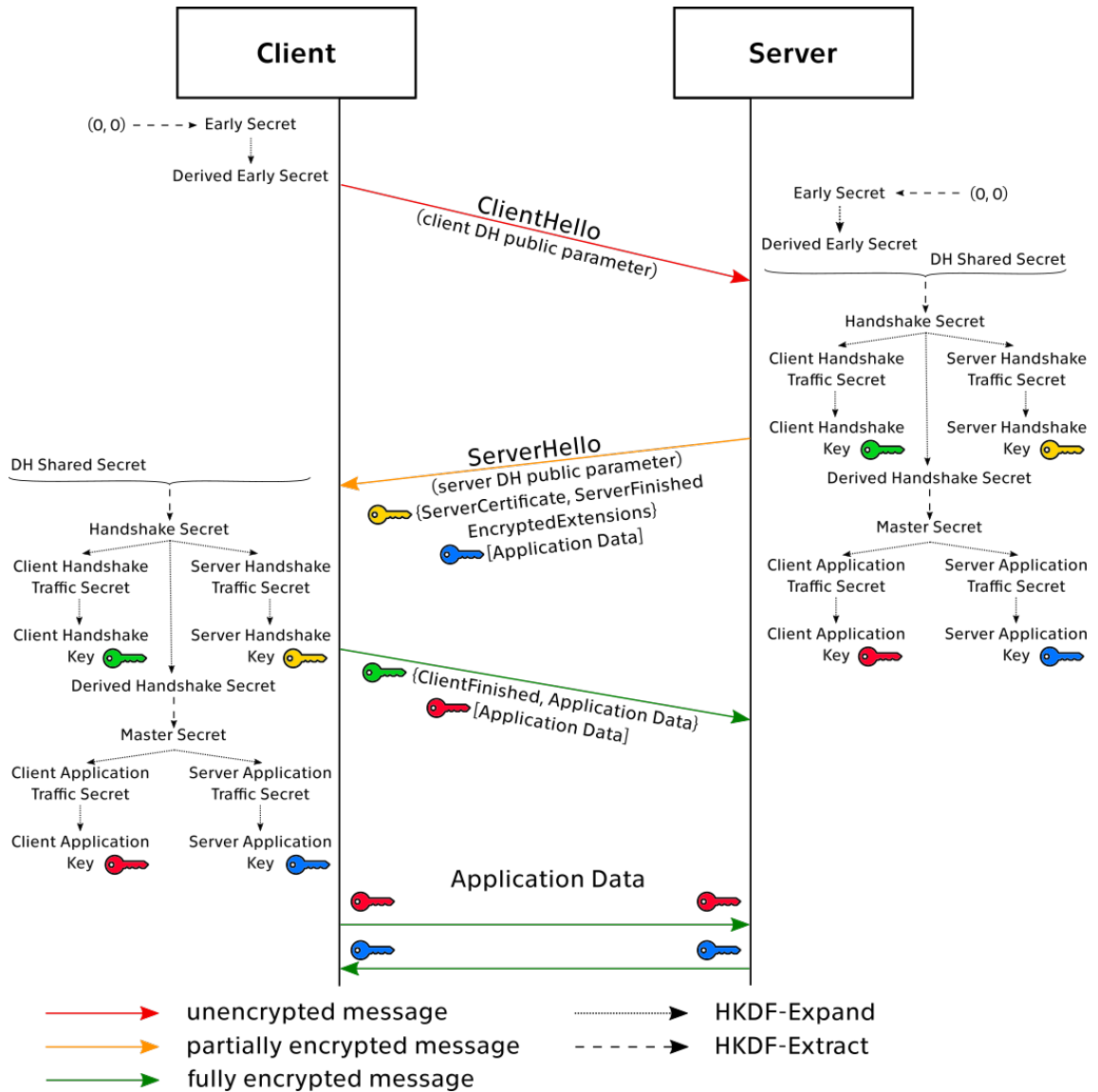
Until now, we have been only concerned with algorithms used mainly today in the context of classical computers. The most significant threat to them lies in quantum computing, as described in Section 1.3. Together with the *store now, decrypt later*, and quantum computing, these pose deficiencies that must be tackled. Possible solutions are provided in Section 1.4.

## 1.2 Transport layer security handshake analysis

**Transport layer security** (TLS) is a protocol which provides secure communication over a computer network. It **is the most widely used cryptographic protocol**. The primary purpose it serves is to provide confidentiality, integrity and authenticity. In order to achieve the objectives mentioned above, TLS consists of various algorithms that form cryptographic sets. The set defines concrete cryptographic algorithms for a session and, among other things, also a way for the key exchange, after which both sides are in possession of a shared secret.

When a new TLS session is to be established, a few messages have to be exchanged, and this is called the handshake. The handshake will be analysed to show the critical part for the *store now, decrypt later* attack. TLS has been evolving since its first version, introduced in 1999 as an upgrade of the Secure Sockets Layer (SSL). The current versions that are in use are TLS 1.2 [8] and TLS 1.3 [9]. Older versions are deprecated and should not be used any more due to the security vulnerabilities and the lack of modern cipher suites [10].

<sup>1</sup> <https://members.loria.fr/PZimmermann/records/factor.html>



**Figure 1.1.** Simplified one round-trip time TLS 1.3 handshake using only Diffie–Hellman for key exchange (no pre-shared key), showing messages exchanged between the client and the server. Message names in the parenthesis indicate that they are unencrypted. Messages listed in the curly brackets are encrypted with the appropriate Client/Server Handshake Keys, and those in square brackets are encrypted using the Client/Server Application Keys. The implementation of the HMAC-based Key Derivation Functions (HKDF) Expand and Extract is described in [11].

There are substantial differences between the two current versions, so the newer 1.3 will be considered in the analysis.

As mentioned, our attention will be focused on the handshake, more precisely, on the key exchange part. Other parts that follow the key exchange are setting up server parameters and the authentication of the server and, optionally, of the client. There are more options for how the handshake can be accomplished. It depends on factors such as whether a pre-shared key has been established

between the client and the server, whether it is the resumption of a session or an opening of a subsequent session. The situation depicted in Figure 1.1 is the most straightforward situation when only one round trip has to be made to establish application encryption keys for a TLS session.

Once a TCP session is established, the TLS handshake will start. A client generates parameters (ephemeral, short-term public and private keys) for a Diffie–Hellman (DH) key exchange and sends the public part in a ClientHello message to the server. The server also generates parameters and combines the private and the client’s public parts. This yields the DH Shared Secret, which is used along with the Derived Early Secret to derive all encryption keys. The derivation process is depicted in Figure 1.1, and it utilises the HMAC-based Key Derivation Functions [11]. The HKDF Expand function combines previously computed secrets with the hash of messages exchanged between the client and the server. In case the integrity of these messages is infringed upon, each party will derive different encryption keys and secure communication will not be possible. For a detailed description of inputs for these functions, refer to the [12].

The server eventually holds the Client/Server Handshake Key used to encrypt part of the ServerHello message sent to the client and the Client/Server Application Keys for encrypting the application data. In the TLS handshake, the ServerHello message (as depicted by the yellow arrow in Figure 1.1) is only partially encrypted. That is because when the client receives this message, it does not yet possess the Server Handshake Key. Therefore, the server’s DH parameter must be unencrypted so the client can also compute the DH Shared Secret. Only then can the client perform the same derivation process as the server. After the successful handshake, both parties will have application encryption keys. These are depicted as the red and blue keys in Figure 1.1, as each direction has a separate pair. For the encryption and ensuring the integrity of application data, algorithms from the negotiated cipher suite are used. The list of supported cipher suites in TLS 1.3 is given in Table 1.1.

AEAD algorithm	Hash function
AES-128-GCM	SHA256
AES-256-GCM	SHA384
ChaCha20-Poly1305	SHA256
AES-128-CCM	SHA256
AES-128-CCM-8	SHA256

**Table 1.1.** List of TLS 1.3 supported cipher suites.

To summarise, the attacker must do the following to recover the symmetrical Client/Server Application Keys in order to decipher the application data:

1. They must store both the ClientHello and the ServerHello messages exchanged during the TLS handshake.
2. Extract the Diffie–Hellman key exchange parameters from the messages and compute either the client’s or server’s discrete logarithm of the public part.

3. Compute the DH shared secret.
4. Derive all intermediate secrets by combining hashes of the handshake messages and the DH shared secret, from which the application encryption keys are obtained.

## 1.3 Quantum computing

American physicist Richard Feynman significantly contributed to the field of quantum electrodynamics, for which he received a Nobel Prize in Physics in 1965. Since then, more work on the possibilities of using quantum physics in computing has been explored. In 1980, Paul Benioff published a paper [13] in which he described a quantum mechanical model of a Turing machine. Yuri Manin published a book called “Computable and Non-Computable” [14] in 1980, which also discusses the idea of quantum computing. In 1981, the First Conference on the Physics of Computation took place, where Richard Feynman spoke about simulating physics using quantum computers [15]. Mainly but not only, these contributions can be considered the start of quantum computing as a field of study. For the past forty years, developments in this area have moved from theoretical models to real working quantum computers available today.

Quantum computing is a field of computer science which deals with computing and solving problems using a quantum computer. Unlike classical computing, where the individual bits (ones and zeros) are typically represented as electrical signals, quantum computing utilises qubits, which can be realised by various physical systems with quantum mechanical properties. It is an elementary unit of quantum information. There could be two states for a qubit ( $|0\rangle$  or  $|1\rangle$ ) that correspond to classical bits (0 or 1). The difference is that a qubit is a linear combination of these states (basis vectors), often called superposition [16]. Thus, a state of a qubit is a vector in  $\mathbb{C}^2$ . Such a state is being changed in the course of the computation. Quantum computer is built from quantum circuits containing quantum gates that manipulate the quantum state of a qubit. The quantum state is measured at the end of a computation, and the result is obtained. Measurement of a qubit causes the superposition state to collapse.

There are many applications of quantum computing, more concretely in areas such as finance, physics, chemistry, biology and many others. Quantum computers allow us to solve problems in optimisation, quantum simulation, quantum search, or factorisation faster than classical computers. That is due to a feature of quantum algorithms called the quantum parallelism. We will not go into detail about how quantum parallelism works and why it is possible, but we will mention its consequences for our thesis. Such algorithms based on the quantum Fourier transform offer even an exponential speedup compared to classical computers [16]. Examples that belong to this class are the Deutsch–Jozsa algorithm [17] or, most importantly, Shor’s algorithm [18]. It is necessary to note that the speedup on quantum computers is not automatically possible with any algorithm. It is believed that the **NP**-complete problems are also *hard* for quantum computers, just like for the classical ones [19–20].

### 1.3.1 Shor's algorithm

In Section 1.1.4, we mentioned that quantum computing poses a threat to current asymmetric cryptography, as described in Section 1.1. The reason is that in 1994, American mathematician Peter Shor developed an efficient algorithm for the IFP and DLP for quantum computers [18]. It is called Shor's algorithm and runs in polynomial time on a quantum computer [21]. It converts the two mentioned problems to the problem of period finding. In the case of factoring a number  $N$ , a period  $r$  is being found, such that for a  $1 < y < N$  and  $\gcd(N, y) = 1$ , it holds that  $1 = y^r \pmod{N}$ . The quantum factoring algorithm takes asymptotically  $O((\log N)^2(\log \log N)(\log \log \log N))$  steps on a quantum computer [21]. Because of this algorithm, the cryptosystems based on the IFP or DLP are considered *quantum non-resistant* algorithms.

The current state of quantum processors differs in the number of physical qubits on the chip. The largest processors today have just over a thousand physical qubits. The problems with physical qubits are that they suffer from issues with stability and quantum decoherence [22]. Therefore, a quantum error correction has to be implemented. That effectively uses several physical qubits to produce one logical qubit. For that reason, it is difficult to predict how many physical qubits would be necessary for efficient and fast factoring of the modules currently used in the RSA. Recent papers, however, provide some estimates for factoring a 2048-bit RSA integers [23–24].

## 1.4 Solutions

For future secure communication, there is a problem that we have addressed, and it needs to be solved. Even though the danger is not imminent, it is expected that once there is a quantum computer capable of breaking currently used asymmetric cryptosystems in a reasonably short time, it will also be used for malicious purposes. Such a use may be kept secret for reasons like the store now, decrypt later attacks. Countries and businesses are aware of the risk, which is why efforts to develop a solution are growing. There are two generally two main directions that can be taken. That is either a Post-quantum cryptography (PQC) or a Quantum key distribution (QKD). It will be shown that combining both approaches is also well desired.

### 1.4.1 Post-quantum cryptography

The category of post-quantum algorithms contains algorithms based on problems that are considered resistant to attacks by classical and quantum computers. Cryptosystems relying on other problems than IFP or DLP existed way before, though they were not put into practice for several reasons. They either had impractically long keys (McEliece encryption) or were inefficient for computation. Here are some other important classes of cryptographic systems [25]:



- Hash-based cryptography
- Code-based cryptography
- Lattice-based cryptography
- Multivariate-quadratic-equations cryptography
- Secret-key cryptography

There are several requirements for the future use of post-quantum algorithms. They must be efficient, people must have confidence in them, and they must be usable. The National Institute of Standards and Technology (NIST), an American government agency, strives to standardise several PQC algorithms for the future. In 2016, NIST announced a public competition with the goal of finding new algorithms for digital signatures and key encapsulation mechanisms. After three rounds, the first candidates for standardisation were announced in 2022. These are listed in Table 1.2 with the appropriate class of problem they rely on.

Name	Purpose	Problem type
CRYSTALS–Kyber	KEM	Lattice-based
CRYSTALS–Dilithium	Digital signature	Lattice-based
FALCON	Digital signature	Lattice-based
SPHINCS+	Digital signature	Hash-based

**Table 1.2.** List of chosen PQC algorithms chosen by NIST for standardisation in 2022 [26].

Although the algorithms in Table 1.2 are probably going to be widely implemented and used, it is still good to point out that post-quantum cryptography still only relies on some mathematical problems that are considered to be hard. For example, lattice-based problems like the shortest vector problem or the closest vector problem are known to be **NP**-hard [27–28]. We only believe that the problem is complicated enough; however, its hardness is based on assumptions underlying these schemes. There might be undiscovered quantum algorithms that could easily break the security of new cryptosystems [29].

## ■ 1.4.2 Quantum key distribution

Alternative to the problem mentioned in the previous paragraph and, this thesis’s primary focus will be on quantum key distribution (QKD). It is one of the applications of quantum cryptography, which uses properties of quantum mechanics, such as the no-cloning theorem or quantum entanglement, to perform cryptographic tasks. The main goal of QKD is to exchange secrets between two parties in a secure manner so that if an eavesdropper were trying to capture and alter the exchange, their presence would be detected. That is possible because the information is encoded and transferred in a quantum state (qubits), which, if once measured, cannot be reconstructed the same by an eavesdropper with absolute certainty. The main advantage is that QKD does not rely on any mathematical problem, and as we will describe in more detail later, it allows the implementation of information-theoretic security.



# Chapter 2

## Architecture of QKD Networks

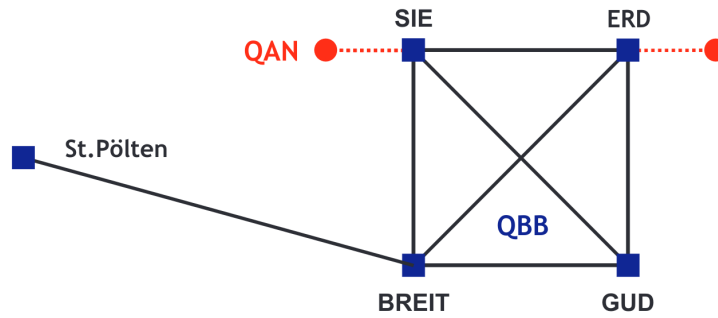
Specialised hardware and interconnections are required to realise a quantum key distribution. Together, these elements form a network with a topology similar to today's classical data networks. However, the devices in the QKD network are fundamentally different as they must implement some quantum protocol. Many other components required for operation are computers communicating in a standard, non-quantum manner. This chapter describes the architecture and topologies of a quantum key distribution network (QKND) and its functional elements sequentially, layer by layer. We also examine the history of already implemented networks that served mainly for research purposes. A brief overview of standardisation efforts in QKDN will also be given.

### 2.1 Historical review of QKDNs

The first mention of the possibility of using quantum mechanics to secure information comes from Stephen Wiesner and Charles Bennett at the turn of the '60s and '70s [30]. This idea, however, fell into oblivion for the next ten years, as Wiesner's paper "Conjugate Coding" was not accepted. The ideas were still remembered, and after some thought distillation, the first paper [31] proposing quantum key distribution was published in 1983. For the last 30 years, since the publication of the first quantum protocol BB84, much work has been done in this field. After experiments conducted in laboratories and with technological advances, real QKDN implementations have been demonstrated worldwide. We mention a few examples that are a staple in developing QKDNs.

#### 2.1.1 DARPA Quantum Network

Historically, the first QKD network to be implemented was the DARPA Quantum Network [32], which was located in the Boston area in 2004. It consisted of ten nodes in total and was running for three years [33]. It proved the possibility of building these networks and their practicality as IPsec tunnels using keys distributed by the network were established between the sites. Four nodes interconnected by an optical switch used attenuated weak coherent pulses from lasers to generate single photons. Other sites implemented free space links using optical telescopes or entangled pairs of photons [34].



**Figure 2.1.** Topological scheme of the Vienna network. QAN is the Quantum Access Network, and the QBB is the Quantum Backbone [35].

### 2.1.2 SECOQC

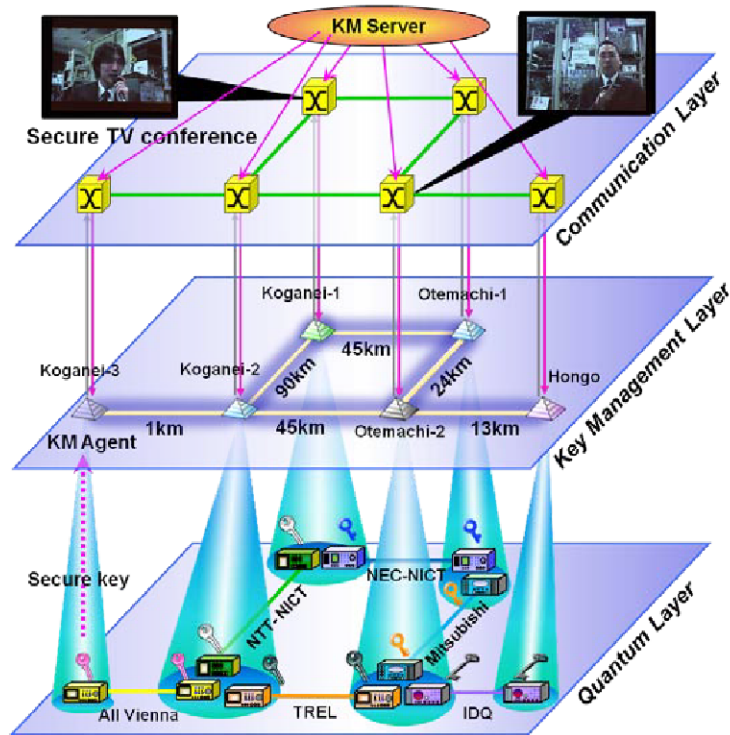
The Secure Communication based on Quantum Cryptography (SECOQC) was a project that aimed to further develop quantum cryptography. It was launched in 2004 and aimed to tackle problems with missing standardisation of QKDNs and integrating different QKD systems into one quantum backbone. The difference from the DARPA network is that the fibre-optical switch does not control the links between individual nodes, but they remain together. The architecture built in Vienna is divided into the quantum backbone network and the quantum access network, as shown in Figure 2.1. Because different implementations of QKD modules were used, a common communication protocol (Q3P) was designed. On top of Q3P, other protocols like QKD Routing Layer Protocol, based on the OSPF, or QKD Transport Layer Protocol, adopting TCP/IP, were employed [35].

### 2.1.3 Tokyo QKD Network

In 2010, another international cooperation on building a QKD network was held in the Tokyo area in Japan [36]. The ambition was to show the capabilities of the QKDN that are going to be required by the potential users. Real-time secure video conferencing and long-distance secure voice communication were demonstrated [36]. Individual links ran different quantum protocols (which will be explained in Section 2.4.1) and demonstrated the transmission of a key even at a distance of 90 km. The topology is depicted in Figure 2.2, which shows a three-layer architecture with a key relaying using trusted nodes. Architecture with a centralised key management server differs from the SECOQC.

## 2.2 Standardisation of QKDNs

Today, many companies like IdQuantique or Toshiba are constructing and selling their implementations of QKD modules, which are the main building blocks of QKDNs. Compatibility and adherence to the standards are necessary for the widespread implementation of these networks worldwide. Several **standardisation bodies are currently making efforts in this area**. The following list



**Figure 2.2.** Three-layer architecture of the Tokyo QKDN [36].

summarises the most important standardisation efforts and series of recommendations available today:

- **ITU-T:** The International Telecommunication Union publishes recommendations developed by three study groups<sup>1</sup>.
  - SG11 – Q series: Switching and signalling and associated measurements and tests
    - Q.4160-Q.4179: Protocols and signalling for Quantum key distribution networks
  - SG13 – X series: Data networks, open system communications and security
    - Y.3800-Y.3999: Quantum key distribution networks
  - SG17 – Y series: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
    - X.1700-X.1729: Quantum communication
- **ETSI:** The European Telecommunications Standards Institute issues group specifications and reports concerning use cases such as the application interfaces, security proofs, module specifications, characterisation of components, standard API for delivery of key material to applications and device communication channel parameters. The published specifications are ordered sequentially under GS/GR QKD 002–019 serial numbers<sup>2</sup>.

<sup>1</sup> <https://www.itu.int/ITU-T/recommendations/index.aspx>

<sup>2</sup> <https://www.etsi.org/technologies/quantum-key-distribution>

- **ISO/IEC:** The International Organization for Standardization and the International Electrotechnical Commission have published two standards under the IT Security (35.030) classification so far. These are the *Security requirements, test and evaluation methods for quantum key distribution*
  - ISO/IEC 23837-1:2023, Part 1: Requirements
  - ISO/IEC 23837-2:2023, Part 2: Evaluation and testing methods

## 2.3 QKDN topology

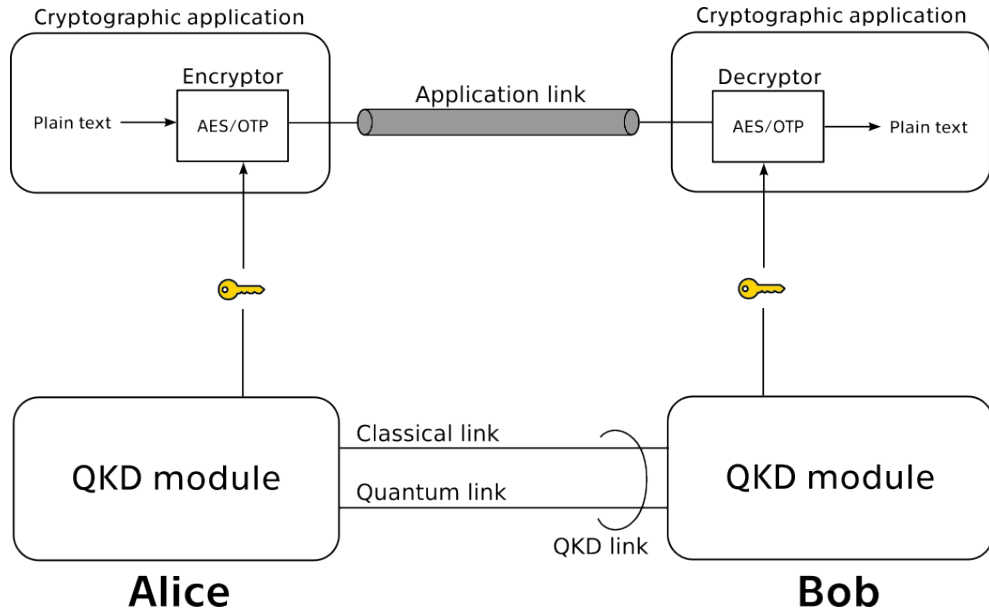
This section provides a descriptive overview of the current topologies in QKD networks. First, the simplest point-to-point (P-to-P) topology is introduced, on which principles of QKD are explained. Based on the limits of the P-to-P topology and the practical requirements people expect from a quantum network, a general, more complex multipoint topology is described.

### 2.3.1 Point-to-point network

The point-to-point quantum key distribution network is the simplest practical topology that can be built. It comprises two quantum key distribution modules. The QKD modules are specialised hardware that implements a quantum protocol (more in Section 2.4.1), which allows the exchange of data between them in a secure manner. Utilising quantum mechanical properties, detecting any eavesdropper trying to listen to or alter the transmitted data is possible. Therefore, the exchanged data, referred to as an encryption key or key material, is deemed safe for further cryptographic operations. The required property for the key material is that it must be generated entirely randomly, typically using Quantum Random Number Generator (QRNG). It again uses properties of quantum physics to generate the true source of entropy, making the generated numbers unpredictable. QRNG is typically a part of the QKD module.

The modules are interconnected using a QKD link. It consists of a quantum and a classical channel. The quantum channel is used to exchange the randomly generated bits encoded by qubits that represent a quantum state. Typically, a single photon coherent state of light is used [37]. The transmission media of the quantum channel is realised either as an optical fibre or as a free-space optical link. The classical channel is used for synchronisation and key distillation, which is a process of sifting, error correction and verification of data transmitted through the quantum channel. The classical channel can also be implemented as a dedicated optical fibre or an authenticated connection in a public network.

Figure 2.3 depicts a P-to-P QKDN topology, where on one side is Alice and on the other is Bob. Alice wants to send Bob some plain text messages through the application link, a public network, usually the Internet. Before she can do so, a symmetrical encryption key must be shared using QKD. A cryptographic application then requests a key from the QKD module, which is then supplied and used to encrypt the communication between Alice and Bob. The encryption is done using some symmetrical cryptosystem (e.g. AES), or



**Figure 2.3.** Point-to-point QKD network between Alice and Bob.

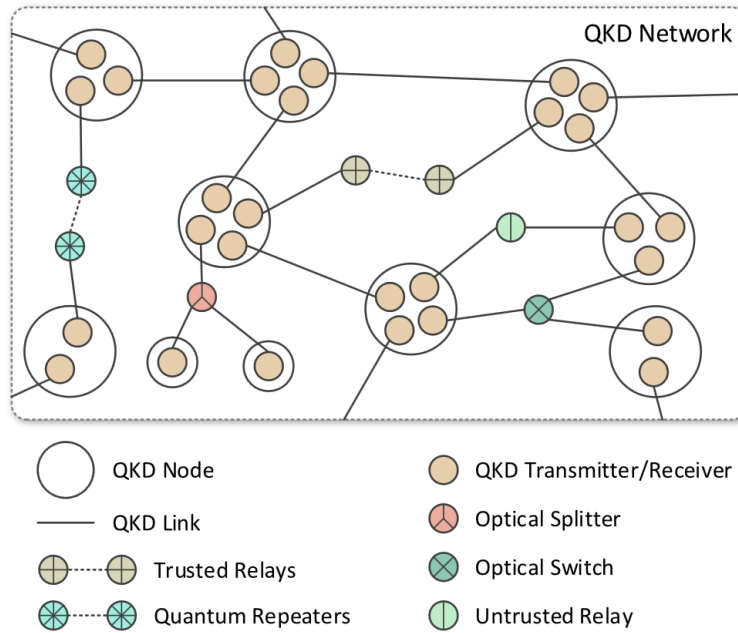
for the best possible secrecy, the key can be used as a one-time pad (OTP) in order to achieve unconditional security [38].

The one-time pad is an encryption method that guarantees the encrypted message cannot be cracked. Thus, it is information-theoretically secure because the ciphertext provides no useful information for a cryptanalysis. An adversary with unlimited computational power and time could not tell the original message. Let  $m$  be the message of length  $\text{len}(m)$  Alice wants to send Bob. She uses a key  $k$  with the following properties:

1. The key  $k$  was generated entirely randomly, using, for example, a QRNG.
2.  $\text{len}(m) \leq \text{len}(k)$
3. The key  $k$  is used only once and never again.

Alice produces the encrypted message  $m_{enc}$  by doing  $m_{enc} = k \oplus m$  and sending it over a network to Bob. Bob has the same key  $k$  distributed using the QKDN, and to get the original message, he does  $m = m_{enc} \oplus k$ . Unconditional security is one of the most important capabilities of quantum cryptography, provided the conditions are ideal; the QKD modules are constructed and working theoretically perfectly.

Constructing P-to-P QKDNs is possible but becomes unfeasible when multiple communicating parties are present. Every location would have to be directly connected to all the other locations, resulting for  $n$  sites, there would have to be  $\frac{n(n-1)}{2}$  QKD links between them. Also, a single P-to-P link is limited by the maximum distance it can reach as the signal gets attenuated and does not reach the other module. Such a single physical link can typically cover distances up to 100 to 150 kilometres. It is possible to cover much greater distances using satellite relaying. This has been successfully achieved between China and Austria [39].



**Figure 2.4.** Multipoint QKD network employing different kinds of relaying options [40].

### 2.3.2 Multipoint network

The solution to problems with the range and too many users is constructing a multipoint quantum network. Figure 2.4 shows such a topology that implements different technologies. There are several ways to extend a single P-to-P link, one of which is the so-called measurement-assisted relaying [37]. This approach uses measurement device independent QKD [41] and the twin field QKD [42]. These fall into the untrusted relay category in Fig. 2.4. The maximum distances achieved using the twin field QKD are up to 500 km [43]. The opposite is trusted relays, where multiple shorter links cover the whole distance. The transmitted key is stored in the intermediate QKD modules along the way. Therefore, it is called the trusted relaying, as the key is in the non-quantum state between the modules — this location is referred to as a trusted node. In case of unauthorised access to the trusted node, relayed keys could be intercepted. More on key relaying is in Section 2.4.2. The ideal situation in terms of security and achievable distance would be the use of quantum repeaters. These could forward the quantum signals without directly measuring or cloning them [40]. However, at the time of writing, the technology is not available for implementing quantum repeaters.

Optical switching and splitting are utilised to accommodate more users of the QKDN while reducing the number of QKD links between them, but do not address their limited length. This situation is also depicted in Figure 2.4. The most widely used method to implement multi-user QKDNs is using trusted nodes. These are the black circles in Fig. 2.4 as the QKD nodes. QKD modules inside a node interchange keys with each other in accordance with the selected



route for the key. So, trusted and untrusted relaying are both methods that allow more users to connect to the network.

## 2.4 Functional layers

A quantum key distribution network is a complex system consisting of different parts that work together, provide services, and exchange data with each other. It has to be able to provide the following capabilities [37]:

- Supply a requested key to a cryptographic application in a required format.
- Ensure all aspects of information security.
- Key management capability.
- The network control and management.
- Manage the quality of service concerning users.

Each layer has defined communication interfaces and concrete functional requirements to fulfil the abovementioned capabilities. We can generally divide the QKDN into four layers:

- the Quantum layer
- the Key management layer
- the Control layer
- the Management layer

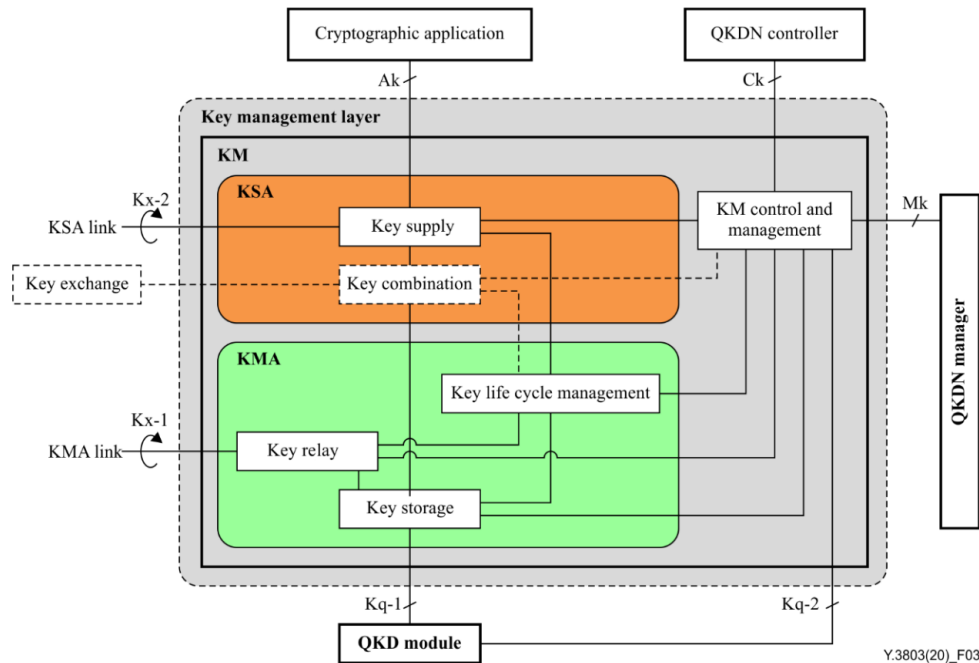
Above these, there is the User network, where cryptographic applications operate.

### 2.4.1 Quantum layer

This layer operates on the lowest level in the network hierarchy. As the name suggests, the principles of its operation are based on quantum physics and optics. It securely exchanges randomly generated bits between two directly connected QKD modules through the quantum channel. The intrinsic randomness of the quantum states provides that the generated bit stream is entirely random, which meets one of the requirements for the information theoretical security. The exchange itself is governed by quantum key distribution protocols that define the rules and procedures for secure key exchange.

There are two practical options for QKD implementation: prepare-and-measure and entanglement-based approaches. Most practical implementations today use the former approach, as the latter requires technological advances that are not available for comprehensive implementation today. Some protocols in this group is the BB84 [44] (also with decoy-state). For completeness, the entanglement-based protocols are, for example, E91 [45] and BBM92 [46].

It was already stated that the BB84 was the first QKD protocol ever developed. We describe its principles as being used, with slight optimisations and using the decoy-state, in practical implementations by companies like IdQuantique and Toshiba. Alice and Bob want to exchange a secret key. Alice prepares qubits that encode individual bits of the key using a randomly chosen basis for



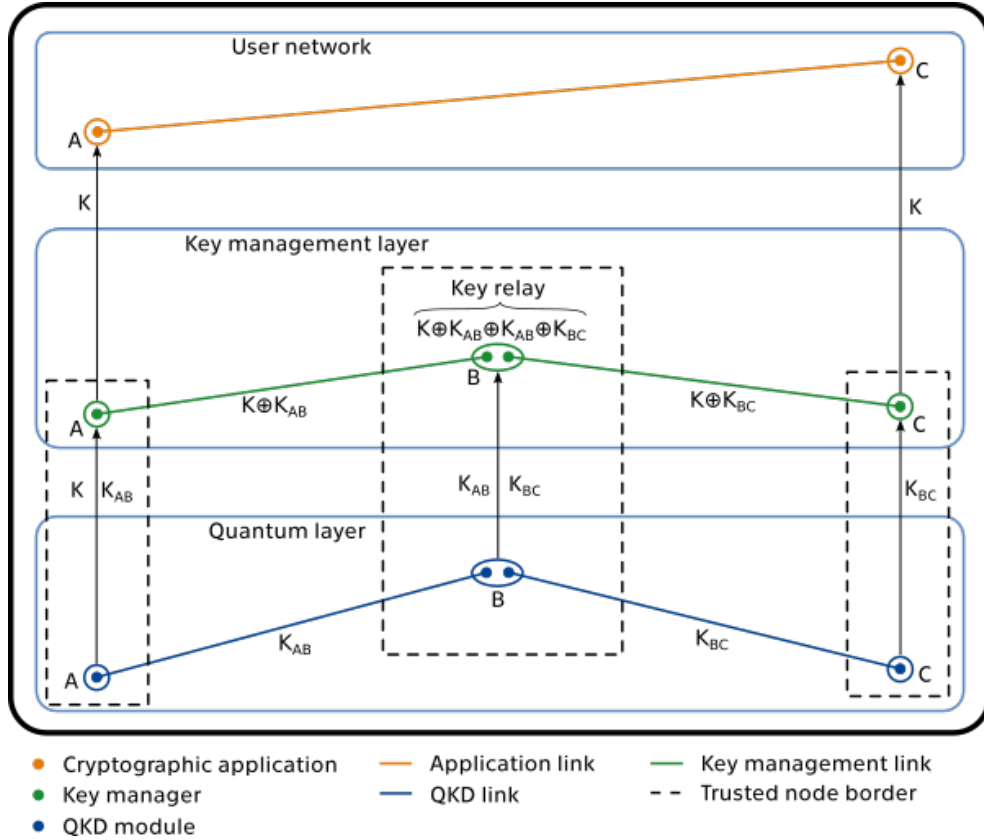
**Figure 2.5.** Functional architecture of a key management layer [47].

each qubit. Then she sends them to Bob. Bob attempts to measure received qubits by trying to guess the same basis Alice used for encoding. Measurement in Alice's basis will produce her original bit. If Bob chose the wrong basis, the measurement would result in uncertainty of the original bit value. Once all qubits are transmitted, they share all bases used. In cases where Alice and Bob used the same basis, these bits are used to generate the key further. They estimate a quantum bit error rate (QBER) in the next stage by publicly comparing some of the exchanged bits (these are then discarded). The whole process is scrapped if the QBER is above a threshold value. That can be caused by a third party, Eve, listening during the key transmission. Eve would inevitably measure some qubits from Alice using the wrong basis, possibly sending Bob a qubit encoding the wrong value (not the value Alice originally encoded).

## 2.4.2 Key management layer

A Key management (KM) layer is where a key manager operates and is responsible for storing, supplying, relaying and deleting the keys exchanged between the QKD nodes [47]. All operations performed on a key during its lifetime are considered key management. The KM layer is located above the quantum layer and below the user network. Key managers are located in every QKD node and are connected using the KM links. According to Figure 2.5, the following paragraphs describe communication interfaces and individual functional parts of a KM layer.

The quantum layer supplies keys generated using QRNG or exchanged with some directly connected QKD module through the  $K_{q-1}$  interface to the key



**Figure 2.6.** Example of trusted key relaying with three nodes.

management agent (KMA). These are securely stored and eventually reformatted with metadata for their identification. The KM control and management function governs the further processing of the keys, communicating with the QKDN controller through the interface  $Ck$  (Section 2.4.3) and the QKDN manager through the interface  $Mk$  (Section 2.4.4). Another functional element of KMA is the key relay, which exchanges keys between two key managers connected through the  $Kx-1$  link [47].

Key relay is necessary when the cryptographic applications are in two distant geographical locations such that the key distribution between them has to go through multiple QKD nodes. This is called trusted relaying, as mentioned in 1.3.2. Such a situation is depicted in Figure 2.6, where relaying is performed in node B while assuming the cryptographic applications are located close to nodes A and C. The key  $K$  is generated in node A and sent to node B through the KM link. It is secured using OTP by the key  $K_{AB}$ , which was exchanged between nodes A and B using QKD. Once  $K \oplus K_{AB}$  arrives in node B, it can be decrypted because node B also has  $K_{AB}$ . Subsequently,  $K$  is secured by  $K_{BC}$  for transmission to node C. Finally,  $K \oplus K_{BC}$  is decrypted, and the  $K$  is supplied to the other application in the user network. There are several possibilities where the key  $K$  could be generated initially. For example, if it were in node B,  $K$  would be secured by keys appropriate to the KM link and decrypted in destination nodes A and C.

The second part of a KM is the key supply agent (KSA). A cryptographic application directly communicates and requests keys from KSA using the interface  $A_k$ , as shown in Figure 2.5. KMA provides a key from its storage and sends it to the KSA. Key combination functionality is an optional feature that allows the combination of the key with another cryptographic material obtained by, e.g. PQC. This offers the advantage of being able to combine a pre-shared key (if there is one) with PQC key material in case of key depletion in KMA storage. Such a situation must be announced to the cryptographic application [48]. A hash value or message authentication should be calculated from the key on both ends of QKDN, where the request was made [49]. This synchronisation and integrity check is done via the  $K_{x-2}$  link. KSA must authenticate the application before providing the key to it. Chapter 4 discusses various secure supply options.

### ■ 2.4.3 Control layer

At this layer, QKDN controllers govern the quantum and key management layers to ensure stable and efficient operation. Controllers also communicate with the management layer. There are several implementational possibilities. A controller can be contained in every QKD node; one central controller can provide services for all nodes, or a hybrid version of the previous two options can be used. Overall, the following functionalities are provided by a controller [50]:

- The routing control function decides the best KM link for a key relay. In case of a link fault, it chooses another link to route to ensure a continuation of the key supply. The key consumption rate information is monitored and used to optimise routing decisions.
- The configuration control function initialises, monitors and configures all components of the QKD modules for an operation. Including the optical switching and splitting functionality of the quantum layer. It performs diagnostics in case of high QBER or a general failure in a quantum link.
- The policy-based control function ensures the quality of service for cryptographic applications based on the QKDN's available resources.
- The access control function authenticates the devices and restricts their activities based on policies.
- The session control function controls key relay and supply sessions.

### ■ 2.4.4 Management layer

There is usually one QKDN manager for the whole network, which communicates with all components. It manages each layer's configuration status and network topology [50]. It also performs security management and accounting. The management layer can distribute configurations to the individual nodes and update their routing tables for a key relay. Management information is also communicated to the user network controllers.

# Chapter 3

## Security Analysis

Quantum key distribution networks are complex networks consisting of many components whose individual security contributes to the network's overall security. It is generally known that a system is only as safe as its weakest component. A potential attacker is likely to attack such the weakest spot therefore it is critical to know where it lies. The following chapter will introduce computer security and risk management strategies. Further, common threats and attack surfaces in QKDN will be identified. Quantum hacking will be described as a method for attacking quantum protocols and QKD modules.

Our security analysis will be inspired by the standard ISO/IEC 15408, also referred to as Common Criteria (CC). It is a framework defining general concepts and principles of IT security evaluation [51], which also will be described. The CC is used to evaluate and certify any IT system or product. In the case of QKDNs, we could analyse each component in detail. However, it was partially done by ETSI in [52], which is a protection profile for a pair of prepare and measure QKD modules. Certified key management systems will be required for practical implementations of QKDNs. We will, therefore, focus on the security of a key management system for QKDN, for which a CC protection profile will be outlined. Many countries require CC certifications if a particular system is to be used by governmental institutions.

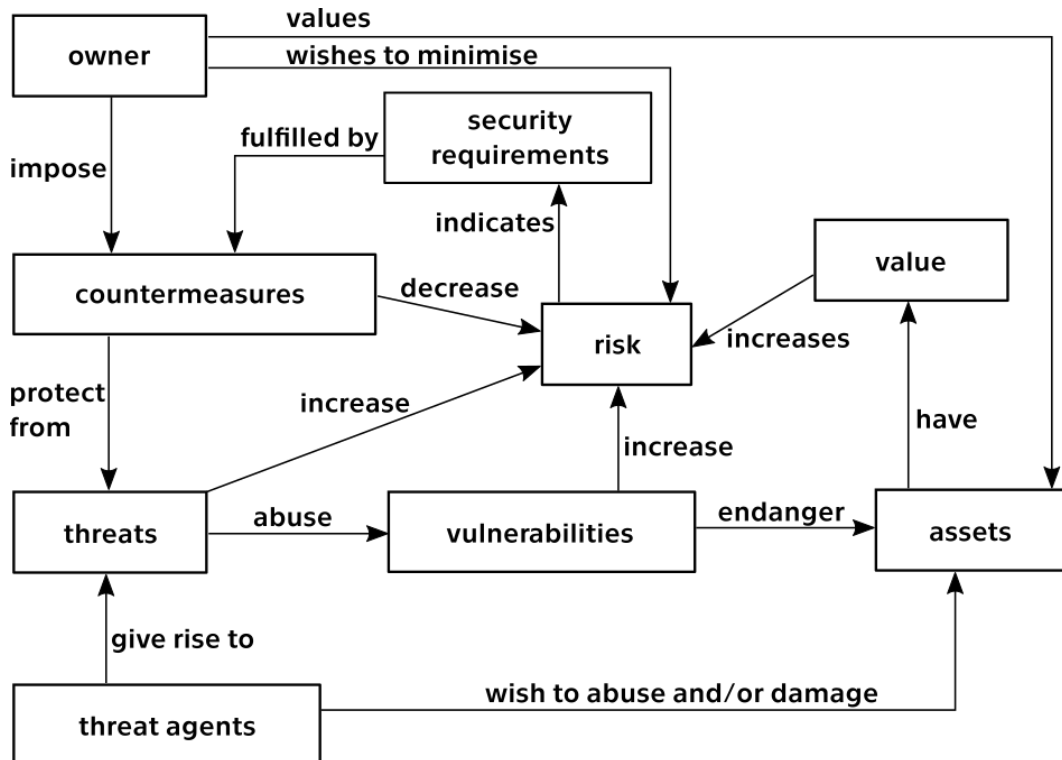
The purpose of quantum networks is to distribute a shared secret on which further security is based, which makes it an item of interest for our security analysis. It must be monitored from its creation in a QKD module throughout its lifetime in the network until it is supplied to the cryptographic application and deleted. That is because, at any time, an unauthorised acquisition of the secret by the attacker is a security breach. For the adversary, there are generally two possible intentions. Either to acquire a key that has been or will be used or to render the network unusable and hinder legitimate users from securely communicating.

### 3.1 Information security

Information security is a process of four continuously repeating stages, where one is building on the other. It is also commonly referred to as a PDCA (Plan–Do–Check–Act) cycle. The individual phases are specified and follows:



to minimise the risk to which they are exposed. This is achieved by identifying threats and vulnerabilities and formulating security requirements, which are fulfilled by the imposed countermeasures. The objective of security analysis is precisely to identify vulnerabilities, threats, and risks.



**Figure 3.1.** Interrelationships between elements of information security in a general system.

**Owner** — is a person who owns the system and places value on the assets and, therefore, has a responsibility to protect them. They own the risk of operating a system and profiting from the valuable assets.

**Asset** — is an item that has a value and shall be protected. In the context of QKDNs, it would be, for example, the shared secret, authentication keys or even the whole QKD module. In some cases, an asset can also be a person with the knowledge or authority to influence the operation of a system.

**Threat agents** — is an entity that has the potential to exercise adverse actions on assets protected by the system [51]. Examples of threat agents can be hackers, users with malicious or even non-malicious intentions, or accidents.

**Threats** — are adverse actions or events that have the potential to cause harm. They undermine the information security objectives. More on threats is described in Section 3.5.

**Vulnerabilities** — is a weakness or a flaw in a system. These can arise from errors in software, configuration errors, poor system design and implementation, and human errors.

**Security requirements** — are formulations of the system’s qualities for achieving the desired security state. Usually, it serves as a blueprint for the countermeasures.

**Countermeasures** — are concrete implementations of strategies to mitigate the risk. They help to reach the information security goals.

**Risk** — is the effect of uncertainty on objectives [54]. A detailed description is provided in the following section.

## 3.2 Risk and risk management

Prior to discussing risk management, it is necessary to define the concept of risk and its implications. As mentioned earlier, the definition introduced risk as the effect of uncertainty on objectives [54]. That is a very general definition applicable to basically any field. However, in the context of IT systems, a more suitable definition would correspond to the relationships shown in Figure 3.1. Then, the risk is the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organisation [55]. Harm in this context is to be understood as the forced adverse change to the functioning of an organisation.

Risk is tied to the uncertainty of negative actions happening. Mathematically, the level of risk  $R$  is given by the product of the likelihood  $L$  of an incident occurring and successfully causing harm and the impact  $I$  of such an incident on the organisation; thus,  $R = LI$  [56]. Establishing the likelihood of an incident happening consists of the likelihood that a threat will occur and the likelihood that the threat will successfully exploit a vulnerability [55]. The combined likelihood is influenced by factors such as:

- the motivation of the threat agents
- the difficulty of executing the attack (opportunity and location conditions)
- the knowledge required for the execution of an attack
- the technical and financial requirements

In case of a successfully executed attack, the organisation calculates its impact. The following factors influence the total impact:

- the breach of security objectives as defined in Section 3.1.1
- the loss of reputation and trust of the organisation
- the financial costs of damage repair and overall financial losses (value of the assets, loss of business, contractual penalties)

We will use all these influences to determine a risk level. It can be a numerical or a descriptive value on a scale that ranks the risks, also called the hazard scale. Evaluating the risk is essential as it is never possible to defend against all threats. A compromise must be made on how likely an incident is to happen and how difficult it is to put up countermeasures against it.

The whole impact can also be expressed in monetary value. That allows the calculation of the costs of the security measures implemented for a concrete



risk. Spending more money protecting something with a lower value would be economically pointless. We must note that putting a monetary value on an asset is not always possible, for example, putting a value on a human life.

So far, we have described the essentials of risk management. Risk management, according to the ISO Guide 73:2009, is defined as the coordinated activities to direct and control an organisation with regard to risk [57]. Its main tasks are risk assessment, risk treatment and risk acceptance. The individual areas are further characterised as [54]:

- Risk Assessment — consists of:
  - Risk Identification — identification of threats and vulnerabilities as well as opportunities and capabilities
  - Risk Analysis — ranking the identified risks on a predefined hazard scale and assigning them a likelihood of happening
  - Risk Evaluation — deciding if current countermeasures suffice or new ones should be put up based on the criteria set by the organisation
- Risk Treatment — deciding on countermeasures that will treat the risk or planning a strategy that will reduce risk by changing the likelihood of it happening or sharing it with other subjects
- Risk Acceptance — accepting and acknowledging the tolerated level of taken risk, usually the one which is costly to mitigate and has a low likelihood of happening

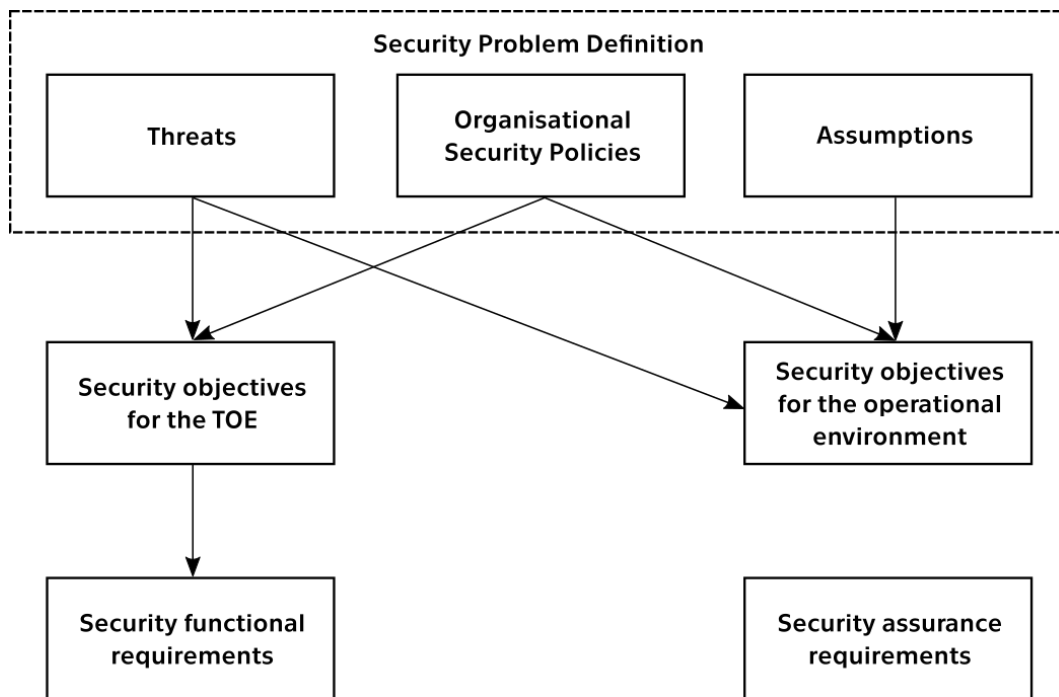
Estimating and evaluating risks is partly a matter of subjectivity. Even though methodologies exist, it is unavoidable not to apply a subjective opinion in the process of risk management.

### 3.3 General model

The Common Criteria defines a flexible framework for evaluating the security of computer systems consisting of several main building blocks. We will identify them and explain the relationships between them. The CC evaluation scheme can be very complex, so we will focus only on the key concepts.

First, a **Target of Evaluation** (TOE) must be identified, which is the specific product that undergoes a security evaluation. It may be a physical device, software application, or more complex system. A precise TOE boundary is defined to distinguish which components are relevant to the analysis. Everything outside the TOE boundary is called an operational environment. Other factors should be taken into consideration, such as the configuration of a system. Different configurations may result in different levels of security; therefore, guidance documentation for the TOE shall be provided.

A **Protection Profile** (PP) serves as a request for a specific security solution. It is completely independent of any specific implementation. Suppose a manufacturer of a system decides to follow one or more PPs. In that case, they can have the system evaluated by a certified Common Criteria testing lab and



**Figure 3.2.** Relationships between parts of a Protection Profile [51].

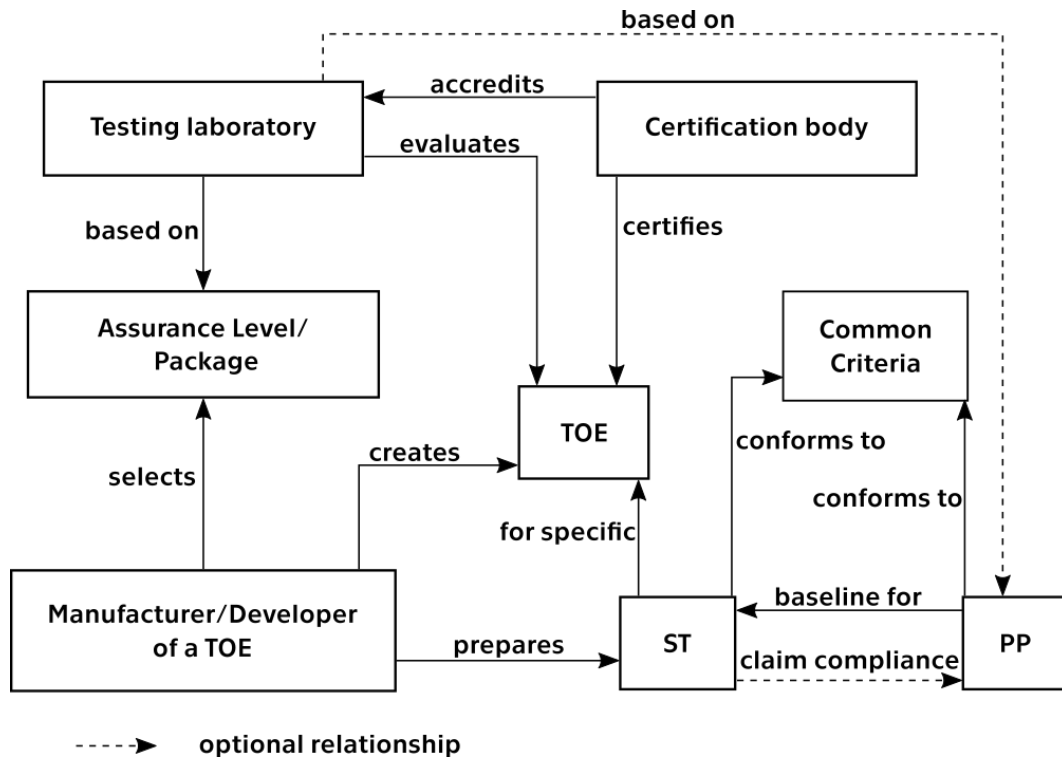
receive a certificate of conformance. The most common authors of PPs are either users aiming to reach a consensus on the requirements for a given product or governments seeking IT systems that comply with given cybersecurity requirements. A PP should consist of the following parts, whose relations are depicted in Figure 3.2:

- **Security problem definition:** Defining security problems to be addressed in the PP. That is done by identifying threats and threat agents for the TOE and its operational environment. Also, security policies are described and procedures that are to be followed in the operational environment. Lastly, assumptions about the operational environment are made so that the TOE can provide all its security functionality.
- **Security objectives:** Intended solutions for security problems are described in detail using natural language. That is done with the TOE and the operational environment.
- **Security requirements:** This section uses standardised language to describe the security requirements applicable to the TOE derived from the security problem definition. It is divided into two parts:
  - *Security functional requirements (SFRs):* a description of how the TOE addresses the security problem definition through the security objectives in a standardised language [51]
  - *Security assurance requirements (SARs):* a description of how assurance is to be gained that the TOE meets the SFRs [51]

The SFRs are still independent of any specific technical solution, as any vendor or manufacturer can use different ways to fulfil them. The secu-

curity functionalities are defined in [58] and contain a number of predefined categories, which we will use later when writing our own Protection profile.

A **Security Target (ST)** is a document provided by the vendor defining the security properties of the evaluated product. It builds upon the PP and further specifies it. It states and explains how the security objectives are met. The ST is an implementation-dependant statement of security requirements for a TOE based on a security problem definition [51]. During the evaluation process, ST serves as the primary document against which the evaluation is carried out, also stating the exact scope of it.



**Figure 3.3.** Relationships between components of the Common Criteria framework and the certification process.

Figure 3.3 provides a better understanding of how the individual parts of Common Criteria fit into the certification process and how they relate to each other. An example of a fictional company, Quantum Module Manufacturing Company (QMMC), that manufactures QKD modules will be described.

**Example:** The QMMC manufactures QKD modules implementing a prepare and measure QKD protocol. Their product claims conformance with a Protection Profile ETSI GS QKD 016 [52]. First, the QMMC must identify the TOE as a pair of QKD modules and prepare the Security Target. Such an ST will claim compliance with the ETSI 016 PP. The QMMC wants to get its product certified by a certification body, one of the national organisations of the Certificate Authorizing Member states. At the time of writing this thesis, eighteen states are eligible to issue a certification, according to their scheme. The Czech Republic is a member of the Certificate Consuming Members, so it only recognises CC certificates.

The assurance level is chosen, which is a collection of SARs that prescribes how deeply and thoroughly the TOE will be tested. There are several options for selecting the desired assurance level. There are predefined Evaluation Assurance Levels ranging from one to seven; the higher the level, the more testing, analysing and extensive documentation is required. Also, SARs can be prescribed in the PP and the ST. When all the mentioned documents are prepared and then approved by the certification body for conformance with the CC standard, a testing laboratory begins the evaluation process. The laboratory proceeds according to the assurance level and the ST that claims compliance with the PP in our example. Once testing is over, an evaluation report is filled out. Based on that, the certification body may issue a certificate stating the assurance level reached or PP compliance.

### 3.4 Users of the QKDN

To understand security concerns, we must identify critical stakeholders using the quantum network as potential threat agents. There are different roles that are required for the network to function. As it is beforehand unknown what person may have malicious intentions, we describe their usual use of the network and general access to it.

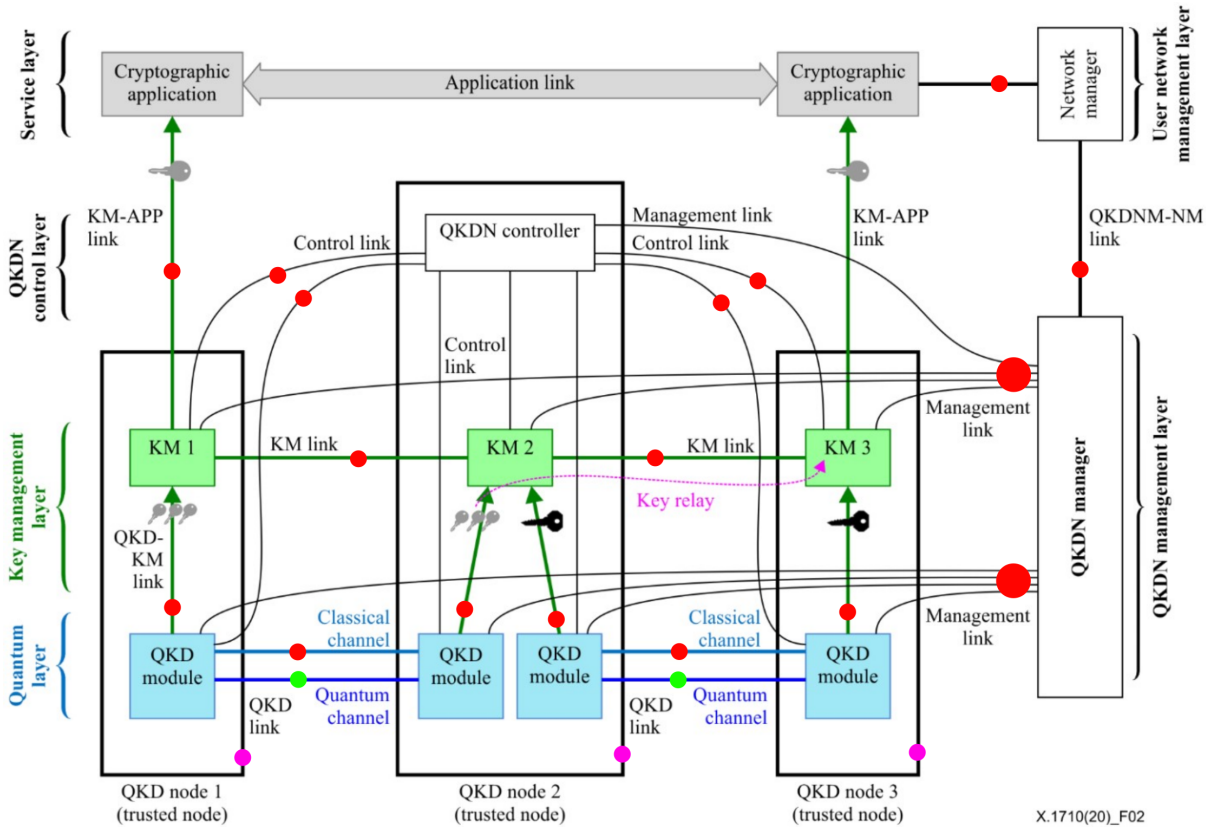
- **Regular users/consumers:** This is the majority of network users. Their legitimate intention is to obtain a key from the network through a predefined communication interface, usually an API. No physical access to the network is needed, nor are any above-standard access rights.
- **Network engineers/maintainers:** This group is responsible for building the infrastructure and ensuring the connectivity of individual components. It follows that they have physical access to the components. Because physical security is one of the critical aspects of the QKDNs, such personnel must be monitored.
- **Administrators:** The task of administrators is to configure and monitor the whole functionality of the network. There may be different levels of privileges granted to people in this group (more in Chapter 5); however, physical access to the devices is not necessary.
- **Auditors:** The role of auditors is to verify compliance with security policies. Their privileges allow them to access audit records and logs. Physical access is granted; thus, this personnel's actions are monitored.
- **Infrastructure owner:** This does not concern a specific person in the classical sense but rather a business entity which owns the physical network. The owner has unlimited access to their network and, therefore, must be trusted by its clients.

### 3.5 Threat identification in QKDNs

Any part of a QKDN is prone to vulnerabilities and is exposed to threats. Two main categories are distinguished based on their nature. First are accidental

threats, whose occurrence is hardly predictable and contains situations like an accidental misconfiguration of a QKD node, power outage or a simple hardware failure. These are important to mention, as they violate the fault tolerance objective but are not a primary concern of this section. Accidental threats are usually mitigated by implementing redundancy in the system, setting up a work policy or adhering to rules during the planning of the network (more in Chapter 5). However, deliberate threats are the result of individuals acting with intent. Generally, these can be classified into the following categories according to the STRIDE model [59] extended by the eavesdropping as we describe as follows:

- **Spoofing** is a method of acting as a legitimate computer system or a person and forging a genuine-looking communication in order to acquire sensitive data or alter the typical functioning of a system, violating authenticity.
  - spoofing any component of a QKDN
  - authentication spoofing, closely related to eavesdropping
  - spoofing of the identity of a QKDN personnel
- **Tampering** (data deletion or corruption) intentionally manipulates data stored in a memory or during transmission, violating the integrity objective.
  - any link between the QKD devices is vulnerable to intentional data corruption
  - corrupting the state of qubits during the exchange
- **Repudiation** is characterised by an attacker performing a particular action and then denying it, violating the non-repudiation objective.
  - announcing incorrect information required for QKD through the classical channel and then denying the fact
  - performing administration and configuration changes on devices and denying the fact
- **Information disclosure** (data leakage) attacks are directed at the computer system so that the attacker acquires valuable data or information about the system, violating confidentiality.
  - revealing configuration details of QKD devices
  - revealing parts of whole encryption keys
  - side-channel attacks, measuring the device and revealing details of its operation
- **Denial of service** is the targeted overloading of a system to prevent the processing of legitimate requests, violating availability.
  - flooding any communication interface with forged requests or data in QKDN
  - physical attacks breaking the communication links between QKD systems
- **Elevation of privileges** actions are directed at acquiring higher rights that do not belong to a user, violating the confidentiality and integrity objectives.
  - user attacking a communication interface of a key management system used for authentication and key delivery
  - operating personnel of a QKDN attacking the management interfaces
  - physical attacks on the trusted node
- **Eavesdropping** is secretly intercepting communication between subjects and possibly deciphering it; confidentiality is violated.
  - traffic interception on any link between the communication interfaces of QKD devices, performing the man-in-the-middle attack

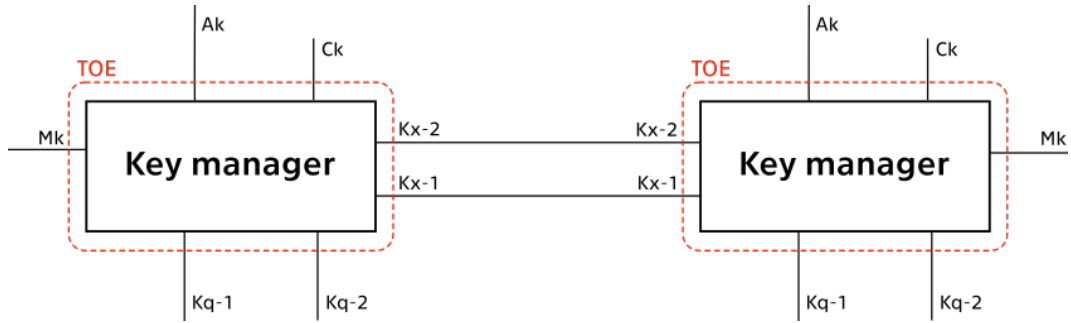


**Figure 3.4.** Three node QKDN topology with identified attack surfaces [60].

The ITU-T Security framework has also identified threats in QKDNs and the communication interfaces between individual components for quantum key distribution networks [60–61].

### 3.6 Attack surfaces in QKDNs

The attack surfaces are specific points in a network from which an attacker can attempt to carry out an attack. It is usually through a communication interface of the functional elements of a QKDN. In general, we strive to reduce the possible area of the attack surface so that the operation of the network remains unlimited and, thus, the threat exposure is reduced. Figure 3.4 shows a general functional topology of three trusted nodes doing a key relay. In Figure 3.4, red dots represent links that are abusable by threats, as described in Section 4.5. The purple dots denote the attack surface on the whole trusted node, which is concerned with physical security and attacks using the side channels. Lastly, the green dots in Figure 3.4 indicate the surface for so-called quantum attacks. These are in the expertise of quantum physicists; therefore, we will recognise the quantum channel attacks by referencing a more detailed description [62].



**Figure 3.5.** Identification of TOE with its boundary.

### 3.6.1 Quantum hacking

In an ideal world, quantum modules would be constructed perfectly so that no anomaly could be detected during the exchange of qubits. That is, however, not a state we find ourselves in. Factors like imperfect photon sources and detectors, optical channels or beamsplitters all play a role in the quantum key distribution. Due to the imperfections in the physical implementation, security loopholes can be found. Quantum hacking is a discipline which aims to use these imperfections to gain or alter information exchanged by the QKD modules. Although quantum layer attacks are theoretically possible, most of them are very unlikely to happen in real operation. They require high expertise, specific equipment and often a longer time for successful execution [62].

Quantum attacks are divided into the following groups according to the principle of execution [63]:

- Faked-state attacks
- Laser damage attack
- Detection efficiency mismatch loophole
- Time-shift attack
- Calibration loophole
- Wavelength-dependent beam splitter attack
- Trojan-horse attacks (Light injection attacks)

## 3.7 Protection profile: Key management system for QKDN

This Protection Profile describes the security requirements for a key management system for a QKDN. It conforms to the Common Criteria version CC:2022 Release 1 [51, 58, 64]. It is written to contribute to the set of protection profiles for the components of a QKDN, as no such PP is known to exist.

### ■ 3.7.1 PP introduction

The PP introduction identifies the target of evaluation (TOE), its general functional requirements and the users operating with it. Method of use is described for an operation in the live network.

The target of evaluation for this PP is a pair of connected key management systems in QKDN because it is a distributed system, and a pair is the smallest building block. TOE boundaries are depicted in Figure 3.5, where the transmission medium connecting the KM system is a non-TOE part. The communication interfaces, according to Fig. 3.5, and the typical functionality of the TOE correspond to the description in Section 2.4.2. The TOE user roles, which may act as threat agents, also correspond to the listing in Section 3.4. A specific threat agent is defined as an accident/malfunction.

In the case of a live deployment in a commercial network, it is necessary to take into account the simultaneous operation of several KM systems belonging to different service providers. While a QKDN spans a relatively wide area and contains many QKD nodes, a provider's KM system can cover only a subset of these nodes, thus operating locally. Only a certified KM system should be admitted to delivering secure encryption keys to the end users.

### ■ 3.7.2 Security problem definition

The TOE is a distributed system comprising individual key managers, which are the primary assets together with keys stored in them. Secondary assets are mainly considered operational and configuration information, such as current trusted key relay session information, audit logs, certificates used to authenticate a KM, signing keys, and possible pre-shared secrets distributed by an administrator. All assets may also be referred to as controlled resources.

Following a list of threats countered by the TOE and the operational environment, threat agents are further described by their expertise, resources and opportunities. First, the scale for properties of threat agents is given:

- Expertise: The abilities and level of knowledge about the TOE required from the threat agent
  - Competent — can follow instructions and is familiar with the TOE as any regular user
  - Proficient — knows and understands the principles of how TOE functions and can execute attacks for which training was received
  - Expert — has extensive knowledge allowing planning and implementing attacks, can work individually
- Resources: The amount of financial and technical support
  - Negligible
  - Average — resources and devices commonly accessible or purchasable by anyone
  - Vast — significant monetary resources, specialised or custom-made gear



- **Opportunity:** The time window required to fulfil a threat without being detected or stopped
  - Unlimited
  - Moderate — time window in a matter of hours to perform an attack
  - Difficult — time window in a matter of minutes to perform an attack

**Threats** with difficulty classification on a threat agent in format (expertise, resources, opportunity):

- Key depletion (T.KeyDepl) — too many key requests through the Ak interface of TOE, possibly by bypassing the quality of service quota, depleting KMA storage. (competent, negligible, moderate)
- Session forging (T.SessForg) — opening or forging a session of some other legitimately authenticated used to obtain somebody their key. (proficient, negligible, difficult)
- Physical attack (T.PhysAttack) — breaching the physical security, allowing the conduct of side-channel attacks. (expert, vast, difficult)
- System malfunction (T.SysMal) — failure to comply with security policies defined by an administrator, for example, improper or no deletion of used keys, reuse of a key.
- Submission to the subverted unit (T.SubvQKDUnit) — accepting and executing commands from a fraudulent QKDN controller or manager through the Ck and Mk interfaces. (expert, vast, moderate)
- Data tampering (T.DataTamp) — deliberate data integrity corruption during transmission to the client via the Ak interface. (competent, average, unlimited)
- Denial of service (T.Flood) — overloading the communication interface Ak with unsolicited requests. (proficient, average, unlimited)
- Unauthorised access to data and functions (T.UnauthAcc) — elevation of privileges of a subject with lower access rights, theft of keys, possible change of configuration of the KM system. (expert, negligible, difficult)
- Eavesdropping (T.Eve) — intercepting and deciphering key data supplied to a cryptographic application. (expert, vast, unlimited)

**Organisational security policies** enforced by the TOE and its operational environment:

- Key serving (OSP.KeyServ) — encryption keys will be provided to the authenticated subjects.
- Use of secure cryptographic mechanisms (OSP.SecCrypto) — only standardised cryptographic algorithms by a certification body shall be used.
- Restricted system access (OSP.RestAcc) — only a restricted group of people shall be allowed to access and configure the TOE remotely.
- Audit (OSP.Audit) — the TOE supports security auditing of administration, key relay operation and key distribution operation.

**Assumptions** about the operational environment are made. The TOE is assumed to be located in an environment where the assumptions are met.



- Security management (OE.SecMgmt) — the operational environment implements measures and policies for reacting to security incidents. It also oversees configuration and operational changes.
- Device inspection and revision (OE.DevInsp) — the operational environment shall conduct regular inspections on the TOE, supporting reliable functionality.

		Security problem definition																
		T.KeyDepl	T.SessForg	T.PhysAttack	T.SysMal	T.SubvQKDNUnit	T.DataTamp	T.Flood	T.UnauthAcc	T.Eve	OSP.KeyServ	OSP.SecCrypto	OSP.RestAcc	OSP.Audit	A.PhysSec	A.TrustSource	A.TrustwPers	
Security objectives	O.Identify								x				x	x				
	O.Authenticate		x			x			x		x		x					
	O.AccRights								x				x					
	O.QuotaDef	x						x										
	O.KeyDelete				x				x									
	O.Audit	x										x		x				
	O.SecChannel						x			x	x		x					
	O.SessionLimit		x					x										
	O.DataInteg		x			x	x			x	x							
	O.SecKeyRelay						x				x							
	OE.AccMon			x										x		x		
	OE.PhysSec			x												x		
	OE.PersSecTr																	x
	OE.SecMgmt							x				x	x			x		
	OE.DevInsp				x							x						

Table 3.1. Security objective rationale.

### 3.7.4 Security requirements

The security requirements are stated in standardised language and address the security objectives defined above. The TOE enforces SFRs, which are chosen from a catalogue provided by the CC specification [58] to cover the previously defined security objectives for the TOE. Table 3.2 traces back the SFRs to the security objectives for the TOE. Lastly, the SARs describe how the TOE will be evaluated.

**Security functional requirements.** The following descriptions of security functional requirements correspond word-for-word to the technical specification [58] and differ in the assignment and selection parts, which are to be



- 1) *administrators are allowed to perform the modify operation on the controlled objects*
- 2) *auditors are allowed to read the controlled objects*
- FDP\_ACF.1.3: The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: —
- FDP\_ACF.1.4: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
  - 1) *if the TOE's physical security is compromised*

**FDP\_ACC.1/KeySuppl Subset access control**

- FDP\_ACC.1.1: The TSF shall enforce the *Key Supply SFP* on:
  - subjects: *consumers*
  - objects: *key*
  - operations: *get\_key, get\_key\_with\_ID, get\_status*

**FDP\_ACF.1/KeySuppl Security attribute-based access control**

- FDP\_ACF.1.1: The TSF shall enforce the *Key Supply SFP* to objects based on the following:
  - subjects: *role consumer*
  - objects: *key with security attributes (key type, key usage, key ID, validity period, level of security)*
- FDP\_ACF.1.2: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
  - 1) *consumers are allowed to perform the get\_key operation*
  - 2) *consumers are allowed to perform get\_key\_with\_ID, get\_status operations on keys with appropriate security attributes*
- FDP\_ACF.1.3: The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: —
- FDP\_ACF.1.4: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
  - 1) *if the TOE is under maintenance or its physical security is compromised*

**FMT\_SMR.1 Security roles**

- FMT\_SMR.1.1: The TSF shall maintain the roles:
  - 1) *consumer*
  - 2) *administrator*
  - 3) *auditor*
  - 4) *maintainer*
- FMT\_SMR.1.2: The TSF shall be able to associate users with roles.

**FMT\_MSA.2 Secure security attributes**

- FMT\_MSA.2.1: The TSF shall ensure that only secure values are accepted for the *key type, key usage, key ID, validity period and level of security*.

**FRU\_PRS.2 Full priority of service**

- FRU\_PRS.2.1: The TSF shall assign a priority to each subject in the TSF.
- FRU\_PRS.2.2: The TSF shall ensure that each access to all shareable resources shall be mediated based on the subjects assigned priority.



- FTP\_TRP.1.2: The TSF shall permit *remote users* to initiate communication via the trusted path.
- FTP\_TRP.1.3: The TSF shall require the use of the trusted path for *initial user authentication and communication to the API for a key request and supply*.

**FTP\_ITC.1 Inter-TSF trusted channel**

- FTP\_ITC.1.1: The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from *modification or disclosure*.
- FTP\_ITC.1.2: The TSF shall permit *another trusted IT product* to initiate communication via the trusted channel.
- FTP\_ITC.1.3: The TSF shall initiate communication via the trusted channel *for exchanging information and operations required for key relay and authentication and synchronising keys between TOEs*.

**FPT\_PHP.1 Passive detection of physical attack**

- FPT\_PHP.1.1: The TSF shall provide unambiguous detection of physical tampering that can compromise the TSF. *In such an event, all keys shall be securely deleted*.

**Security assurance requirements.** The TOE shall be evaluated at the EAL6 (Evaluation assurance level), augmented with ASE\_TSS.2 to reach prescribed security functional requirements. This level provides semi-formal verification and testing. It was chosen as it first contains the highest vulnerability analysis available. Considering how vital the architectural design of a key management system in a QKDN is, the ASE\_TSS.2 was selected to require the TOE specification with an architectural design summary. The specification of EAL6 is defined in the CC standard [65].

	Security objectives									
	O.Identify	O.Authenticate	O.AccRights	O.QuotaDef	O.KeyDelete	O.Audit	O.SecChannel	O.SessionLimit	O.DataInteg	O.SecKeyRelay
FIA_UID.1	x	x								
FIA_ATD.1	x									
FIA_API.1		x							x	
FIA_UAU.1		x								
FIA_UAU.5		x								
FDP_ACC.1/Mgmt			x			x				
FDP_ACC.1/KeySupply			x	x						
FDP_ACF.1/Mgmt		x	x			x				
FDP_ACF.1/KeySupply		x	x							
FMT_SMR.1			x			x				
FMT_MSA.2			x							
FRU_PRS.2				x						
FRU_RSA.2				x						
FAU_GEN.1						x				
FTA_SSL.3								x		
FTA_SSL.4								x		
FCS_CKM.1					x				x	x
FCS_CKM.2							x			x
FCS_CKM.3							x		x	
FCS_CKM.6					x					x
FTP_TRP.1							x		x	
FTP_ITC.1							x		x	x
FPT_PHP.1										x

**Table 3.2.** Security functional requirement rationale.



## Chapter 4

# Design Towards Optimised QKDN Architecture

Despite extensive standardisation efforts, many practical problems still need to be solved before QKDNs can be widely deployed. Some real-life implementations in Chapter 2 were mentioned that proved the theoretical functionality of these networks. We are on our way to building full-fledged quantum networks in the future, but the main obstacle is the need for more advanced technology, like quantum memory. The goal, for now, is to show that the trusted node networks implementing prepare and measure quantum protocols are feasible for commercial use. Governmental institutions, critical infrastructure facilities, and business subjects like banks could benefit from having access to potentially unconditionally secure communication. This chapter identifies and proposes some steps leading towards an optimised QKDN architecture with respect to security.

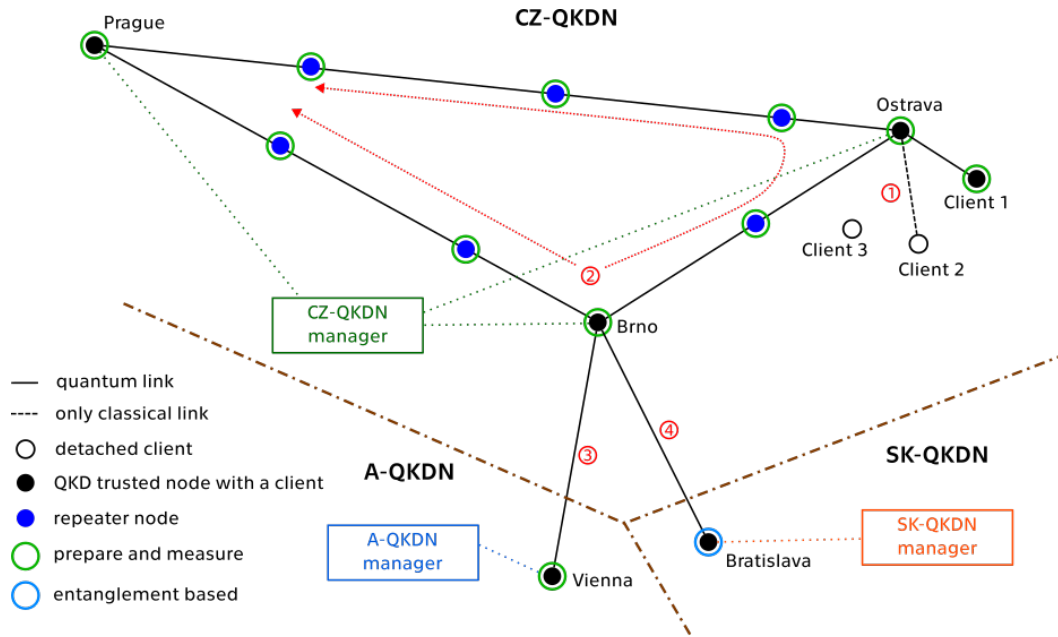
On top of increasing the security of the key management system in Chapter 3, in this chapter, we will strive to increase the security of the QKDN as a whole. Conceptually, we will approach this chapter by illustrating an example of realisable architecture like the one in Figure 4.1. It depicts a QKDN between major Czech cities and connecting abroad. The red numbers in a circle correspond to the following problems:

- 1) How to distribute an encryption key to clients that are located far from the QKD module outside the trusted node in a secure manner. (4.1)
- 2) How to enhance security by providing multiple encryption keys through different paths and their combination. (4.2)
- 3) Interworking of separately managed QKDNs and how to ensure their functioning. (4.3)
- 4) Interoperability between QKD modules that implement different QKD protocols or are based on different technologies. (4.3)

Solutions to these problems are addressed in the following sections, listed in parentheses at the end of the problem definition.

### 4.1 Key delivery to cryptographic applications

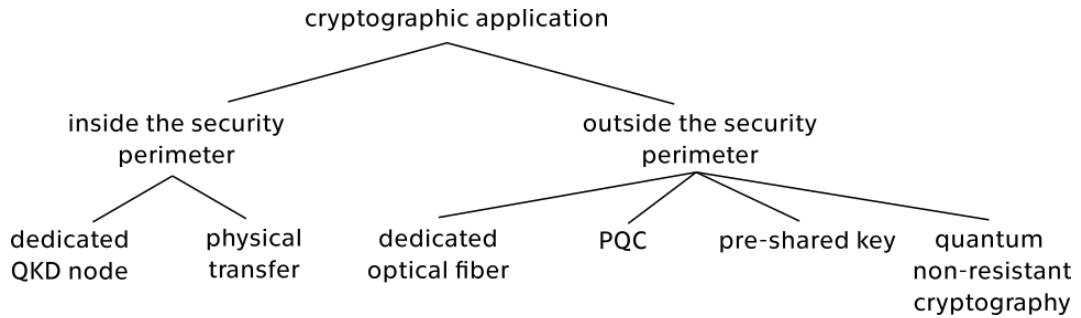
Considering a typical four-layered QKDN architecture described in Chapter 2, there is one more layer, the application layer, which is part of the user network. According to the recommendations by ITU, the communication interfaces between the QKDN and the user network shall provide privacy protection



**Figure 4.1.** The realisable topology of a QKDN consists of three interconnected, separately managed networks (Czech, Slovak, and Austrian). A client in this context is a cryptographic application utilising keys transferred by the QKDN. A repeater node is a trusted node without a client.

schemes [37]. However, no concrete solutions are prescribed. From the perspective of a client’s cryptographic application, it establishes a connection with a key management system; mutual authentication is performed, and a request for a key is sent through an API. Typically, it can be a REST API according to the standardisation by ETSI [66–67]. We will suggest security profiles that address the possibilities of transferring the key to the user (cryptographic application). Each security profile has assumptions that depend on a delivered key’s required security properties.

According to the diagram below, we have to distinguish whether the cryptographic application is included in the security perimeter of a trusted node or not. Strong security assumptions are made about the environment inside the security perimeter of a trusted node so that the confidentiality and integrity of connections between devices can be passed on to it. Assuming further that the QKDN can exchange keys such that no adversary can intercept and acquire them. It is undesirable to degrade the potential for information-theoretical security QKD offers. In that case, data exchanges are encrypted using the OTP and authenticated by Wegman–Carter [68] type of authentication. Thus, applications located inside the security perimeter are able to authenticate and communicate information theoretically securely. The opposite situation presents a case where the application has to establish an authenticated connection to the KM system, relying on cryptography based on computationally hard problems. This solution stems from the non-existent initial shared secret between the KSA and the cryptographic application, which is usually the outcome of the authentication process. Such a secret is then used to secure communication.



**Figure 4.2.** Tree of possible locations of a cryptographic application and the appropriate methods for key delivery.

Having distinguished the main differences and limitations, several methods for the key delivery will be presented.

#### ■ 4.1.1 Security profiles — application inside the security perimeter

**Dedicated QKD node:** This solution is considered the most secure. By the nature of things, when we transport the key in quantum form directly to the cryptographic application in the same QKD node. However, the requirements for the user are vast. The costs of a quantum module and the dedicated quantum channels must be considered. Also, the distance between the quantum backbone network and the user must be a maximum of approximately 100 km. Trusted repeater nodes are unsuitable for the user access network due to high operational costs and security requirements. This solution is likely suitable for large institutions located in cities near the backbone network.

**Physical transfer:** This method is unconventional but feasible in situations when a small amount of data needs to be transferred very securely over large distances. Assuming a trustworthy courier, under supervision, enters the security perimeter of the closest QKDN node and draws keys on an encrypted hard drive. The authentication depends on the QKDN provider's policy. The keys are physically transferred to the cryptographic application, which consumes them at its need. The main disadvantage is the interruption of encrypted data transmission once the keys are depleted.

#### ■ 4.1.2 Security profiles — application outside the security perimeter

All solutions in this section degrade information theoretical security due to the inevitable use of asymmetric cryptography schemes [61]. So, there will be an effort to minimise the risk to which we expose a key from a QKDN for the end-user. Many of the following methods are suitable for use simultaneously.

**PQC:** Even though PQC algorithms are sometimes perceived as an alternative to QKD, their cooperation is desired. From the future standardised algorithms listed in Table 1.2, the algorithm Kyber is suitable for transferring a key to the user. Kyber is a key-encapsulation mechanism (KEM) algorithm. KEM

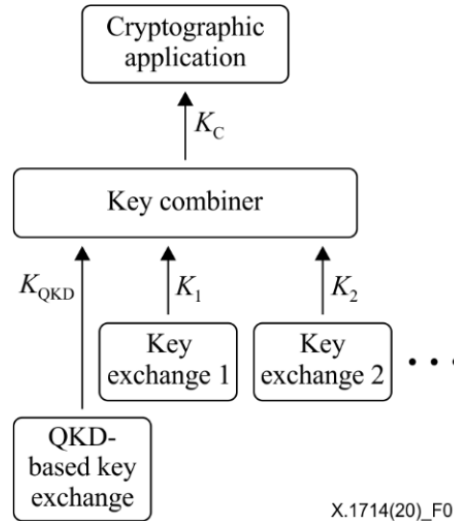
secures a symmetric key for transmission using a public-key algorithm. Part of the KEM is also a key generation, which is unnecessary as the QKDN supplies the encryption key. Kyber can, however, provide public key encryption, which would secure the transmission of an encryption key ( $\text{QKD}_{key}$ ) from the QKDN to the user. It is called the Kyber.Hybrid, and it combines the KEM with a symmetric encryption scheme.

First, a client (cryptographic application) must have its public-private keypair ( $pub, priv$ ). The key manager performs encapsulation operation  $(CT_{encaps}, K_{sym}) \leftarrow \text{Kyber.Encaps}(pub)$ , producing a key  $K_{sym}$  for a symmetrical cryptosystem and its encrypted encapsulation  $CT_{encaps}$ . Then, a  $\text{QKD}_{key}$  is encrypted using symmetric encryption with  $K_{sym}$ , producing  $CT_{sym}$ . Both  $(CT_{encaps}, CT_{sym})$  are sent to the client, which then decapsulates  $K_{sym} \leftarrow \text{Kyber.Decaps}(priv, CT_{encaps})$ . The  $\text{QKD}_{key}$  is then easily retrieved by deciphering  $CT_{sym}$  with  $K_{sym}$  [69]. Such use will be possible once the post-quantum public key infrastructure is deployed so that the identities of the communicating sides can be verified. PQC is also viable when a user is connected over a public network, as this solution is quantum-safe.

**Pre-shared key:** A pre-shared key (PSK) between the KM system and the client can be used to derive an encryption key for the QKD key transfer. For this purpose, TLS 1.3 is suitable. The current implementation allows PSK to be used in the key exchange phase. The disadvantage of this approach is reduced security in the case of a poorly chosen PSK with low entropy. In that case, the handshake is vulnerable to passive exhaustive search attacks, yielding the traffic encryption keys. PSK can be used with Diffie–Hellman key exchange to enhance security in TLS 1.3, as there are two independent factors. Even though DH key exchange is not future-proof, together with PSK, they represent a temporary option for securing the QKD key transfers. The advantage of this approach is that possessing appropriate PSK easily proves the authentication of the communicating sides.

**Quantum non-resistant cryptography:** Quantum non-resistant asymmetric schemes may seem illogical to use because that is what we are trying to avoid. However, their use is justifiable in some cases, although not ideal. Assuming a computer network outside the security perimeter is directly connected to a router inside a trusted node. An example of such a situation is a large institution connected to the QKDN through a dedicated QKD node. The security perimeter cannot span the entire office building, but the employees must receive keys. With a strong assumption that it is improbable that there is an attacker on the inside and that the user network is well protected, such security for QKD keys is sufficient. A significant advantage of this solution is that it is technology used everywhere today, with the support of current public-key infrastructure. Again, this is a temporary solution until at least post-quantum cryptography is implemented in most personal computers.

**Dedicated optical fibre:** Using dedicated optical fibre is concerned with security on the physical layer. Eavesdropping on optical fibre is already quite difficult today. Most of it shall be physically inaccessible. In the case of a continuous piece, the fibre can be monitored for interruptions or spikes in attenuation, revealing tampering attempts.



**Figure 4.3.** Key combination method for QKDN [48].

## 4.2 Methods of enhancing key security

The method by which most of today’s QKDNs will be built must use trusted nodes. In previous chapters, we placed many security demands on the trusted nodes, stressing physical security. This spot is still a weak point in the design of the QKDNs, as there are few guarantees. We propose several methods to reinforce security if a trusted node is unknowingly compromised. These methods include a key combination that allows users to rely on different key exchange methods and a multi-path key delivery in the QKDN.

### 4.2.1 Key combination

Figure 2.5 shows the functional architecture of a key management system. One of the optional features is a key combination functionality. The combiner takes on the input two or more statistically independent keys and produces a final key, which is supplied to a cryptographic application. Figure 4.3 depicts a scheme of the combination with a  $K_{QKD}$  supplied by the QKDN and other  $K_{1..n}$ , typically acquired using asymmetric cryptography. There are several methods how to perform key combination:

- Concatenation: Simple concatenation of multiple keys, serving as an input for a KDF, which outputs the final key of the required length [70].
- Operation XOR: If the keys are the same length, the XOR operation combines them.
- Key derivation function (HKDF, PBKDF2)
- Hashing the concatenated keys using, e.g. SHA256, SHA512.

This approach is beneficial in case the QKDN cannot supply the keys due to depletion. The other exchange methods may substitute the QKD and produce an encryption key. Such a situation must be communicated to the application. There is another approach, considering the cryptographic application

governing the key combination. The individual encryption keys can be used independently, performing so-called cascade encryption. It requires more computational power. However, the individual encryption layers are independent of each other; thus, an adversary would have to break all the encryption layers to get to data. The overall security is maintained if the algorithms are chosen correctly, even if one encryption layer is broken [71].

### 4.2.2 Multi-path key delivery

The multi-path key delivery is based on the idea that the keys are delivered using two or more different paths which share as few trusted nodes as possible. The number of different paths taken is a parameter on which a cryptographic application agrees with the QKDN provider and depends on the required security. For simplicity, we will assume only two distinct paths. A QKDN controller is responsible for routing control. This functionality finds the appropriate paths to use for the delivery. In the end node, the two keys are combined as stated in Section 4.2.1 or independently supplied to the application.

There are numerous criteria on how to choose the ideal paths. We will suggest a method built on top of a routing mechanism used in the SECOQC QKD network. It is a customised routing protocol based on the OSPF [72]. OSPF is a link-state routing protocol implementing a Dijkstra algorithm to find the shortest path in a weighted graph with non-negative weights. It uses link bandwidth to determine the link cost in classical data networks. Large QKDN topologies consist of multiple networks that are under different QKD managers. A single instance of OSPF will operate only on QKD nodes under the same QKD manager. Even though it is not our primary focus, we propose a different metric calculation that is more suitable for QKD networks, noting that there are many other feasible metrics.

A general QKDN topology can be modelled as an undirected graph. Each vertex represents a trusted node; the edge is the point-to-point QKD link between them. A path from  $v_1$  to  $v_n$  is a sequence  $P = (v_1, e_{1,2}, v_2, \dots, e_{n-1,n}, v_n)$  of vertices ( $v$ ) and edges ( $e$ ) such that  $e_{i,j} = \{v_i, v_j\}$  and no vertex is repeated. The routing protocol chooses a path with the lowest metric, which is calculated based on the parameters of the path. Each QKD node has a controller that is aware of the whole topology of the QKDN and contains a link-state database for it. According to [72], every QKD node should have calculated the shortest path tree to compare multiple different paths to the same destination (if they exist). The factors that are taken into account for the proposed metric calculation of a particular path  $P_{a,b}$  from  $a$  to  $b$  are:

- $n(P_{a,b})$  is the total number of edges along the path  $P_{a,b}$ .
- $n^*(a,b)$  is the number of edges along the shortest path from  $a$  to  $b$ .
- $K_e(t)$  is the amount of available key material for securing communication over a link  $e$  at a discrete time  $t$  in bytes.
- $G_e(t)$  is the amount of generated key on the QKD link modelled by the edge  $e$  at a discrete time  $t$  in bytes.
- $C_e(t)$  is the key consumption rate on the QKD link modelled by the edge  $e$  at a discrete time  $t$  in bytes.

Considering a set  $S_{a,b}$  of all possible distinct paths from vertex  $a$  to  $b$ . Then, the overall metric  $m$  for a path  $P \in S_{a,b}$  is calculated as follows:

$$m(P) = \alpha \left( 1 - \frac{n^*(a,b)}{n(P)} \right) + \beta \left( 1 - \min_{e \in P} \left( \frac{1}{2} + \frac{1}{\pi} \arctan(K_e(t) + G_e(t) - C_e(t)) \right) \right)$$

We set the weighting parameters of the components to  $\alpha = \frac{1}{3}$  and  $\beta = \frac{2}{3}$ . The first component of the equation evaluates the path length to the shortest path; the second one characterises the state of the worst link on the path in terms of stored keys and key generation performance. The parameter  $\beta$  is slightly greater as we want to prefer links with enough keys. The best path is then selected as:

$$P^* = \min_{P \in S_{a,b}} m(P)$$

If there are more paths with the same minimum metric,  $P^*$  is chosen randomly from these. If it exists, the best alternative path  $P_{alt}$  is chosen from all the remaining paths as

$$P_{alt} = \min_{P \in S_{a,b}^*} m(P)$$

$$S_{a,b}^* = \arg \min_{P \in S_{a,b} \setminus P^*} \phi(P^*, P)$$

where  $\phi(P_1, P_2)$  returns the number of vertices that are simultaneously in path  $P_1$  and  $P_2$ .

The advantage of this routing metric is that the first selected route uses a shorter path with a higher key generation rate for OTP. That is beneficial as different nodes are used to distribute the load. When the keys for securing the key relay are running low, the algorithm will prefer another path so there is enough time to replenish them. Also, if the delivery along the alternative path timeouts, the first key will likely be delivered with higher reliability.

### 4.3 Interworking of the QKDN networks

Massive deployment of QKDNs will require their interconnection. Figure 4.1 shows a situation in which there are three separately managed networks. These are labelled according to the country they are located in but are generally any network provider. Such one network can be compared to the autonomous systems in classical networks, which also constitute a separately administered domain. The challenge is to make multiple different domains cooperate and be able to distribute keys to applications located each in a different one. It corresponds to problems three and four, as marked in Figure 4.1 by a red circle.

Nodes belonging to different domains should not be able to communicate with each other in general. Network providers do not share confidential information such as client database or configuration of nodes. Only nodes on the

edge of a network shall exchange necessary information for interworking, such as keys and their metadata, routing and session information and mutual authentication and authorisation messages. The QKDN managers communicate mainly charging information and help coordinate routing through the network. Such functionality is called the gateway function (GWF), and the node capable of GWFs is the gateway node (GWN). There are two options for interconnecting GWNs [73].

- 1) Each GWN is located inside the perimeter of a provider's network and is connected on all functional layers (quantum, key management, control and management layer). It is an ordinary point-to-point link, like the one illustrated between Brno and Vienna in Figure 4.1. The advantage of such a solution is that it has no extra cost. However, both providers must possess QKD modules that implement the same QKD protocol.
- 2) The two GWN nodes are concentrated in a single location called the interworking node (IWN). It comprises a single trusted node on the network boundary containing QKD modules from both network providers. These are, however, not connected on the quantum layer. A key is exchanged between the networks using the key managers' key transfer capability inside the IWN. This situation is depicted in Figure 4.4 and is advantageous as both providers may use different QKD protocols and can still interconnect. That corresponds to problem four in Figure 4.1. Drawbacks are higher costs as each side has to operate one more QKD module. The IWN trusted node is also a potential vulnerability, as it hosts the technologies of two subjects, allowing for mutual access to them.

### 4.3.1 Threat identification

The ITU-T Recommendation Y.3810 [73] specifies the framework for interworking; however, it states that the threat identification is outside the scope of that recommendation. Thus, this section identifies threats to the components of a QKDN that carry out a gateway function. The potential threats to communication interfaces and functional elements are highlighted in Figure 4.4. Possible threats taken into account are listed in Section 3.6.

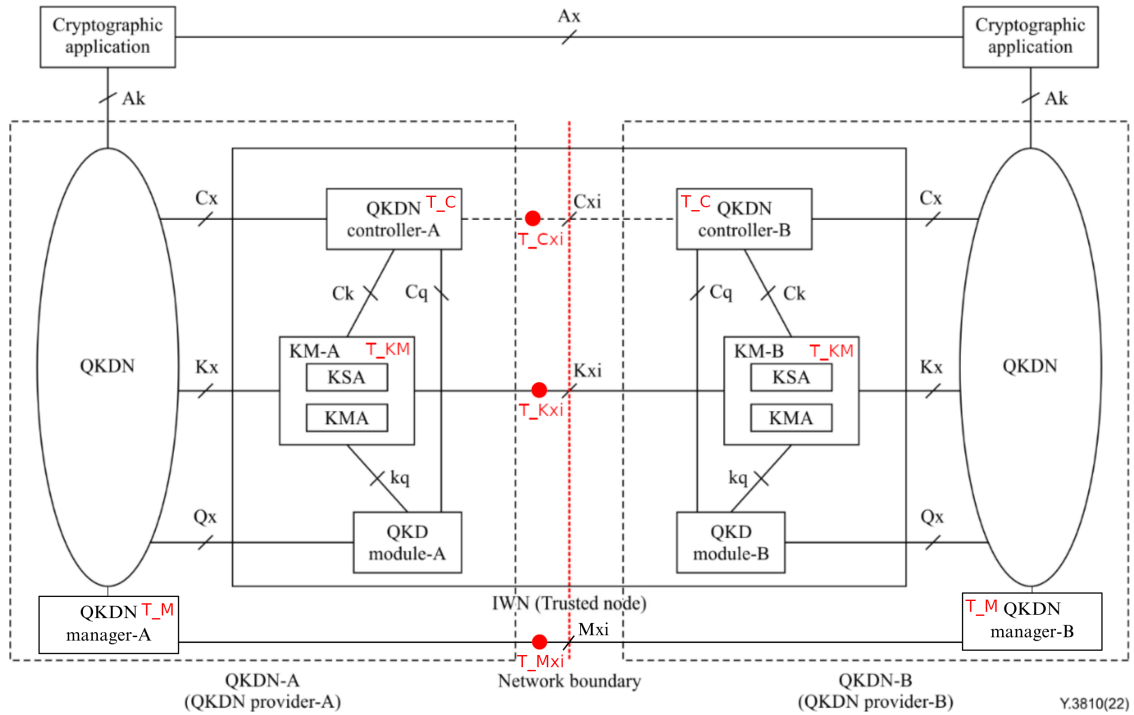
**T\_M:** Security threats on a QKDN manager through a Mxi link.

- **Spoofing:** An attacker crafts a fraudulent QKDN manager and can send or receive information from legitimate managers in other QKDNs.
- **Repudiation:** An attacker modifies records (e.g. billing information, QoS policy) about the state or setting of the network and subsequently denies the fact.
- **Information disclosure:** An attacker makes the QKDN manager disclose confidential information.

**T\_Mxi:** Security threats on the Mxi link.

- **Tampering:** Data between QKDN managers are deleted or modified, thus hindering communication.
- **Denial of service:** The link is flooded with generated valid-looking information or overloaded with requests.
- **Eavesdropping:** An attacker can intercept, decipher and read the traffic.





**Figure 4.4.** Scheme of functional components and communication interfaces of an interworking node. The red dots and labels represent parts of IWN exposed to threats identified in 4.3.1. [73] (modified).

**T<sub>KM</sub>:** Security threats on a key manager through Ck, Kxi and kq links.

- **Spoofing:** A fraudulent key manager is put in place to intercept transferred keys from the other network.
- **Repudiation:** An attacker sends cryptographically weak keys to the KM or performs other key management actions and then denies the fact.
- **Information disclosure:** Keys, metadata, and the operations on them can be disclosed due to incorrect configuration or a vulnerability in the key management software.
- **Elevation of privileges:** A hacker can acquire administrator privileges due to authentication and authorisation errors.

**T<sub>Kxi</sub>:** Security threats on the Kxi link.

- **Tampering:** The transferred keys are corrupted, deleted or otherwise manipulated.
- **Denial of service:** The link is flooded by fake key transfers, overloading and hindering the processing of legitimate traffic.
- **Eavesdropping:** Capturing keys during the transfer.

**T<sub>C</sub>:** Security threats on a QKDN controller through Cxi, Ck and Cq links.

- **Spoofing:** A fraudulent controller breaches information security by receiving and sending control information and making it seem like some other controller did it.
- **Repudiation:** An attacker issues a malicious configuration to thwart the provisioning of key transfer routes and then denies the fact.



# Chapter 5

## Methodology for Development, Deployment and Operation of QKDNs

The developed methodology concerning the development, deployment and operation of QKDNs is essential to advances in the construction and use of QKDNs. Investors and large companies must be presented with a methodological solution on how to realise a QKD network successfully and fulfil the demands for a modern cryptographic infrastructure. The deployment demands theoretical knowledge and practical implementation. Educated people are integral to this process as they build and operate the infrastructure.

This chapter aims to help face these challenges by offering guidance on how to proceed when building, maintaining or connecting to the QKDN. First, the overview and recommendations for the development of QKD modules are given, followed by strategies for network deployment, highlighting operational resilience and seamless integration within the existing infrastructure. Lastly, the focus is on proper network operation, which includes operational policies and security methods that minimise the risk of a security incident. This chapter essentially serves as a hub for navigating the complex process of incorporating quantum networks into everyday practice.

### 5.1 Development

Developing QKDNs is a collective effort of experts in many different professions. As it was described in Chapter 2, the architecture consists of several layers. The quantum layer is integral as the QKD modules perform the key distribution itself. It is a device containing specialised optical hardware capable of optical processes for random number generation and, consequently, for the QKD. Hardware development is the domain of physicists, opticians and cryptographers. Any device must have software that controls it and allows configuring it. Both hardware and software development should follow current recommendations and specifications, which include the following:

- **ETSI GR QKD 003:** *Quantum Key Distribution (QKD); Components and Internal Interfaces* — This group report is concerned with specifying properties for quantum physical devices such as the photon sources and detectors and other hardware components.
- **ETSI GS QKD 008:** *Quantum Key Distribution (QKD); QKD Module Security Specification* — This group specification defines the properties of the QKD



- **ETSI GS QKD 014:** *Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API*— This group specification somewhat builds on the previous specification by defining a REST API between the KM system and the application as it is more straightforward and widely deployed.
- **ITU-T Y.3803:** *Quantum key distribution networks – Key management* — This recommendation addresses the technical specifications of a key management system, including functional requirements, KM operations and key formats.

Lastly, the QKDN controllers and managers are mentioned. These typically do not require any specialised hardware and, thus, are entirely implemented by software. The specifications that the programmer should follow during implementation are determined in the following documents:

- **ETSI GS QKD 015:** *Quantum Key Distribution (QKD); Control Interface for Software Defined Networks* — This group specification defines functionality and integration with software-defined networks.
- **ETSI GS QKD 018:** *Quantum Key Distribution (QKD); Orchestration Interface for Software Defined Networks* — This group specification defines an orchestration interface between the QKDN and the software-defined optical transport network, which a QKDN manager implements when such an integration is desired.
- **ITU-T Y.3804:** *Quantum key distribution networks – Control and management* — This recommendation specifies procedures, functions and functional elements of the control and management of a QKDN.

## 5.2 Deployment

Current data networks can be divided into backbone, metro and access networks. Each differs in its purpose. A similar distinction applies to quantum networks, which integrate in all these areas. That is because it is advantageous to maximise the use of the already laid out optical infrastructure. A QKD link comprises a quantum and classical channel. These are typically two physically separated optical fibres that provide isolation from noise inducted by the other channel. It is, however, possible to multiplex both these channels into a single optical fibre, thus reducing resources and costs. When planning quantum networks, it is vital to be aware of the topology of the current optical infrastructure.

This section focuses on deploying the QKDN, which includes planning the topology of the QKD network and the requirements for an end-user access node. The former is regarding the placement of trusted nodes with an emphasis on reducing costs while ensuring sufficient robustness. The latter specifies the procedure for implementing the access node, including security requirements. Several articles [74–75] dealing with this issue plan the network as a whole. We, however, provide strategies for deploying each part of the QKDN separately from the backbone and metro to the access network.



- 2) Determine a threshold  $T_{min}$ , which determines a required minimum SKR between any two directly connected QKD modules in the network.
- 3) Set the cost of an edge in graph  $G$  to a predicted SKR calculated using the equation above.
- 4) Create a graph  $G' = (V, E')$ , such that  $E'$  contains only edges from  $E$  whose cost  $c(e) \geq T_{min}$ . That may cause the graph to break into more components, which indicates that there may be a need for the construction of a new optical connection. Further on, we suppose that the graph  $G'$  comprises one component.
- 5) Set costs of every edge in  $E'$  to 1. That is done because, further on, we need only the total length of a path in terms of the number of edges.
- 6) To strive for as few trusted nodes as possible in the QKDN, we need to connect all vertices in  $S$  using the least possible number of edges in  $G'$ . That is because every edge represents a point-to-point link and requires two additional QKD modules, which are costly. We need to compute a subset  $N$  of  $E'$  such that:
  - a) all  $s \in S$  are connected by paths composed only of edges in  $E'$
  - b)  $\sum_{s \in S} c(s)$  is a minimum

This is known as the Steiner problem on  $S$  in graph  $G'$  [77]. It is an **NP**-hard problem [78]. However, there are approximation algorithms that solve such a problem within a given threshold, for example [79].

The solution to the Steiner problem is a tree  $T$  connecting all required vertices, utilising links with a sufficient throughput and using as few links as possible. The vertices of the tree  $T$  represent trusted nodes of the newly created QKDN backbone, and the edges are QKD links between them. Each trusted node contains as many QKD modules as is the degree of the appropriate vertex. We are aware that such a topology does not contain loops, so there is no link redundancy, and it may not be optimal; however, it serves as a good starting point for the backbone planning when the costs matter. The QKDN can be further expanded along the unused links in the optical network, eventually forming a loop.

## ■ 5.2.2 Metropolitan and access network

The metropolitan, or metro network for short, is a network spanning a city and the surrounding area. It serves as a middle point between the backbone and the access network. The range limitation of a QKD link in a metro area is a factor we no longer have to consider much. The distances between the trusted nodes are relatively small as there are multiple trusted nodes from which client access networks connect. A star topology consisting of one central node would undoubtedly save some construction costs. However, such a point represents a single point of failure for all clients in the metro area, rendering the network not resilient and unsuitable for expansion. For a recommended topology, we stick to the ring topology, proving as cost-efficient as proposed in [80]. Again, it is advantageous to reuse the already installed optical core network; however, in the city area, building new dedicated optical lines is much easier as there






of the optical fibres for the quantum and classical channels must be measured to comply with the required quality. The device is calibrated correctly to ensure accurate operation. If a key management system is not integrated within the QKD module, it shall be connected and contained in close proximity to the module. The following is the connection between the QKDN controller and the QKDN manager. All these devices must be configured to communicate with each other, including providing authentication information such as certificates or pre-shared keys, depending on the authentication mechanism. The setup must be functionally tested before the start of the live operation to ensure it generates secure keys as expected. Once the interoperability verification from the QKDN network provider is successfully completed, the node will be fully operational.

## 5.3 Operation

Quantum key distribution networks introduce a whole new technological stack, which has to be maintained and operated by network engineers and administrators. This section outlines operational procedures and critical areas that must be considered when operating a QKDN. Adhering to these procedures decreases the risks of degrading the security and reliability of the network.

- 1) **User training:** It should be ensured that all the personnel operating on a QKDN receive comprehensive training. Because these are security-sensitive systems, the personnel must be aware of the consequences a security breach may have on a customer. The training includes configuration and operation training from a manufacturer of a QKD module, familiarisation with potential security incidents, troubleshooting training and user account management.
- 2) **Operational monitoring:** QKDN is a distributed system which reacts and can recover from various situations, such as topological changes or interruption of communication with a node. The whole infrastructure must be monitored as the QKDN manager continuously analyses status information and operational logs collected from all components in the network, as described in Section 2.4. That allows for timely fault or overload identification.
- 3) **Maintenance:** All hardware, including the optical links, must be regularly maintained to ensure long-term reliability. The QKD module manufacturer shall prescribe the maintenance schedule.
- 4) **Security management:** A security management team is responsible for risk management, as described in Section 3.2. It creates and maintains security policies and emergency security procedures. These are applied in case of an incident to ensure quick resolution and recovery. Lastly, they are responsible for keeping documentation of such incidents.
- 5) **Compliance and auditing:** Critical infrastructure like QKDN is expected to be under strict regulation. Compliance with not only standards and security regulations but also laws is required. Regular audits are carried out to verify compliance by identifying configuration errors and deficiencies in security practices, which are minimised by applying procedures stated in Section 3.1.





# Chapter 6

## Conclusion

This thesis introduced and motivated the creation of quantum key distribution networks. The primary focus was on security, as the security of other systems relies just on quantum networks. We suggested several enhancements to expedite the deployment of these networks, making them more appealing for general use.

To motivate quantum key distribution, we thoroughly explained the problem of today's widely used asymmetric cryptography by explaining the mathematical problems it relies on in Section 1.1 and why it is deficient for the future. Several examples of affected areas were provided to demonstrate the vast scale of the issue. Analysis of the Transport layer security protocol in Section 1.2 showed how encrypted communication is vulnerable now. Post-quantum cryptography and quantum key distribution are possible solutions, the latter being our point of interest in the rest of the thesis.

Chapter 2 illustrates the principles of quantum key distribution by analysing real-world cases and providing an overview of current standardisation efforts in Sections 2.1 and 2.2. We described the design of a quantum key distribution network architecture according to the newest standards, dividing it into functional layers in Section 2.3. Next, we explained how the individual layers function and distribute an encryption key through the network in Section 2.4.

Having established the architectural details, we focused on security. We clarified the objectives and principles of information security (Section 3.1) as well as the Common Criteria evaluation model (Section 3.3), which is recognised worldwide and plays a crucial role in certifying the security of computer systems and devices. Because components of quantum key distribution networks must be trustworthy, we contributed by writing a missing protection profile for a key management system for a quantum key distribution network in Section 3.7. We identified vulnerabilities by characterising users of the network in Section 3.4 and attack surfaces and quantum hacking in Section 3.6. In the security analysis, the threats were enumerated in Section 3.5 and further analysed in the protection profile and Section 4.3.1.

In Chapter 4, we suggested several improvements for the quantum networks, such as increasing their overall security so that they can be deployed for real-world applications. Specifically by differentiating ways to deliver secure keys to the user in Section 4.1 and enhancing security practice by combining keys from a quantum key distribution with other key exchange methods in Section





## References

- [1] RIVEST, R. L., A. SHAMIR, and L. ADLEMAN. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM*. New York, NY, USA: Association for Computing Machinery, feb, 1978, Vol. 21, No. 2, pp. 120–126. ISSN 0001-0782. Available from DOI 10.1145/359340.359342. Available from <https://doi.org/10.1145/359340.359342>.
- [2] ELGAMAL, T.. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*. 1985, Vol. 31, No. 4, pp. 469-472. Available from DOI 10.1109/TIT.1985.1057074.
- [3] DIFFIE, W., and M. HELLMAN. New directions in cryptography. *IEEE Transactions on Information Theory*. 1976, Vol. 22, No. 6, pp. 644-654. Available from DOI 10.1109/TIT.1976.1055638.
- [4] ARORA, S., and B. BARAK. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006. ISBN 978-0-521-42426-4. Available from <https://theory.cs.princeton.edu/complexity/book.pdf>.
- [5] BUHLER, J. P., H. W. LENSTRA, and Carl POMERANCE. Factoring integers with the number field sieve. In: Arjen K. LENSTRA, and Hendrik W. LENSTRA, eds. *The development of the number field sieve*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. pp. 50–94. ISBN 978-3-540-47892-8.
- [6] HOFFSTEIN, J., J. PIPHER, and J.H. SILVERMAN. *An Introduction to Mathematical Cryptography*. New York, NJ, USA: Springer, 2008. Undergraduate Texts in Mathematics. Available from DOI 10.1007/978-0-387-77993-5. Available from <http://books.google.com/books?id=62PYj63QalkC>.
- [7] ADLEMAN, Leonard M, and Jonathan DEMARRAIS. A subexponential algorithm for discrete logarithms over all finite fields. *Mathematics of Computation*. 1993, Vol. 61, No. 203, pp. 1–15.
- [8] RESCORLA, Eric, and Tim DIERKS. *The Transport Layer Security (TLS) Protocol Version 1.2* [RFC 5246]. Available from DOI 10.17487/RFC5246. Available from <https://www.rfc-editor.org/info/rfc5246>.
- [9] RESCORLA, Eric. *The Transport Layer Security (TLS) Protocol Version 1.3* [RFC 8446]. Available from DOI 10.17487/RFC8446. Available from <https://www.rfc-editor.org/info/rfc8446>.
- [10] MORIARTY, Kathleen, and Stephen FARRELL. *Deprecating TLS 1.0 and TLS 1.1* [RFC 8996]. Available from DOI 10.17487/RFC8996. Available from <https://www.rfc-editor.org/info/rfc8996>.

- [11] KRAWCZYK, Dr. Hugo, and Pasi ERONEN. *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)* [RFC 5869]. Available from DOI 10.17487/RFC5869. Available from <https://www.rfc-editor.org/info/rfc5869>.
- [12] DOWLING, Benjamin, Marc FISCHLIN, Felix GNANTHER, AND DOUGLAS STEBILA. A CRYPTOGRAPHIC ANALYSIS OF THE TLS 1.3 HANDSHAKE PROTOCOL. *JOURNAL OF CRYPTOLOGY*. SPRINGER, 2021, VOL. 34, NO. 4, PP. 37.
- [13] BENIOFF, Paul. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of statistical physics*. Springer, 1980, Vol. 22, pp. 563–591.
- [14] MANIN, Yuri. Computable and uncomputable. *Sovetskoye Radio, Moscow*. 1980, Vol. 128, pp. 15.
- [15] FEYNMAN, Richard P.. Simulating physics with computers. *International Journal of Theoretical Physics*. Jun, 1982, Vol. 21, No. 6, pp. 467–488. ISSN 1572-9575. Available from DOI 10.1007/BF02650179. Available from <https://doi.org/10.1007/BF02650179>.
- [16] NIELSEN, Michael A., and Isaac L. CHUANG. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. ISBN 9781107002173. Available from DOI <https://doi.org/10.1017/CBO9780511976667>.
- [17] DEUTSCH, David, and Richard JOZSA. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*. 1992, Vol. 439, No. 1907, pp. 553–558. Available from DOI 10.1098/rspa.1992.0167. Available from <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167>.
- [18] SHOR, Peter W.. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994. pp. 124–134. Available from DOI 10.1109/SFCS.1994.365700.
- [19] PRESKILL, John. Quantum Computing in the NISQ era and beyond. *Quantum*. Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, aug, 2018, Vol. 2, pp. 79. ISSN 2521-327X. Available from DOI 10.22331/q-2018-08-06-79. Available from <http://dx.doi.org/10.22331/q-2018-08-06-79>.
- [20] BENNETT, Charles H., Ethan BERNSTEIN, Gilles BRASSARD, and Umesh VAZIRANI. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*. Society for Industrial Applied Mathematics (SIAM), oct, 1997, Vol. 26, No. 5, pp. 1510–1523. ISSN 1095-7111. Available from DOI 10.1137/s0097539796300933. Available from <http://dx.doi.org/10.1137/S0097539796300933>.
- [21] SHOR, Peter W.. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*. Society for Industrial Applied Mathematics (SIAM), oct, 1997, Vol. 26, No. 5, pp. 1484–1509. ISSN 1095-7111. Available from DOI

- 10.1137/s0097539795293172. Available from <http://dx.doi.org/10.1137/S0097539795293172>.
- [22] RISTÈ, D., S. POLETTI, M.-Z. HUANG, A. BRUNO, V. VESTERINEN, O.-P. SAIRA, and L. DICARLO. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nature Communications*. Springer Science and Business Media LLC, apr, 2015, Vol. 6, No. 1. ISSN 2041-1723. Available from DOI 10.1038/ncomms7983. Available from <http://dx.doi.org/10.1038/ncomms7983>.
- [23] GIDNEY, Craig, and Martin EKERÅ. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*. Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, apr, 2021, Vol. 5, pp. 433. ISSN 2521-327X. Available from DOI 10.22331/q-2021-04-15-433. Available from <http://dx.doi.org/10.22331/q-2021-04-15-433>.
- [24] GOUZIEN, Élie, and Nicolas SANGOUARD. Factoring 2048-bit RSA Integers in 177 Days with 13 436 Qubits and a Multimode Memory. *Physical Review Letters*. American Physical Society (APS), sep, 2021, Vol. 127, No. 14. ISSN 1079-7114. Available from DOI 10.1103/physrevlett.127.140503. Available from <http://dx.doi.org/10.1103/PhysRevLett.127.140503>.
- [25] BERNSTEIN, Daniel J., Johannes BUCHMANN, and Erik DAHMEN. *Post Quantum Cryptography*. 1st ed. Springer Publishing Company, Incorporated, 2008. ISBN 978-3-540-88702-7. Available from DOI <https://doi.org/10.1007/978-3-540-88702-7>.
- [26] Available from <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- [27] DINUR, I., G. KINDLER, and S. SAFRA. Approximating-CVP to within almost-polynomial factors is NP-hard. In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. 1998. pp. 99-109. Available from DOI 10.1109/SFCS.1998.743433.
- [28] AJTAI, Mikl  
s. The shortest vector problem in L2 is NP-hard for randomized reductions. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998. pp. 10-19.
- [29] PIRANDOLA, S., U. L. ANDERSEN, L. BANCHI, M. BERTA, D. BUNANDAR, R. COLBECK, D. ENGLUND, T. GEHRING, C. LUPO, C. OTTAVIANI, J. L. PEREIRA, M. RAZAVI, J. SHAMSUL SHAARI, M. TOMAMICHEL, V. C. USENKO, G. VALLONE, P. VILLORESI, and P. WALLDEN. Advances in quantum cryptography. *Advances in Optics and Photonics*. Optica Publishing Group, dec, 2020, Vol. 12, No. 4, pp. 1012. ISSN 1943-8206. Available from DOI 10.1364/aop.361502. Available from <http://dx.doi.org/10.1364/AOP.361502>.
- [30] BRASSARD, G.. Brief history of quantum cryptography: a personal perspective. In: *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005*. IEEE, 2005. Available from DOI 10.1109/itwtpi.2005.1543949. Available from <http://dx.doi.org/10.1109/ITWTPI.2005.1543949>.

- [31] BENNETT, Charles H, and Gilles BRASSARD. Quantum cryptography and its application to provably secure key expansion, public-key distribution, and coin-tossing. In: *IEEE International Symposium on Information Theory*. 1983.
- [32] ELLIOTT, Chip. *The DARPA Quantum Network*.
- [33] ELLIOTT, Chip, Alexander COLVIN, David PEARSON, Oleksiy PIKALO, John SCHLAFER, and Henry YEH. Current status of the DARPA quantum network. In: *Quantum Information and computation III*. 2005. pp. 138–149.
- [34] METER, Rodney Van. *Quantum Networking*. John Wiley Sons, Incorporated, 2014. ISBN 9781118648926.
- [35] POPPE, A., M. PEEV, and O. MAURHART. Outline of the SECOQC Quantum-Key-Distribution Network in Vienna. *International Journal of Quantum Information*. World Scientific Pub Co Pte Lt, apr, 2008, Vol. 06, No. 02, pp. 209–218. ISSN 1793-6918. Available from DOI 10.1142/s0219749908003529. Available from <http://dx.doi.org/10.1142/S0219749908003529>.
- [36] SASAKI, M., M. FUJIWARA, H. ISHIZUKA, W. KLAUS, K. WAKUI, M. TAKEOKA, S. MIKI, T. YAMASHITA, Z. WANG, A. TANAKA, K. YOSHINO, Y. NAMBU, S. TAKAHASHI, A. TAJIMA, A. TOMITA, T. DOMEKI, T. HASEGAWA, Y. SAKAI, H. KOBAYASHI, T. ASAI, K. SHIMIZU, T. TOKURA, T. TSURUMARU, M. MATSUI, T. HONJO, K. TAMAKI, H. TAKESUE, Y. TOKURA, J. F. DYNES, A. R. DIXON, A. W. SHARPE, Z. L. YUAN, A. J. SHIELDS, S. UCHIKOGA, M. LEGRÉ, S. ROBYR, P. TRINKLER, L. MONAT, J.-B. PAGE, G. RIBORDY, A. POPPE, A. ALLACHER, O. MAURHART, T. LÄNGER, M. PEEV, and A. ZEILINGER. Field test of quantum key distribution in the Tokyo QKD Network. *Optics Express*. The Optical Society, may, 2011, Vol. 19, No. 11, pp. 10387. ISSN 1094-4087. Available from DOI 10.1364/oe.19.010387. Available from <http://dx.doi.org/10.1364/OE.19.010387>.
- [37] ITU-T. *Recommendation ITU-T Y.3800, Overview on networks supporting quantum keydistribution*. Available from <https://handle.itu.int/11.1002/1000/14257>.
- [38] SHANNON, C. E.. Communication theory of secrecy systems. *The Bell System Technical Journal*. 1949, Vol. 28, No. 4, pp. 656-715. Available from DOI 10.1002/j.1538-7305.1949.tb00928.x.
- [39] LIAO, Sheng-Kai, Wen-Qi CAI, Johannes HANDSTEINER, Bo LIU, Juan YIN, Liang ZHANG, Dominik RAUCH, Matthias FINK, Ji-Gang REN, Wei-Yue LIU, Yang LI, Qi SHEN, Yuan CAO, Feng-Zhi LI, Jian-Feng WANG, Yong-Mei HUANG, Lei DENG, Tao XI, Lu MA, Tai HU, Li LI, Nai-Le LIU, Franz KOIDL, Peiyuan WANG, Yu-Ao CHEN, Xiang-Bin WANG, Michael STEINDORFER, Georg KIRCHNER, Chao-Yang LU, Rong SHU, Rupert URSIN, Thomas SCHEIDL, Cheng-Zhi PENG, Jian-Yu WANG, Anton ZEILINGER, and Jian-Wei PAN. Satellite-Relayed Intercontinental Quantum Network. *Phys. Rev. Lett.* American Physical Society, Jan, 2018, Vol. 120, pp. 030501. Available from DOI 10.1103/PhysRevLett.120.030501. Available from <https://link.aps.org/doi/10.1103/PhysRevLett.120.030501>.



- [40] CAO, Yuan, Yongli ZHAO, Qin WANG, Jie ZHANG, Soon Xin NG, and Lajos HANZO. The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet. *IEEE Communications Surveys Tutorials*. 2022, Vol. 24, No. 2, pp. 839-894. Available from DOI 10.1109/COMST.2022.3144219.
- [41] LO, Hoi-Kwong, Marcos CURTY, and Bing QI. Measurement-Device-Independent Quantum Key Distribution. *Physical Review Letters*. American Physical Society (APS), mar, 2012, Vol. 108, No. 13. ISSN 1079-7114. Available from DOI 10.1103/physrevlett.108.130503. Available from <http://dx.doi.org/10.1103/PhysRevLett.108.130503>.
- [42] LUCAMARINI, M., Z. L. YUAN, J. F. DYNES, and A. J. SHIELDS. Overcoming the rate–distance limit of quantum key distribution without quantum repeaters. *Nature*. Springer Science and Business Media LLC, may, 2018, Vol. 557, No. 7705, pp. 400–403. ISSN 1476-4687. Available from DOI 10.1038/s41586-018-0066-6. Available from <http://dx.doi.org/10.1038/s41586-018-0066-6>.
- [43] CHEN, Jiu-Peng, Chi ZHANG, Yang LIU, Cong JIANG, Wei-Jun ZHANG, Zhi-Yong HAN, Shi-Zhao MA, Xiao-Long HU, Yu-Huai LI, Hui LIU, Fei ZHOU, Hai-Feng JIANG, Teng-Yun CHEN, Hao LI, Li-Xing YOU, Zhen WANG, Xiang-Bin WANG, Qiang ZHANG, and Jian-Wei PAN. Twin-field quantum key distribution over a 511 km optical fibre linking two distant metropolitan areas. *Nature Photonics*. Springer Science and Business Media LLC, jun, 2021, Vol. 15, No. 8, pp. 570–575. ISSN 1749-4893. Available from DOI 10.1038/s41566-021-00828-5. Available from <http://dx.doi.org/10.1038/s41566-021-00828-5>.
- [44] BENNETT, Charles H., and Gilles BRASSARD. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*. Elsevier BV, dec, 2014, Vol. 560, pp. 7–11. ISSN 0304-3975. Available from DOI 10.1016/j.tcs.2014.05.025. Available from <http://dx.doi.org/10.1016/j.tcs.2014.05.025>.
- [45] EKERT, Artur K.. Quantum cryptography based on Bell’s theorem. *Phys. Rev. Lett.* American Physical Society, Aug, 1991, Vol. 67, pp. 661–663. Available from DOI 10.1103/PhysRevLett.67.661. Available from <https://link.aps.org/doi/10.1103/PhysRevLett.67.661>.
- [46] BENNETT, Charles H., Gilles BRASSARD, and N. David MERMIN. Quantum cryptography without Bell’s theorem. *Phys. Rev. Lett.* American Physical Society, Feb, 1992, Vol. 68, pp. 557–559. Available from DOI 10.1103/PhysRevLett.68.557. Available from <https://link.aps.org/doi/10.1103/PhysRevLett.68.557>.
- [47] ITU-T. *Recommendation ITU-T Y.3803, Quantum key distribution networks – Key management*. Available from <https://handle.itu.int/11.1002/1000/14408>.
- [48] ITU-T. *Recommendation ITU-T X.1714, Key combination and confidential key supply for quantum key distribution networks*.
- [49] ITU-T. *Recommendation ITU-T Y.3801, Functional requirements for quantum key distribution networks*. Available from <https://handle.itu.int/11.1002/1000/14408>.

- [50] ITU-T. *Recommendation ITU-T Y.3804, Quantum key distribution networks – Control and management*. Available from <https://handle.itu.int/11.1002/1000/14409>.
- [51] ISO/IEC. *Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 1: Introduction and general model*. ISO/IEC 15408-1:2022, 2022.
- [52] ETSI. *Quantum Key Distribution (QKD); Common Criteria Protection Profile - Pair of Prepare and Measure Quantum Key Distribution Modules*. ETSI GS QKD 016, 2023.
- [53] ISO/IEC. *Information security, cybersecurity and privacy protection - Information security management systems*. ISO/IEC 27001:2022, 2022.
- [54] ISO. *Risk management - Guidelines*. ISO 31000:2018, 2018.
- [55] KATSIKAS, Sokratis K.. *Chapter 34 - Risk Management*. Available from DOI <https://doi.org/10.1016/B978-0-12-803843-7.00034-X>. Available from <https://www.sciencedirect.com/science/article/pii/B978012803843700034X>.
- [56] WILLIAMS, Jeff. *OWASP Risk Rating Methodology* [[https://owasp.org/www-community/OWASP\\_Risk\\_Rating\\_Methodology](https://owasp.org/www-community/OWASP_Risk_Rating_Methodology)]. Accessed: 2-4-2024.
- [57] ISO. *Risk management - Vocabulary*. ISO Guide 73:2009, 2009.
- [58] ISO/IEC. *Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 2: Security functional components*. ISO/IEC 15408-2:2022, 2022.
- [59] KOHNFELDER, Loren, and Praerit GARG. The threats to our products. *Microsoft Interface, Microsoft Corporation*. 1999, Vol. 33.
- [60] ITU-T. *Recommendation ITU-T X.1710, Security framework for quantum key distribution networks*.
- [61] LELLA, Eufemia, and Giovanni SCHMID. On the Security of Quantum Key Distribution Networks. *Cryptography*. 2023, Vol. 7, No. 4. ISSN 2410-387X. Available from DOI 10.3390/cryptography7040053. Available from <https://www.mdpi.com/2410-387X/7/4/53>.
- [62] MARQUARDT, Christoph, Ulrich SEYFARTH, Sven BETTENDORF, Martin BOHMANN, Alexander BUCHNER, Marcos CURTY, Dominique ELSER, Silas EUL, Tobias GEHRING, Nitin JAIN, Thomas KLOCKE, Marie REINECKE, Nico SIEBER, Rupert URSIN, Marc WEHLING, and Henning WEIER. Implementation Attacks against QKD Systems. *dec*, 2023, No. 575, pp. 171. Available from [https://www.bsi.bund.de/EN/Service-Navi/Publikationen/Studien/QKD-Systems/Implementation\\_Attacks\\_QKD\\_Systems\\_node.html](https://www.bsi.bund.de/EN/Service-Navi/Publikationen/Studien/QKD-Systems/Implementation_Attacks_QKD_Systems_node.html).
- [63] JAIN, Nitin, Birgit STILLER, Imran KHAN, Dominique ELSER, Christoph MARQUARDT, and Gerd LEUCHS. Attacks on practical quantum key distribution systems (and how to prevent them). *Contemporary Physics*. Informa UK Limited, mar, 2016, Vol. 57, No. 3, pp. 366–387. ISSN 1366-5812. Available from DOI 10.1080/00107514.2016.1148333. Available from <http://dx.doi.org/10.1080/00107514.2016.1148333>.

- [64] ISO/IEC. *Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 3: Security assurance components*. ISO/IEC 15408-3:2022, 2022.
- [65] ISO/IEC. *Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 5: Pre-defined packages of security requirements*. ISO/IEC 15408-5:2022, 2022.
- [66] ETSI. *Quantum Key Distribution (QKD); Application Interface*. ETSI GS QKD 004, 2020.
- [67] ETSI. *Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API*. ETSI GS QKD 014, 2019.
- [68] WEGMAN, Mark N., and J. Lawrence CARTER. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*. 1981, Vol. 22, No. 3, pp. 265-279. ISSN 0022-0000. Available from DOI [https://doi.org/10.1016/0022-0000\(81\)90033-7](https://doi.org/10.1016/0022-0000(81)90033-7). Available from <https://www.sciencedirect.com/science/article/pii/0022000081900337>.
- [69] BOS, Joppe, Leo DUCAS, Eike KILTZ, T LEPOINT, Vadim LYUBASHEVSKY, John M. SCHANCK, Peter SCHWABE, Gregor SEILER, and Damien STEHLE. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: *2018 IEEE European Symposium on Security and Privacy (EuroSP)*. 2018. pp. 353-367. Available from DOI 10.1109/EuroSP.2018.00032.
- [70] PETCHER, Adam, and Matthew CAMPAGNA. Security of hybrid key establishment using concatenation. *The International Association for Cryptologic Research (IACR)*. 2020. Available from <https://www.amazon.science/publications/security-of-hybrid-key-establishment-using-concatenation>.
- [71] HERZBERG, Amir. *Folklore, Practice and Theory of Robust Combiners* [Cryptology ePrint Archive, Paper 2002/135]. Available from <https://eprint.iacr.org/2002/135>. <https://eprint.iacr.org/2002/135>.
- [72] DIANATI, Mehrdad, Romain ALLÉAUME, Maurice GAGNAIRE, and Xuemin (Sherman) SHEN. Architecture and protocols of the future European quantum key distribution network. *Security and Communication Networks*. 2008, Vol. 1, No. 1, pp. 57-74. Available from DOI <https://doi.org/10.1002/sec.13>. Available from <https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.13>.
- [73] ITU-T. *Recommendation ITU-T Y.3810, Quantum key distribution network interworking - Framework*. Available from <https://handle.itu.int/11.1002/1000/15063>.
- [74] GUNKEL, Matthias, Felix WISSEL, and Andreas POPPE. Designing a Quantum Key Distribution Network - Methodology and Challenges. In: *Photonic Networks; 20th ITG-Symposium*. 2019. pp. 1-3.
- [75] WENNING, Mario, Sai Kireet PATRI, Tobias FEHENBERGER, and Carmen MAS-MACHUCA. Optimized Deployment and Routing Strategies for QKD and DWDM Networks. In: *Photonic Networks; 24th ITG-Symposium*. 2023. pp. 1-6.

- [76] BEBROV, Georgi. On the (relation between) efficiency and secret key rate of QKD. *Scientific Reports*. feb, 2024, Vol. 14, No. 1, pp. 3638.
- [77] DREYFUS, S. E., and R. A. WAGNER. The steiner problem in graphs. *Networks*. 1971, Vol. 1, No. 3, pp. 195-207. Available from DOI <https://doi.org/10.1002/net.3230010302>. Available from <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230010302>.
- [78] KARP, Richard M.. Reducibility among Combinatorial Problems. In: Raymond E. MILLER, James W. THATCHER, and Jean D. BOHLINGER, eds. *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*. Boston, MA: Springer US, 1972. pp. 85-103. ISBN 978-1-4684-2001-2. Available from DOI 10.1007/978-1-4684-2001-2\_9. Available from [https://doi.org/10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9).
- [79] KOU, L, G MARKOWSKY, and L BERMAN. A fast algorithm for Steiner trees. *Acta Informatica*. jun, 1981, Vol. 15, No. 2, pp. 141-145.
- [80] WANG, Hua, Yongli ZHAO, Avishek NAG, Xiaosong YU, Xinyi HE, and Jie ZHANG. End-to-End Quantum Key Distribution (QKD) from Metro to Access Networks. In: *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*. 2020. pp. 1-5. Available from DOI 10.1109/DRCN48652.2020.1570611062.

# Appendix A

## Abbreviations

AEAD	■	Authenticated Encryption with Associated Data
AES	■	Advanced Encryption Standard
CC	■	Common Criteria
CIA	■	Confidentiality, Integrity, Availability
DH	■	Diffie–Hellman
DLP	■	Discrete Logarithm Problem
ECDSA	■	Elliptic-curve Digital Signature Algorithm
GWF	■	Gateway Function
GWN	■	Gateway Node
HKDF	■	HMAC-based Key Derivation Function
HMAC	■	Hash-based Message Authentication Code
IFP	■	Integer Factorisation Problem
IWN	■	Interworking Node
KEM	■	Key Encapsulation Mechanism
KM	■	Key Management
KMA	■	Key Management Agent
KSA	■	Key Supply Agent
NIST	■	National Institute of Standards and Technology
OTP	■	One-time Pad
PDCA	■	Plan–Do–Check–Act
PP	■	Protection Profile
PQC	■	Post-quantum Cryptography
QBER	■	Quantum Bit Error Rate
QKD	■	Quantum Key Distribution
QKDN	■	Quantum Key Distribution Network
QRNG	■	Quantum Random Number Generator
RSA	■	Rivest, Shamir, Adleman
SAR	■	Security Assurance Requirement
SFP	■	Security Functional Policy
SFR	■	Security Functional Requirement
SKR	■	Secure Key Generation Rate
SSL	■	Secure Sockets Layer
ST	■	Security Target
TCP	■	Transmission Control Protocol
TLS	■	Transport Layer Security
TOE	■	Target of Evaluation
TSF	■	TOE Security Functionality