NANCY

An Artificial Intelligent Aided Unified Network for Secure Beyond 5G Long Term Evolution [GA: 101096456]

Deliverable 5.1

Quantum Safety Mechanisms

Programme: HORIZON-JU-SNS-2022-STREAM-A-01-06

Start Date: 01 January 2023

Duration: 36 Months

Co-funded by the European Union

6GSNS

NANCY project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096456.

Document Control Page

Deliverable Name	Quantum Safety Mechanisms
Deliverable Number	5.1
Work Package	WP5
Associated Task	T5.1 Quantum Safety Mechanisms
Dissemination Level	Public
Due Date	31 October 2024 (M22)
Completion Date	31 October 2024
Submission Date	31 October 2024
Deliverable Lead Partner	TDIS
Deliverable Author(s)	Jean-Paul Truong (TDIS), Stylianos Trevlakis (INNO), Theodoros Tsiftsis (INNO), Vasileios Kouvakis (INNO), Lamprini Mitsiou (INNO), Giuseppe Celozzi (TEI), Marco Tambasco (TEI), Dimitrios Pliatsios (UOWM), Panagiotis Sarigiannidis (UOWM), Thomas Lagkas (UOWM), Athanasios Liatifis (UOWM), Sotirios Tegos (UOWM), Ioannis Makris (MINDS), Nikolaos Ntampakis (MINDS), Andreas Maropoulos (MINDS), Vasileios Gavresis (MINDS), Nikolaos Moschos (MINDS), Dimitrios-Christos Asimopoulos (MINDS)
Version	1.0

Document History

Version	Date	Change History	Author(s)	Organisation
0.1	05/07/2024	Initial version	Aitor Brazaola	TECN
0.2	09/10/2024	Addition to QKD section	Stylianos Trevlakis, Theodoros Tsiftsis, Vasileios Kouvakis, Lamprini Mitsiou	INNO
0.3	16/10/2024	PQC for Secure Communications added	Giuseppe Celozzi, Marco Tambasco	TEI
0.4	28/10/2024	Integration of ITL Review	Jean-Paul. Truong/ Aitor Brazaola/ Stylianos Trevlakis	TDIS/ TECN/ INNO
1.0	31/10/2024	Quality Checks & Revisions	Dimitrios Pliatsios, Panagiotis Sarigiannidis, Thomas Lagkas, Athanasios Liatifis, Sotirios Tegos / Ioannis Makris, Nikolaos Ntampakis, Andreas Maropoulos, Vasileios Gavresis, Nikolaos Moschos,	UOWM / MINDS

			Dimitrios-Christos Asimopoulos	

Internal Review History

Name	Organisation	Date
Antonella Clavenna	ITL	24 October 2024
F. Javier Vicente	NEC	28 October 2024

Quality Manager Revision

Name	Organisation	Date
Anna Triantafyllou, Dimitrios Pliatsios	UOWM	31 October 2024

Legal Notice

The information in this document is subject to change without notice.

The Members of the NANCY Consortium make no warranty of any kind about this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The Members of the NANCY Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SNS JU. Neither the European Union nor the SNS JU can be held responsible for them.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Explanation
ANSSI	Agence nationale de la sécurité des systèmes d'information
API	Application Programming Interface
BS	Base Station
BSI	Bundesamt für Sicherheit in der Informationstechnik
COW	Coherent-One-Way
CV	Continuous Variable
DD	Data Detector
DPR	Distributed-Phase-Reference
DPS	Differential Phase Shift
DV	Discrete Variable
ECC	Elliptic curve cryptography
ETSI	European Telecommunications Standards Institute
FIPS	Federal Information Processing Standards
GUI	Graphical User Interface
KEM	Key Encapsulation Mechanism
KME	Key Management Entity
KMS	Key Management System
LMS	Leighton-Micali Hash-Based Signatures
MD	Monitoring Detector
NIST	National Institute of Standards and Technology
PFX	Personal Information Exchange
PKCS	Public Key Cryptographic Standards
PNS	Photon Number Splitting
PQC	Post-Quantum Cryptography
QBER	Quantum Bit Error Rate
QKD	Quantum Key Distribution
QMS	Quantum Management System
RSA	Rivest-Shamir-Adleman
SAE	Secure Application Entity
UE	User Equipment
WDM	Wavelength Division Multiplexing

Executive summary

In an era of increasing reliance on digital communications and the rapid evolution of quantum technologies, ensuring the security and integrity of data has become paramount. This report presents a comprehensive analysis of Quantum Key Distribution (QKD) and Post-Quantum Cryptography (PQC) technologies, highlighting their significance in safeguarding sensitive information against potential quantum threats.

QKD Technologies: The report delves into the principles and protocols of QKD, which utilize quantum mechanics to enable secure key exchange between parties. Prominent protocols, such as BB84 and E91, are discussed, along with deployment methods like fibre-optic and free-space QKD. Notably, advancements in satellite-based QKD signal a transformative approach to global secure communication. The integration of QKD into telecommunications networks, particularly in 5G infrastructures, showcases its potential to enhance security profiles for internal communications, including fronthaul connections between base stations.

PQC Technologies: As quantum computing advances, traditional cryptographic methods face vulnerabilities. The report outlines the necessity of developing quantum-resistant algorithms to maintain secure communications. The NIST standardization process plays a critical role in this landscape, with several promising candidates now being refined for implementation. Key algorithms are categorized into families, including lattice-based, code-based, and hash-based cryptography, each with unique advantages and challenges. Importantly, PQC also serves to ensure the integrity of blockchain technology, safeguarding decentralized systems from quantum attacks.

Experimental Implementations: The report details the architectural design and implementation of QKD in the context of a 5G Base Station demonstrator, demonstrating practical applications of QKD in enhancing the security of telecommunication infrastructures. Additionally, the development of a PQC Digital Signature Solution by Thales DIS illustrates the ongoing efforts to integrate quantum-resistant algorithms into real-world applications.

This report is structured into clearly defined sections:

- A state-of-the-art analysis, including an introduction to the concepts and motivations behind QKD and PQC,
- Three sections dedicated to the QKD experimentations, the PQC Signature Solution and the PQC for Secure Communications,
- For each one of the above-dedicated sections, details are given about the technical foundations of the technologies and protocols and their applications. Experimental designs and implementations are examined next, showcasing real-world scenarios.
- The report concludes with insights and future directions for research and development in these critical areas.

In conclusion, as cyber threats evolve alongside advancements in quantum computing, QKD and PQC technologies emerge as critical components in securing the future of digital communications and protecting the integrity of blockchain systems. The findings of this report underscore the importance of continued research and development in these fields, as well as the need for collaborative efforts in standardization and implementation.

1. Introduction

The emergence of quantum computing represents a clear technology breakthrough in the landscape of information security, threatening the foundations of classical cryptography. With their strong increased capabilities, quantum computers will have the potential to break widely used cryptographic algorithms, such as RSA and ECC. Therefore, there is an urgent need to deploy quantum-resistant cryptographic solutions.

This report presents the NANCY project strategic initiative aiming to address these challenges through three approaches:

- Quantum Key Distribution (QKD): is the foreseen technology to address the resilience of secure communications in the quantum era. QKD is a technology that relies on quantum physics to secure the distribution of symmetric encryption keys offering thus a robust defence against potential threats. In our project, we designed a QKD experimental framework combined with a simulation. This setup not only demonstrates the practical integration of QKD technology but also evaluates its performance and reliability.
- Post-Quantum Cryptography (PQC) signature solution: PQC is a technology that relies on the robustness and resilience of cryptographic algorithms designed to resist potential threats in the quantum era. The development of a PQC digital signature solution is critical for ensuring data integrity and authenticity. The TDIS PQC Signature Token, a core component of this solution, integrates a PQC signature algorithm within a tamperproof hardware environment. This token serves various applications, including identity verification and corporate security, by utilizing asymmetric key pairs for data signing. Our project follows the ongoing NIST initiative to standardize quantum-resistant algorithms, ensuring our solution is aligned with global security standards.
- PQC for secure communications: the implementation of PQC for secure communications focuses on integrating PQC algorithms into existing communication protocols. By adopting these algorithms, we can enhance the resilience of data transmitted over networks. Purpose of the Document

1.1. Relation to Other Tasks and Deliverables

This deliverable is associated with Task 5.2, where the PQC Signature Solution will be integrated to ensure the resilience, integrity, and authenticity of Blockchains, as well as with Task 6.8 which is focused on the Italian outdoor demonstrator. Specifically, PQC will be leveraged for securing the communications among the devices.

1.2. Structure of the Document

The rest of the document is structured as follows:

- **Section 2 – State-of-the-art Analysis** presents the state-of-art associated with QKD and PQC technologies.
- **Section 3 – QKD Experimentations** documents the QKD experimental scenarios that were carried out, by outlining the architectural design, the interfaces, and the deployment details.
- **Section 4 – PQC Signature** presents the PQC signature solution that was developed. Specifically, it provides details on the architectural design, communication interfaces, and testing and deployment approaches.

- **Section 5 – PQC for Secure Communications** documents the leverage of the PQC signature solution for enhancing communication security.
- **Section 6 – Conclusion** summarizes and concludes the deliverable.

2. State-of-the-art Analysis

2.1. QKD Technologies

Quantum Key Distribution (QKD) is a secure communication method that uses quantum mechanics principles to exchange encryption keys between two parties. Unlike classical encryption methods, QKD leverages the properties of quantum particles, such as photons, to ensure that any attempt to eavesdrop on the key exchange can be detected. This is because measuring a quantum system inevitably disturbs it, alerting the communicating parties to the presence of an intruder. As a result, QKD provides a theoretically unbreakable method of key distribution, making it highly attractive for securing sensitive communications.

There are several types of QKD protocols, with the most well-known being the BB84 protocol [1], developed by Charles Bennett and Gilles Brassard in 1984. In this protocol, the sender (Alice) transmits photons polarized in one of four possible states, and the receiver (Bob) measures the polarization using randomly chosen bases. By comparing a subset of their measurements, Alice and Bob can detect eavesdropping and establish a shared secret key. Another notable protocol is the E91 protocol [2], proposed by Artur Ekert in 1991, which uses entangled photon pairs to achieve secure key distribution. The security of the E91 protocol is based on the fundamental principles of quantum entanglement and Bell’s theorem.

QKD can be deployed in various ways, depending on the communication infrastructure and requirements. Fibre-optic QKD is one common deployment method, where quantum keys are transmitted through optical fibres. This method is suitable for relatively short distances, typically up to a few hundred kilometres, due to the attenuation of photons in the fibre. To extend the range, trusted node networks can be used, where intermediate nodes relay the quantum keys while maintaining security. Another deployment method is free-space QKD, which involves transmitting quantum keys through the air or space. This method is useful for long-distance communication, such as between ground stations and satellites, as it avoids the limitations of fibre-optic attenuation.

In recent years, significant progress has been made in the development and deployment of QKD systems. Satellite-based QKD has emerged as a promising approach for global secure communication. For example, the Chinese satellite Micius has successfully demonstrated QKD over thousands of kilometres [3], paving the way for a global quantum communication network. Additionally, efforts are being made to integrate QKD with existing classical communication networks, enabling hybrid systems that combine the strengths of both quantum and classical technologies [4]. As research and development continue, QKD is expected to play a crucial role in secure communication, protecting sensitive information from increasingly sophisticated cyber threats.

Depending on the encoding techniques, there are three main QKD families of protocols [5]:

Discrete Variable (DV) QKD

DV-QKD protocols use individual photons in discrete quantum states to encode information. These states are typically represented using bases such as the polarization or phase of the photons. Some protocols of this family are:

- **BB84 Protocol**: Proposed by Bennett and Brassard in 1984 [1], it uses two sets of conjugate bases (e.g., rectilinear and diagonal polarization states).
- **E91 Protocol**: Developed by Ekert in 1991 [2], this protocol is based on entanglement and uses Bell's theorem to ensure security.

- **B92 Protocol**: Introduced by Bennett in 1992 [6], this simplified version of BB84 uses only two non-orthogonal states.

Continuous Variable (CV) QKD

CV-QKD encodes information using continuous quantum variables, such as the quadratures of the electromagnetic field. These protocols often rely on measuring quantum states using homodyne or heterodyne detection. Some protocols of this family are:

- **GG02 Protocol**: Named after its developers Grosshans and Grangier [7], it uses Gaussian-modulated coherent states and homodyne detection.
- **MSZ96 Protocol**: Bob measures the received states using randomly chosen quadrature phase amplitudes, introducing some uncertainty due to the nonorthogonal nature of the states. [8]

Distributed-Phase-Reference (DPR) QKD

DPR-QKD protocols encode information using the relative phase between successive pulses. They are known for their robustness against photon number splitting (PNS) attacks and other types of eavesdropping. Some protocols of this family are:

- **Coherent-One-Way (COW) Protocol**: This protocol uses sequences of coherent states and relies on the time of arrival and the phase of the pulses to distribute the key securely. It can be implemented using 3 or 4 states. [9]
- **Differential Phase Shift (DPS) Protocol**: This protocol encodes information in the phase differences between consecutive pulses, making it less sensitive to loss and errors than other QKD methods. [10]

A significant milestone in QKD technology was achieved with the long-distance QKD networks composed of independent Quantum links made by different vendors [11]. This advancement demonstrates the potential for establishing secure long-distance quantum communication networks, which could revolutionize data protection against cyber threats. However, the practical deployment of QKD still faces challenges, particularly in creating large-scale networks due to the limitations of current quantum light sources. Despite these challenges, the progress in QKD technology suggests a promising direction for the future of secure communication in the quantum computing era.

QKD is a technology that can potentially enforce the security of telecom and 5G networks [12]. However, its point-to-point nature limits its applicability within the network architecture. The infrastructure limitations attached to QKD make it hard to apply to the whole communications picture, especially for the user equipment. Instead, QKD can provide a high protection profile for internal backbone communications between fronthaul equipment. In some works, the implementation of QKD has been tied to protecting the communication link in some of the network functions between the Baseband Unit and the Radio Equipment Control [13] using the AES256 encryption algorithm, the results point that QKD can enable unconditional security in specific network topologies without a negative impact in terms of latencies and the underpinning quality of service.

On the other hand, the integration of QKD in existing telecommunications infrastructure is another field to explore, in works like [14] the authors study the performance of QKD applying Wavelength Division Multiplexing over pre-deployed optical fibres, although the results are promising, there are technical limitations in terms of spontaneous Raman scattering, four-wave mixing and amplified spontaneous emission.

In order to extend QKD deployment, simulators are an essential tool since, at this point, QKD equipment is hard to purchase in terms of prices and vendor availability. Although there are many open-source simulators to emulate the Quantum channel for physical studies purposes [15], there are not too many that put the focus on the classical part, which, in terms of network deployment and integration, is an essential tool for mocking how the standard defined ETSI interfaces can be streamlined with the existing communication mechanisms [16]. In addition, another missing point in many simulators is all the classical stack attached to every commercial QKD equipment, mainly the Key Management System, which is the software layer that the rest of the classical network infrastructure needs to deal with. This layer holds a cryptographic material key buffer to provide a continuous service of keys to Secure Application Entities.

In this work, there are two main contributions to the field of study. The first is an experimental simulation of Peer-to-peer communication between 5G base stations performed using actual QKD equipment over previously existing dark fibres with different distance variations and protocols. The main goal is to demonstrate how commercially available equipment can be integrated into novel B5G architectures, as proposed by NANCY, and which protocol configurations are more appropriate for certain tasks.

The other contribution is related to the QKD simulation field where a quite popular protocol in commercial equipment, like Coherent-One-Way, is implemented not only at the Quantum layer level but also including the classical KMS ETSI-014 interface to make it available to be integrated into realistic network environments without having to make strong financial investments purchasing QKD hardware. The provided simulator has been developed with the actual QKD equipment that mimics in mind, trying to make it as close as possible to reality. To achieve this, the development compared the same experimental parameters with the actual deployment in order to fine-tune the software to reproduce the same behaviour as commercial equipment from popular vendors.

2.2. PQC Technologies

Introduction to Post-Quantum Cryptography

Post-Quantum Cryptography (PQC) involves the development of cryptographic algorithms that are secure against potential threats posed by quantum computers. Classical cryptographic methods, such as RSA and ECC, are vulnerable to quantum attacks, particularly Shor’s algorithm [17]. The advancement of quantum computing necessitates the adoption of new cryptographic methods to maintain secure communications in the future.

Categories of Post-Quantum Algorithms

PQC algorithms are categorized based on their underlying mathematical principles:

- **Lattice-based Cryptography**: Algorithms such as Kyber and Dilithium rely on the hardness of lattice problems. Lattice-based schemes are highly regarded for their security and efficiency.
- **Code-based Cryptography**: Classic McEliece is a prominent example in this category, based on the difficulty of decoding random linear codes. These algorithms are recognized for their robustness but often require larger key sizes.
- **Hash-based Cryptography**: This category includes algorithms like SPHINCS+, which use hash functions to generate secure digital signatures. Hash-based cryptography is noted for its simplicity and security, though it may result in larger signature sizes.

- **Multivariate Quadratic Equations**: The Rainbow algorithm is a key representative, based on the difficulty of solving systems of multivariate quadratic equations.
- **Isogeny-based Cryptography**: This emerging area includes algorithms like SIKE (Supersingular Isogeny Key Encapsulation), though it has recently encountered security challenges.

NIST PQC Standardization Process

The PQC landscape features significant research and standardization efforts, particularly those led by the National Institute of Standards and Technology (NIST). NIST's ongoing process of evaluating and standardizing quantum-resistant cryptographic algorithms, initiated in 2016, has highlighted several promising candidates that are now being refined for broad adoption[1].

NIST's PQC standardization process has proceeded through multiple phases:

- **Round 1 (2017-2019):** NIST received over 60 algorithm submissions, initially evaluating their security, efficiency, and practicality.
- **Round 2 (2019-2020):** The pool was reduced to 26 candidates, with further analysis focusing on their security against both classical and quantum attacks.
- **Round 3 (2020-2022):** NIST narrowed the field to 15 finalists and 9 alternates, focusing on those with the greatest potential for standardization.
- **Post-Round 3 (2022-2024):** After Round 3, NIST selected several algorithms for standardization, resulting in the publication of FIPS203, FIPS204, and FIPS205, which provide comprehensive guidelines for implementing these algorithms.
- **Round 4 and the On-Ramp Process (>2024):** As quantum computing continues to evolve, NIST initiated Round 4 to further refine the selection of Public-key Encryption and Key-establishment Algorithms. This phase also introduces the "on-ramp" process, which allows new and previously evaluated algorithms to be considered for inclusion as PQC standards. The on-ramp process is designed to ensure that emerging cryptographic techniques and newly developed algorithms can be incorporated into the standardization framework. Notably, NIST is focusing on selecting additional digital signature schemes, expanding the range of standardized quantum-resistant signatures.

FIPS Publications in Post-Round 3

In the post-Round 3 phase, NIST released the following Federal Information Processing Standards (FIPS):

- **FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism (KEM) Standard**

FIPS 203 formalizes the adoption of module-lattice-based key-encapsulation mechanisms, specifically focusing on algorithms like CRYSTALS-Kyber. This standard provides technical specifications, security parameters, and implementation guidelines for these KEMs, which are essential for secure key exchange in a post-quantum world.

- **FIPS 204: Lattice-Based Digital Signatures**

FIPS 204 covers lattice-based digital signature algorithms, particularly CRYSTALS-Dilithium. This standard outlines the framework for using Dilithium in digital signatures, including processes for key

[1] Post-Quantum Cryptography | CSRC (nist.gov)

generation, signing, and verification. FIPS 204 aims to replace or complement existing digital signature standards such as those based on RSA or ECDSA.

- **FIPS 205: Stateless Hash-Based Digital Signature Standard**

FIPS 205 standardizes stateless hash-based digital signature algorithms, such as SPHINCS+. This standard provides guidelines for implementing these signatures, which rely on cryptographic hash functions and do not require the maintenance of state between signatures. FIPS 205 is particularly significant for applications requiring long-term security and resistance to quantum attacks, with a focus on minimizing the risks associated with stateful signature schemes.

NIST Security Levels

During the competition, a security classification has been defined. As shown in Table 1, The security levels have been set in comparison with strengths from classical algorithms:

Table 1: Security against both classical and quantum attacks

Level	Security Description
1	At least as hard to break as AES128
2	At least as hard to break as SHA256
3	At least as hard to break as AES192
4	At least as hard to break as SHA384
5	At least as hard to break as AES 256

Positions of BSI and ANSSI on PQC and Crypto Agility

Germany’s Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik - BSI) and France’s National Cybersecurity Agency (Agence nationale de la sécurité des systèmes d'information - ANSSI) have played active roles in shaping the Post-Quantum Cryptography landscape. Both agencies emphasize the critical importance of crypto agility—the ability to quickly switch to new cryptographic algorithms as needed in response to emerging threats or advancements in technology. BSI, while being cautious about adopting lattice-based cryptography due to concerns about long-term security and implementation challenges, stresses the need for systems that can adapt to new algorithms, particularly code-based cryptography like Classic McEliece. ANSSI, on the other hand, supports a balanced approach, advocating for the development of a diverse set of PQC algorithms and hybrid approaches that combine classical and quantum-resistant algorithms. This approach is crucial for maintaining security during the transition to fully quantum-resistant systems. Both BSI and ANSSI recognize that the future cryptographic landscape will require flexibility and the ability to integrate new standards swiftly, ensuring robust security against both current and future threats.

Implementation Challenges and Considerations

Implementing PQC in real-world systems presents several challenges:

- **Performance Overhead:** PQC algorithms, particularly those based on lattices and hash functions, typically have larger key sizes and higher computational demands, which can be challenging for resource-constrained environments like smartcards.

- **Interoperability and Transition:** Transitioning to PQC requires careful planning to ensure interoperability between existing systems and new quantum-resistant algorithms. Hybrid cryptographic schemes are commonly used during this transition phase.
- **Security Assumptions:** The security of PQC algorithms relies on different mathematical foundations than classical cryptography. Continuous research and cryptanalysis are necessary to validate these algorithms against evolving quantum threats.
- **Standardization and Adoption:** The adoption of PQC will depend on how quickly industries and government agencies can integrate new standards, such as FIPS 203, 204, and 205, into their existing cryptographic infrastructures.

Emerging Trends and Future Directions

- **Hybrid Cryptographic Schemes:** The use of hybrid schemes, which combine classical and post-quantum algorithms, will be crucial for ensuring security during the transition to fully quantum-resistant systems.
- **Efficient Hardware Implementations:** Research is focusing on optimizing PQC algorithms for hardware implementation, including in smartcards, to minimize performance overhead.
- **Quantum-Resistant Blockchain:** Securing blockchain systems against quantum threats is an emerging area of research, with PQC algorithms being integrated into blockchain protocols to ensure the long-term security of decentralized systems.
- **Ongoing Cryptanalysis:** Continuous cryptanalysis is essential for identifying potential weaknesses in PQC algorithms. This ongoing effort ensures that the algorithms remain secure as quantum computing technology evolves.

PQC advance In Thales DIS (TDIS)

For several years, Thales DIS has been evaluating cryptographic technologies resistant to quantum computers and is following with interest the NIST standardization initiative initiated in 2017.

TDIS initially focused on LMS-type schemes, based on hash functions. These functions have the advantage of having well-known security properties and were quickly pre-selected by NIST[2]. However, the limitations concerning the number of possible signatures were a blocking point putting a stop to these directions.

Since 2021, TDIS has therefore focused on Lattice-based algorithms, and more precisely Dilithium Crystals[3]. In its first approach, within the H2020 ELECTRON project[4], TDIS has proven the implementation of such an algorithm on a tiny chip hardware such as a smartcard. However, the conclusions from this ELECTRON project showed that it was not, a priori, possible to have a secure implementation at security levels 3 and 5 as defined by NIST of this algorithm, mainly for reasons of RAM constraints. Within the present NANCY project, the defined research scope was to deepen this study.

Conclusion

The field of PQC is advancing rapidly, with significant progress made through the standardization of key algorithms and the publication of guidelines such as FIPS203, FIPS204, and FIPS205. These

[2] SP 800-208, Recommendation for Stateful Hash-Based Signature Schemes | CSRC (nist.gov)

[3] Dilithium (pq-crystals.org)

[4] ELECTRON – project (electron-project.eu)

standards play a crucial role in defining the future of secure digital communications in the quantum era. The integration of these standards into technologies like smartcards is a significant step towards achieving a quantum-secure future, ensuring that our cryptographic infrastructure can withstand the challenges posed by emerging quantum technologies.

3. QKD Experimentations

3.1. Architectural Design & Implementation

3.1.1. 5G Base Station demonstrator applications

In order to provide a use case scenario in line with the NANCY B-RAN fronthaul architecture to experiment with integrating QKD, a set of two web applications has been developed around the standard interfaces of QKD devices. The communication use case emulated by the applications is stated as follows:

"The marketplace reached an agreement between two operators to share a portion of their infrastructure. The Operator 1 (O1) needs to establish a P2P connection between one of its Base Stations (BS1) with another owned by a different Operator (O2). To set up the connection, BS1 needs to send BS2 a set of technical parameters to make it available to incoming connections."

The step-by-step breakthrough would be:

- Step 1: The operators communicate their resources in the NANCY marketplace.
- Step 2: O1 is interested in leasing resources from O2.
- Step 3: A smart contract is signed between O1 and O2.
- Step 4: The smart contract is translated into connection requirements and details.
- Step 5: The BS of O1 (BS1) requests a key from ALICE (QKD equipment at BS1).
- Step 6: BS1 encrypts the data with the received key.
- Step 7: BS1 sends the key identifier and the encrypted data to the BS of O2 (BS2).
- Step 8: BS2 requests a key from BOB (QKD equipment at BS2).
- Step 9: BS2 decrypts the received data.
- Step 10: Communication between BS1 and BS2 is completed.

Based on the above premise, the base stations have been developed as web applications running in container technology. The main goal of mocking up the base stations as independent applications within NANCY is to explore, without technical constraints, the integration of QKD technology using ETSI standard QKD APIs in a way that enables rapid deployment and platform independence to carry out the experiments.

To interact with QKD REST API, each application must own a pre-deployed client certificate. This client certificate will be used to authenticate and retrieve cryptographic keys either from Alice or Bob side. The client certificates are signed by a self-signed authority located at QKD devices and deployed in the Docker containers before communication starts.

The applications have been developed with the following software stack:

- Backend
  - Programming language: RUST
  - Web Framework: ACTIX
  - Database: PostgreSQL
- Frontend
  - Programming language: WebAssembly
  - Asset bundler: TRUNK
  - Web Framework: YEW-RS
  - Styles: TailwindCSS

In the simulated use case, the communication is unidirectional and is always initiated by BS1. The applications are designed to be manually controlled from a web browser to facilitate demonstration purposes. However, both expose all functionality over REST APIs to facilitate an automated interaction, which is especially required for developing the experiments. The components involved in the described functionality can be seen in Figure 1.

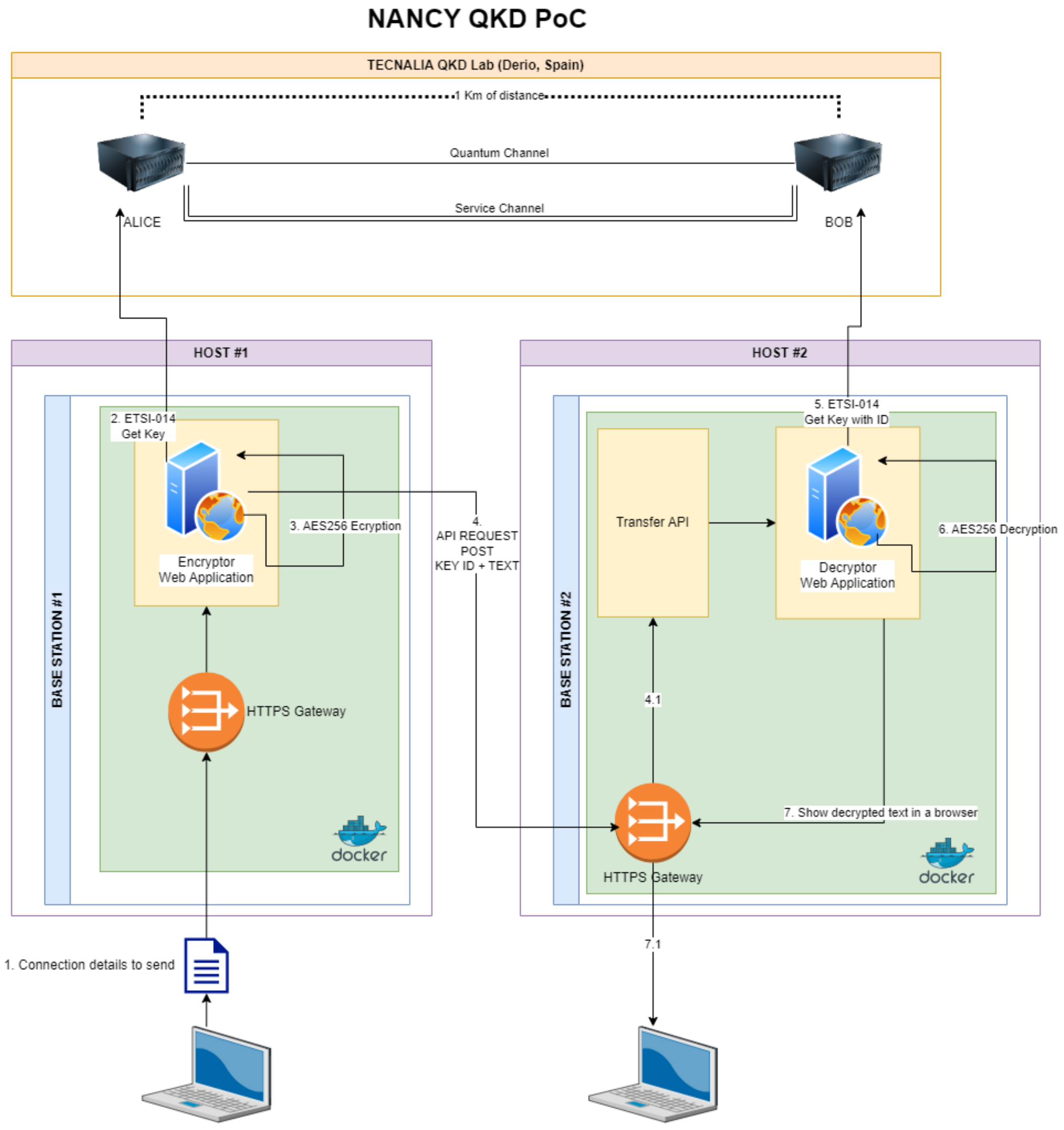


Figure 1: NANCY QKD Applications architecture

Figure 2 shows the activity diagram of an operator sending an encrypted message requesting keys to QKD infrastructure from Base Station 1 to Base Station 2.

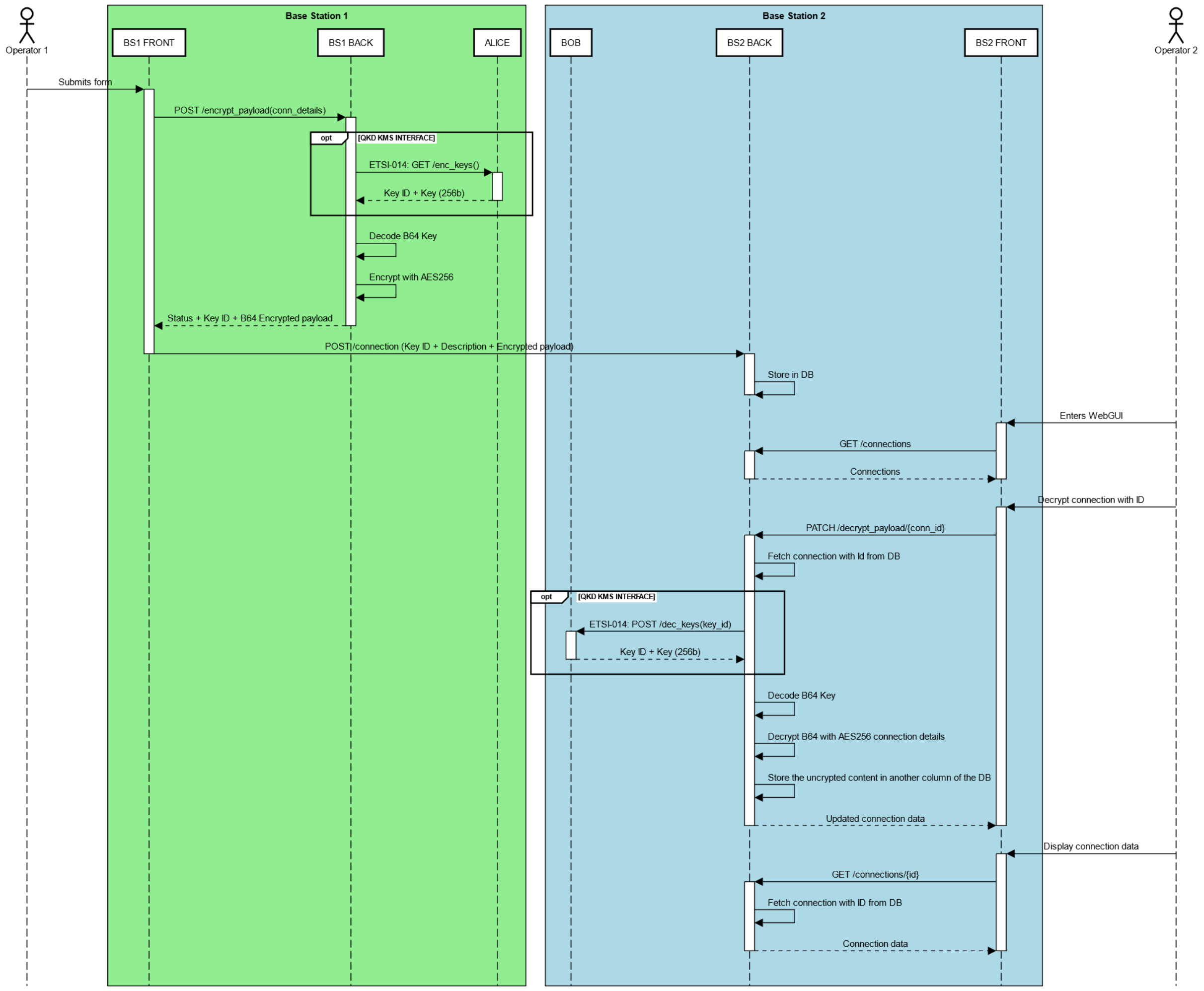

Base Station 1
Base Station 2
Operator 1
BS1 FRONT
BS1 BACK
ALICE
BOB
BS2 BACK
BS2 FRONT
Operator 2
Submits form
POST /encrypt_payload(conn_details)
opt
[QKD KMS INTERFACE]
ETSI-014: GET /enc_keys()
Key ID + Key (256b)
Decode B64 Key
Encrypt with AES256
Status + Key ID + B64 Encrypted payload
POST /connection (Key ID + Description + Encrypted payload)
Store in DB
Enters WebGUI
GET /connections
Connections
Decrypt connection with ID
PATCH /decrypt_payload/{conn_id}
Fetch connection with Id from DB
opt
[QKD KMS INTERFACE]
ETSI-014: POST /dec_keys(key_id)
Key ID + Key (256b)
Decode B64 Key
Decrypt B64 with AES256 connection details
Store the uncrypted content in another column of the DB
Updated connection data
Display connection data
GET /connections/{id}
Fetch connection with ID from DB
Connection data

Figure 2: Activity diagram of BS demo apps

Applications interfaces

The Base Station application REST API specifications are defined in Table 2 and Table 3.

Table 2: BS1 REST API Specification

Endpoint	Method	Body	Response	Description

/api/encrypt_payload	POST	{
		"payload": text
		}
			{
			"status": text,
			"data": {
			"key_id": text,
			"payload": text
			}
			}
				Request to encrypt a text payload.
				BS1 request a new key to ALICE and returns the content encrypted with AES256.

Table 3: BS2 REST API Specification

Endpoint	Method	Body	Response	Description

/decrypt_payload/{id}	PATCH	None	{
			"status": text,
			"data": { "connection": <connection>
			}
			}
				Request to decrypt a connection record stored in the database with the given ID.
				BS2 requests the key associated with the related KeyID from BOB, decrypts the cyphertext with AES256, and stores the clear text within a field

				of the same record in the database.

/api/connections	GET	None	{
			"status": text,
			"results": integer,
			"connections": dict<connection>
			}
				List all connections stored in the database.

/api/connections/{id}	GET	None	{
			"status": text,
			"results": integer,
			"connection": <connection>
			}
				Get a specific connection record from the database matching the given ID.

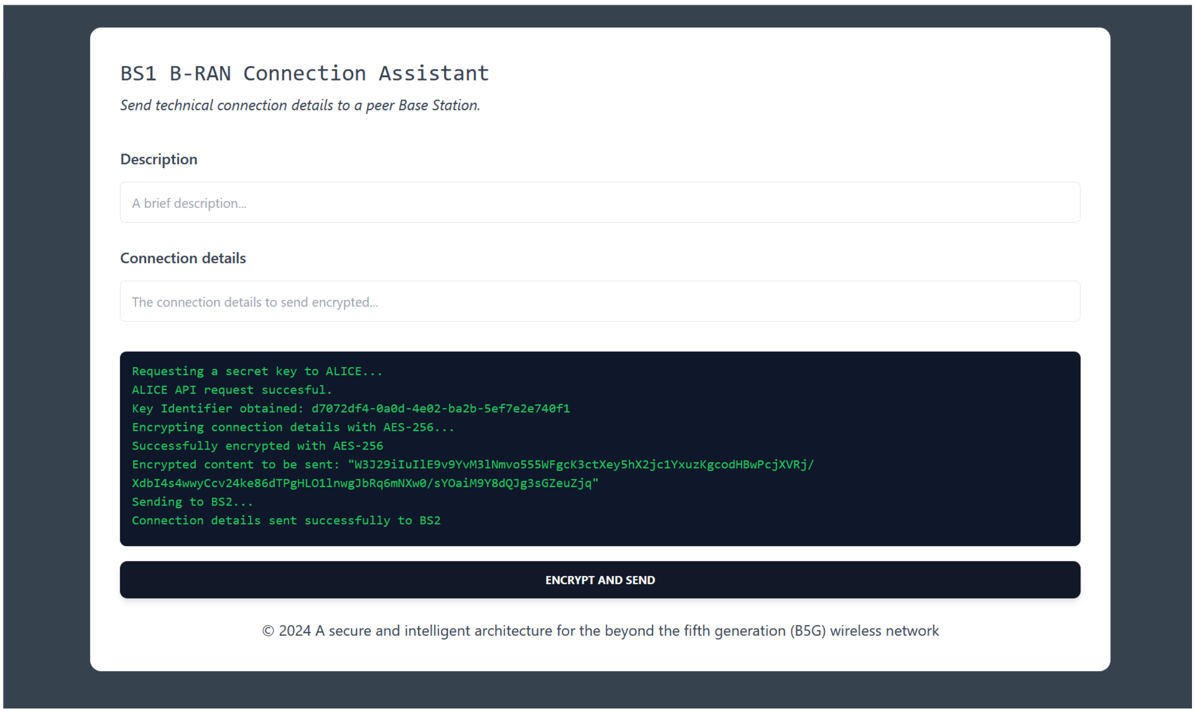
/api/connections/{id}	PATCH	{
		"encryption_key_id": text,
		"description": text,
		"encrypted_content": text,
		"clear_content": text
		}
			{
			"status": text,
			"data": { "connection": <connection>
			}
			}
				Update a connection record of the database matching the given ID.

/api/connections	POST	{
		"encryption_key_id": text,
		"description": text,
		"encrypted_content": text
		}
			{
			"status": text,
			"data": { "connection": <connection>
			}
			}
				Adds a new connection record to the database.

/api/connections/{id}	DELETE	None	HTTP 200 OK (No content)	Deletes a connection record from the database.

Graphical User Interfaces

Both applications can be accessed and operated with a web browser. Base Station 1, or sender GUI, is represented in Figure 3.



BS1 B-RAN Connection Assistant

Send technical connection details to a peer Base Station.

Description

A brief description...

Connection details

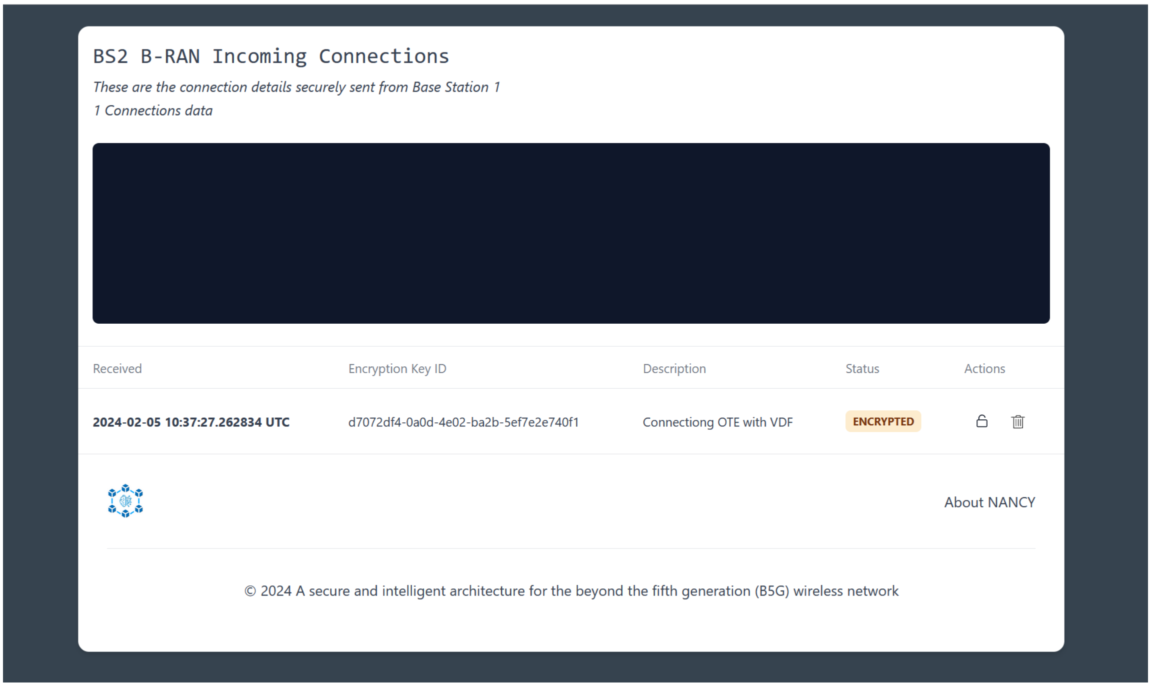
The connection details to send encrypted...

```
Requesting a secret key to ALICE...
ALICE API request succesful.
Key Identifier obtained: d7072df4-0a0d-4e02-ba2b-5ef7e2e740f1
Encrypting connection details with AES-256...
Successfully encrypted with AES-256
Encrypted content to be sent: "W3J29iIuIlE9v9YvM3lNmvo555WFgcK3ctXey5hX2jc1YxuzKgcodHBwPcjXVRj/
XdbI4s4wwyCcv24ke86dTPgHLO1lnwgJbRq6mNXw0/sYOaiM9Y8dQJg3sGZeuZjq"
Sending to BS2...
Connection details sent successfully to BS2
```

© 2024 A secure and intelligent architecture for the beyond the fifth generation (B5G) wireless network

Figure 3: BS1 GUI

The BS2 GUI is represented in Figure 4 showing the list of stored incoming connections with a small indicator of the encryption status.



BS2 B-RAN Incoming Connections

These are the connection details securely sent from Base Station 1

1 Connections data

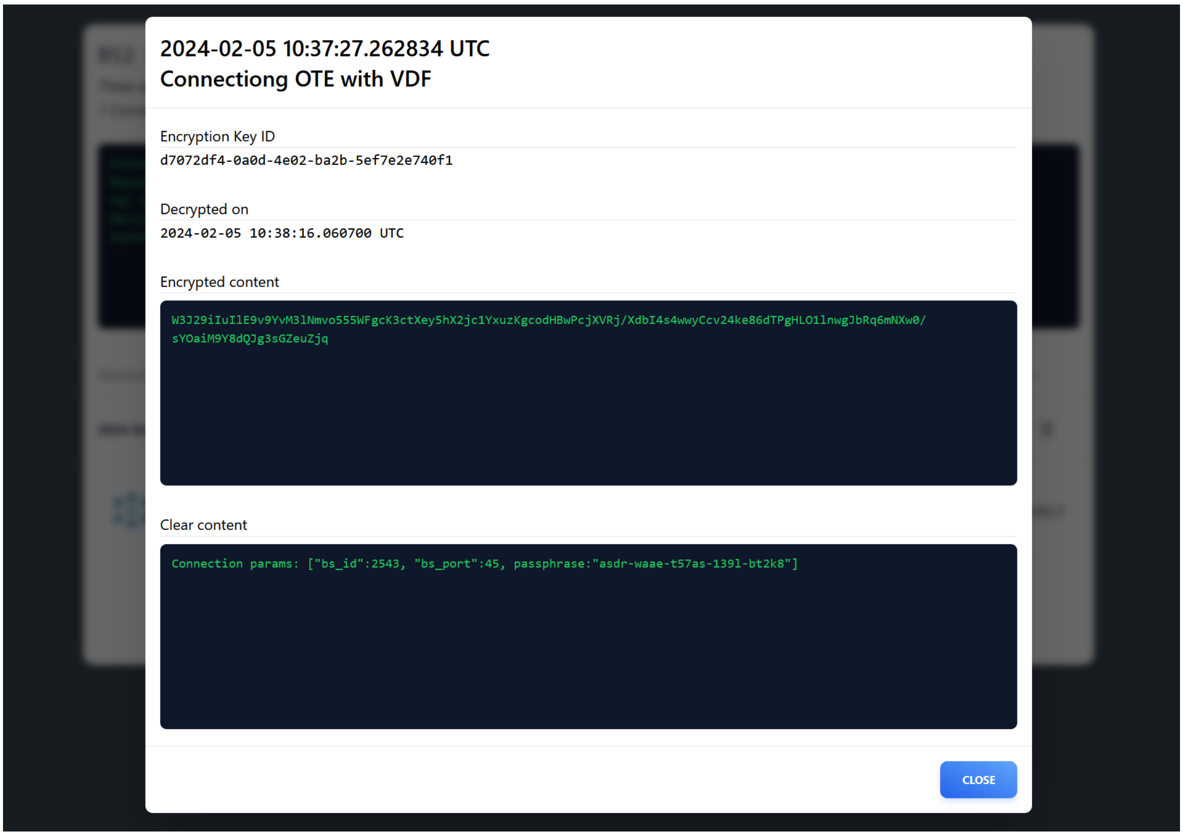
Received	Encryption Key ID	Description	Status	Actions
2024-02-05 10:37:27.262834 UTC	d7072df4-0a0d-4e02-ba2b-5ef7e2e740f1	Connectiong OTE with VDF	ENCRYPTED	

About NANCY

© 2024 A secure and intelligent architecture for the beyond the fifth generation (B5G) wireless network

Figure 4: BS2 GUI with the list of stored connections.

Once the user launched manually the decryption process with the unlock button, a display option appears which opens the dialog of Figure 5 where all details of the connection are displayed.



2024-02-05 10:37:27.262834 UTC
Connectiong OTE with VDF

Encryption Key ID
d7072df4-0a0d-4e02-ba2b-5ef7e2e740f1

Decrypted on
2024-02-05 10:38:16.060700 UTC

Encrypted content

```
W3J29iIuIlE9v9YvM3lNmvo555WFgcK3ctXey5hX2jc1YxuzKgcodHBwPcjXVRj/XdbI4s4wwyCcv24ke86dTPgHLO1lnwgJbRq6mNXw0/
sYOaiM9Y8dQJg3sGZeuZjq
```

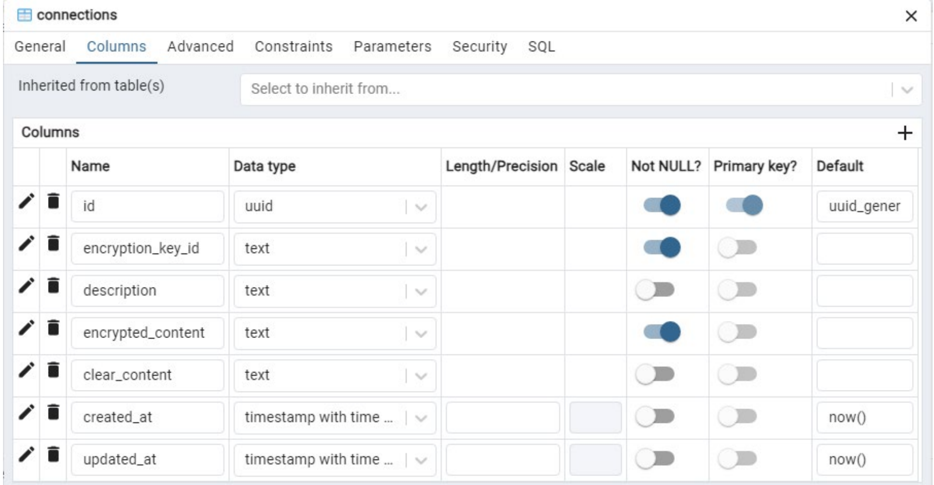
Clear content

```
Connection params: ["bs_id":2543, "bs_port":45, passphrase:"asdr-waae-t57as-139l-bt2k8"]
```

Figure 5: BS2 GUI showing the decrypted content.

Database schema

Base Station 2 application holds a PostgreSQL database where the incoming connection data records are stored, waiting to be requested to decrypt by an administrator. The database also tracks a record's lifecycle timestamps, such as when the connection was received, the decryption date, the ciphertext, cleartext, or the associated key ID. The database schema is depicted in Figure 6



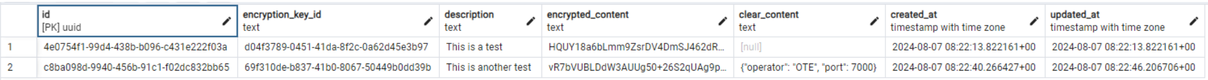
Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	uuid			●	●	uuid_gener
encryption_key_id	text			●	○	
description	text			○	○	
encrypted_content	text			●	○	
clear_content	text			○	○	
created_at	timestamp with time ...			○	○	now()
updated_at	timestamp with time ...			○	○	now()

Figure 6: BS2 Database Schema

Figure 7 shows two records stored in the Connections table. The first entry has an empty "clear_content" column. This is because the connection data has been received from the API but is still pending decryption. In the second row, a connection record with the "clear_content" column filled contains the actual content after decrypting.



	id [PK] uuid	encryption_key_id text	description text	encrypted_content text	clear_content text	created_at timestamp with time zone	updated_at timestamp with time zone
1	4e0754f1-99d4-438b-b096-c431e222f03a	d04f3789-0451-41da-8f2c-0a62d45e3b97	This is a test	HQUY18a6bLmm9ZsrDV4DmSJ462dR...	[null]	2024-08-07 08:22:13.822161+00	2024-08-07 08:22:13.822161+00
2	c8ba098d-9940-456b-91c1-f02dc832bb65	69f310de-b837-41b0-8067-50449b0dd39b	This is another test	vR7bVUBLDdW3AUUg50+26S2qUAg9p...	{"operator": "OTE", "port": 7000}	2024-08-07 08:22:40.266427+00	2024-08-07 08:22:46.206706+00

Figure 7: BS2 Database content example

Each record includes a column that denotes the KeyID linked to the QKD key used for encryption operations. This allows the encrypted content to be stored independently when decryption is requested.

3.1.2. QKD Simulations

One of the most innovative technologies in the field of quantum communications is QKD. This technique guarantees that encryption keys are safely passed between remote receivers, therefore offering unmatched security. The main advantage of QKD is its capacity to identify any attempt of eavesdropping during communication, therefore preserving security and privacy. Unlike conventional encryption systems, which depend on complex mathematical procedures, QKD security is built on the fundamental laws of quantum physics, including Heisenberg's uncertainty principle and the quantum no-cloning theorem [18]. These ideas ensure the integrity of the key exchange mechanism by stopping eavesdroppers from intercepting or copying quantum states without producing obvious disturbances.

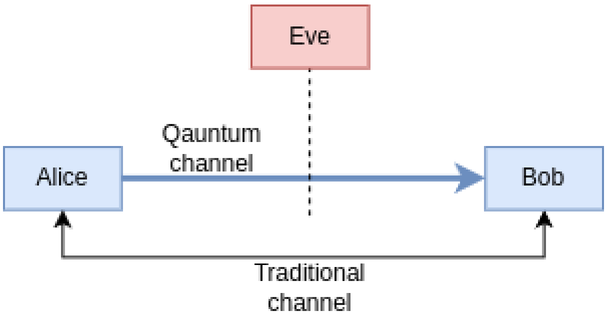


Figure 8: BS2 Quantum communication model

The basic framework of the QKD communication process incorporates both quantum and traditional channels that operate in collaboration. The quantum channel is responsible for the sensitive work of quantum key distribution, as illustrated in Figure 8, while the traditional channel supports the essential post-processing section, including key agreement and eavesdropper detection between the sender and receiver. In order to ensure that any unauthorized access is quickly identified and eradicated, this dual channel structure is an essential component of QKD systems. The utilization of the unique physical properties of quantum information carriers enables QKD to be attained while simultaneously protecting against surveillance. Each attempt by an unauthorized third party to obtain knowledge of the shared key in QKD results in a substantial increase in the QBER of the transmitted data. The secure communication rate, when combined with the QBER, is a critical metric for assessing the security and efficacy of a variety of QKD systems [19].

The NANCY QKD Simulator

NANCY provides a unique simulation environment that is essential for in-depth investigation of QKD systems. This simulator provides comprehensive and flexible methods to gain insights into discrete and DPR-QKD approaches. To create a uniform environment, our simulator combines protocols from the discrete sector including BB84 and B92 as well as from the DPR sector with the COW protocol. This presents a benefit for researchers and users seeking a comprehensive grasp of the advantages and drawbacks of protocols without being constrained by the limitations often encountered in real-world testing scenarios. This simulator has distinctive characteristics and flexibility as it can be customized to fit individual requirements. Unlike QKD deployments with limiting factors like cost, time, and resources that could compromise equipment performance, our simulation overcomes these obstacles. For instance, researchers have the freedom to explore fibre lengths – whether within typical ranges or stretching to extreme distances – without worrying about logistical limitations. This adaptability allows them to look at scenarios that could prove difficult or even impossible to recreate in a lab environment. Such capacity to surpass pragmatic constraints promotes significant comprehension and analysis that results in a better knowledge of how these mechanisms function in many contexts.

Versatility is not the sole attribute of our simulator. Furthermore, the simulator plays a vital part in evaluating the security components of QKD systems and lets users investigate the features of every protocol under several eavesdropping situations. This makes the simulator a versatile tool for both research and deployment purposes.

Moreover, the simulator provides real-time visual depictions of important performance metrics including key generating rate and QBER. These graphic tools allow anyone to understand the dependability and efficiency of the applied techniques under several contexts. By looking at how QBER changes statistically with regard to parameters such as fibre length, signal power, frequency, and eavesdropping intensity, users can evaluate the strength of the protocols. Likewise, comparing the key generation rate under many scenarios helps one to grasp the practicality and effectiveness of the technique. Real-world applications depend on properly using these visual aids to make decisions and maximize the efficiency of the simulation.

This simulation tool is not only a research tool but also a major resource for encouraging deployment in the field of quantum cryptography through the use of direct comparison of several protocols in identical environments to highlight their advantages and disadvantages. It provides an opportunity to improve understanding of quantum communication as well as the opportunity to mimic realistic circumstances. All in all, it offers a controlled but realistic environment that overcomes conventional physical limits, therefore representing a step forward in QKD exploration.

Protocol Simulation and Process Analysis

Under the framework of a single-photon quantum key distribution system, two distinct channels—quantum and traditional—formulate the basis of communication. Every channel has a different function in ensuring the safe key exchange and the consequent eavesdropping detection.

1. Transmission of qubits, which are applied in key distribution systems, is dependent on the quantum channel. Usually encoded in single photons, these qubits show important information in different quantum states. Quantum states are delicate; hence any eavesdropping effort always alters the qubits, which real communication parties would find.
2. After the qubits have been sent via the quantum channel, the classical channel is used for public conversation among the communication parties. This channel is used to reconcile keys, repair errors, and detect eavesdropping. Importantly, while the classical channel is susceptible to eavesdropping, it does not jeopardize the security of the key because the information transmitted here is insufficient for an eavesdropper to reassemble the final encryption key without discovery.

Eavesdropping detection in the QKD process

An eavesdropping event is most likely to happen at the key distribution phase when an opponent could try to intercept qubits travelling the quantum channel. However, according to quantum physics rules, this kind of interference clearly defects the transmitted key. After the transmission, both parties compare a portion of their key via the traditional route to ascertain the error rate once the key distribution is finished. Should the error rate exceed a preset level, an eavesdropper is detected, the key is removed.

The Role of QBER in Quantum Key Distribution

An essential metric of the QKD system's security and performance is the QBER. It shows the errors of the raw key that has been transmitted between the authorized members, Alice and Bob. In the scenario, when an eavesdropper, also known as Eve, attempts to intercept the quantum states containing the key, the disturbance entering the system shows up as a rise in the QBER. Basically, the QBER is a direct assessment of the eavesdropping sensitivity of the system.

A high QBER indicates that the eavesdropper could be learning more about the key at the expense of the authorized receiver. Therefore, a greater QBER influences the general security and efficiency of the key distribution process, thereby reducing the secure key rate in the stage of following error correction of the protocol. Under these conditions, the secure key rate is the count of error-free bits following privacy amplification and error correction.

Balancing QBER with Secure Communication Rate

Maintaining appropriate levels of QBER will ensure robust security. Studies show that Alice and Bob may still generate a safe key with conventional post-processing methods including error correction and privacy amplification as long as the QBER is less than a certain threshold. Privacy amplification in particular is crucial in decreasing any partial information that Eve may have gathered, assuring the final key's integrity. However, as the QBER approaches this level, the amount of data that must be destroyed during error correction increases, lowering the secure transmission rate.

Achieving an equilibrium between security and key generating efficiency depends on keeping a low QBER. Should the QBER exceed the threshold, too much information becomes available to an eavesdropper, making a secure key generation impossible. This dynamic demonstrates the need to reduce QBER to provide strong security and effective key distribution.

Considering the above traits, we can conclude that QBER is more than just an indicator of the communication channel's error rate; it also provides a significant indication of the information that a possible eavesdropper would have acquired. Therefore, while raising general communication efficiency, QKD systems can attain safe key distribution by tracking and regulating the QBER. Parties can effectively distil a secure key as long as the QBER stays below the crucial threshold, therefore preserving communication secrecy against efforts at eavesdropping.

Simulated protocols

With the rapid advancement of technology and the increasing frequency of cyber-attacks, ensuring secure communication is a basic requirement. A promising area in addressing the aforementioned problem is the use of quantum physics. A typical method is the use of QKD techniques to securely transfer an encryption key from the transmitter to the receiver. In the context of this simulation, 3 different QKD techniques were used, namely BB84, B92, and COW. The first 2 are discrete with BB84 being the first and also the most widely used technique created, while B92 is characterized by a simpler structure and higher performance. At the same time, in the context of a better but deeper study of QKD, the COW protocol belonging to Distributed-Phase-Reference was also used.

COW

Utilizing decoy states to boost security, H. Zbinden et al. developed the COW protocol in 2004 [20, 21]. Since the COW protocol is mostly based on passive optical components, it stands out for simplicity of implementation. Moreover, it is polarization insensitive, which makes it perfect for fibre-based communications free of polarization control devices [20]. This architectural benefit allows the COW protocol to link easily into a contemporary fibre-optic system.

The COW protocol's resistance against photon number splitting (PNS) attacks—a common weakness in many QKD systems—is among its most important benefits. Though its resistance to other kinds of attacks is still under investigation [21], new experimental implementations have produced positive findings. For instance, although other studies have reported rates of up to 15 bps over 250km, safe key distribution rates of 2.5 bps have been recorded throughout lengths of 150km [22]. The parts that follow provide a full description of how the COW protocol works.

Overview of the COW Protocol

The COW technique is intended to generate high key rates by utilizing the timing of photon arrivals at the receiver. Figure 9 shows a schematic diagram of how the protocol operates. The procedure can be summarized as follows [20]:

1. Alice compresses binary bits into time slots before transmitting them to Bob. In a basic configuration without decoy states, the likelihood of communicating a "1" is the same as the probability of broadcasting a "0" (both 50%). However, if decoy states are introduced, the odds are changed. If the probability of creating a decoy bit is f, the remaining probability $(1-f)/2$ is equally divided between "1" and "0," with each having a probability of $(1-f)/2$.
2. Bob uses two detectors to measure photon arrival times: the data detector (DD) for raw key generation and the monitoring detector (MD) for security analysis. The key bits are generated

based on the time periods in which DB detects a photon, whereas MD detections are utilized to ensure the transmission's integrity and detect potential eavesdropping.

3. Bob publicly publishes the bit positions corresponding to DD clicks and the detection times recorded by MD. Alice uses this information to determine whether the decoy sequences and bit sequences ("1" and "0") are visible at the interferometer's output. If Eve is present, her interference will disrupt the coherence between successive pulses, causing a noticeable abnormality in vision.
4. Alice informs Bob which bits were the decoys, so he can eliminate them from his key. This guarantees that only the valid bits are going to be stored.
5. For the final key, error corrections and privacy amplification may be applied to increase the possibilities of a message without Eve's presence.

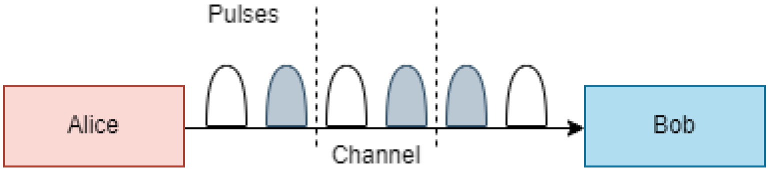


Figure 9: Schematic diagram of the COW protocol

Advantages and Practical Implementation

Emphasizing simplicity and efficiency, the architecture of the COW protocol offers several advantages. It is an ideal choice for long-distance fibre-based QKD systems that do not require complex polarization control techniques due to its polarization insensitivity and dependence on passive optical components. Moreover, the protocol's resistance to PNS assaults provides even another degree of security.

The COW protocol's high key generation rates, ease of implementation, and robust security features make it a strong candidate for real-world quantum communication systems. Ongoing research and experiments continue to investigate the protocol's performance under various settings, as well as its responsiveness to various attack techniques, with the objective of improving its effectiveness and scalability.

BB84

The BB84 protocol is among the first and most well-known uses of quantum physics for encryption. Over the years, the BB84 protocol has attracted much study and development [23]. It modulates a sequence of random bits onto the polarization states of individual photons that function as qubits.

How does the BB84 protocol work?

Using two sets of corresponding bases, the computational (rectilinear) basis (CC) and the diagonal basis (DD), the transmitter, Alice, and the receiver, Bob, apply the BB84 protocol. Subsequently, individual photons are polarized based on these bases, which also represent binary values "0" and "1". From the fact, that two non-orthogonal polarization states make up each basis, hence it is difficult to measure the quantum states without clearly causing errors—a necessary quality for detecting possible eavesdropping.

Photon Polarization and Basic Encoding

The BB84 protocol uses four polarization states:

- Horizontal (0°),
- Vertical (90°),
- Diagonal (45°), and

- Antidiagonal (135°),

which are separated into two orthogonal bases:

- The computational basis (Z-basis) is made up of two orthogonal polarization states: horizontal (0°) and vertical (90°). In the simulation, this base is represented by the sign +.
- Diagonal Basis (X-basis): The diagonal (45°) and anti-diagonal (135°) polarization states are orthogonal to one another. This basis is represented by the symbol x.

Table 4 summarizes the encoding of converting quantum state information into classical bits.

Table 4: Polarization Coding Scheme

Base	Polarization angle	Bit value
+	0°	0
	90°	1
X	45°	0
	135°	1

Execution of the BB84 Protocol

To carry out the BB84 protocol, Alice sends Bob a succession of single photons, each randomly polarized using either the computational basis (Z-basis) or the diagonal basis (X-basis). Each photon's polarization is achieved using an electro-optical modulator, which modifies the qubit's polarization state in accordance with Alice's random bit sequence.

When Bob receives the photons, he measures their polarization state using a randomly chosen basis with equal chance. Bob's choice of basis is independent of Alice's, thus there's a 50% probability he'll select the correct basis for each photon, allowing him to correctly infer the value of the associated bit. If Bob chooses the erroneous basis, his measurement produces a random result that causes some inconsistencies in the shared key.

Key Sifting and Eavesdropping Detection

Following the transmission, Alice and Bob publicly compare only the bases and not the outcomes for each photon via a classical channel. They keep only the sections in which their chosen bases match. This procedure, known as key sifting, usually discards roughly half of the originally sent bits. The resulting filtered key can then be subjected to mistake correction and privacy amplification to generate a safe key.

The BB84 protocol's capacity to identify eavesdropping is one of its most important aspects. When an eavesdropper (Eve), trying to catch and measure photons, will inevitably disturb the quantum states, the QBER will increase. Based on this, Alice and Bob can assume that their communication has been compromised and remove the keys when the QBER value surpasses a threshold.

Conclusion

The BB84 protocol offers a safe method for realizing a quantum key distribution system. It is a crucial protocol in the building of safe quantum communication networks since it offers a high barrier against eavesdropping by leveraging non-orthogonal bases and photon quantum properties.

B92

The B92 protocol is a simpler and more cost-effective adaptation of BB84. While the B92 protocol has some similarities with the BB84 protocol, it was expressly designed to reduce complexity while retaining secure key distribution [24]. Unlike BB84, which relies on two non-orthogonal bases, the B92 protocol employs only one quantum alphabet, or, more precisely, a single set of basis states. This speeds up the process by lowering the number of states required to encode information.

How Does the B92 Work

The B92 protocol only uses two of the four non-orthogonal states from the BB84 protocol to represent binary values. In this approach, Alice encodes her classical bits (0s and 1s) using only two polarization states, commonly represented as:

$$\begin{cases} "1" = |\theta_+\rangle \\ "0" = |\theta_-\rangle \end{cases}$$

These states represent photon polarizations at particular angles relative to the vertical axis, where $0 < \theta < \pi/4$. In practice, classical bit "0" is frequently encoded by a photon with horizontal polarization, whereas bit "1" is encoded by a photon polarized at a 45° angle.

The Encoding and Transmission Process

Alice prepares her qubits using the B92 protocol, randomly selecting one of the two non-orthogonal states for each bit she wants to communicate. For example, she may use horizontal polarization to indicate a "1" and diagonal polarization (45°) to represent a "0." Bob receives the photon qubits after they have been encoded via a quantum channel, as it's represented as

$$\begin{cases} "1" = 0° \\ "0" = 45° \end{cases}$$

Decoding and Key Sifting

Bob uses one of two non-orthogonal bases to measure the polarization states of incoming photons. However, unlike BB84, Bob just utilizes one basis for each measurement, allowing him to identify whether the received bit was a "1" or a "0." The protocol's simplicity stems from the use of fewer polarization states, which makes it easier and less expensive to implement.

Following these measures, Bob notifies Alice of the discovered events via a conventional channel. Importantly, Bob does not disclose the methodology for the measurement or the outcome of the measurement. Using this feedback, Alice and Bob can reach an agreement on a filtered key. Only the instances in which Bob successfully identified a photon are preserved; the rest are discarded. This step assures that the final key is constructed entirely from unambiguously received bits.

Security and Efficiency

The B92 protocol is simpler than the BB84 protocol, but it still ensures resilience against quantum attacks. Because the encoding states are non-orthogonal, any eavesdropper attempting to intercept and measure photons will inevitably introduce noticeable mistakes, as seen in BB84. The protocol's ability to detect eavesdropping stems from the fact that measurements of non-orthogonal states disrupt the system, causing changes that Alice and Bob can witness.

Conclusion

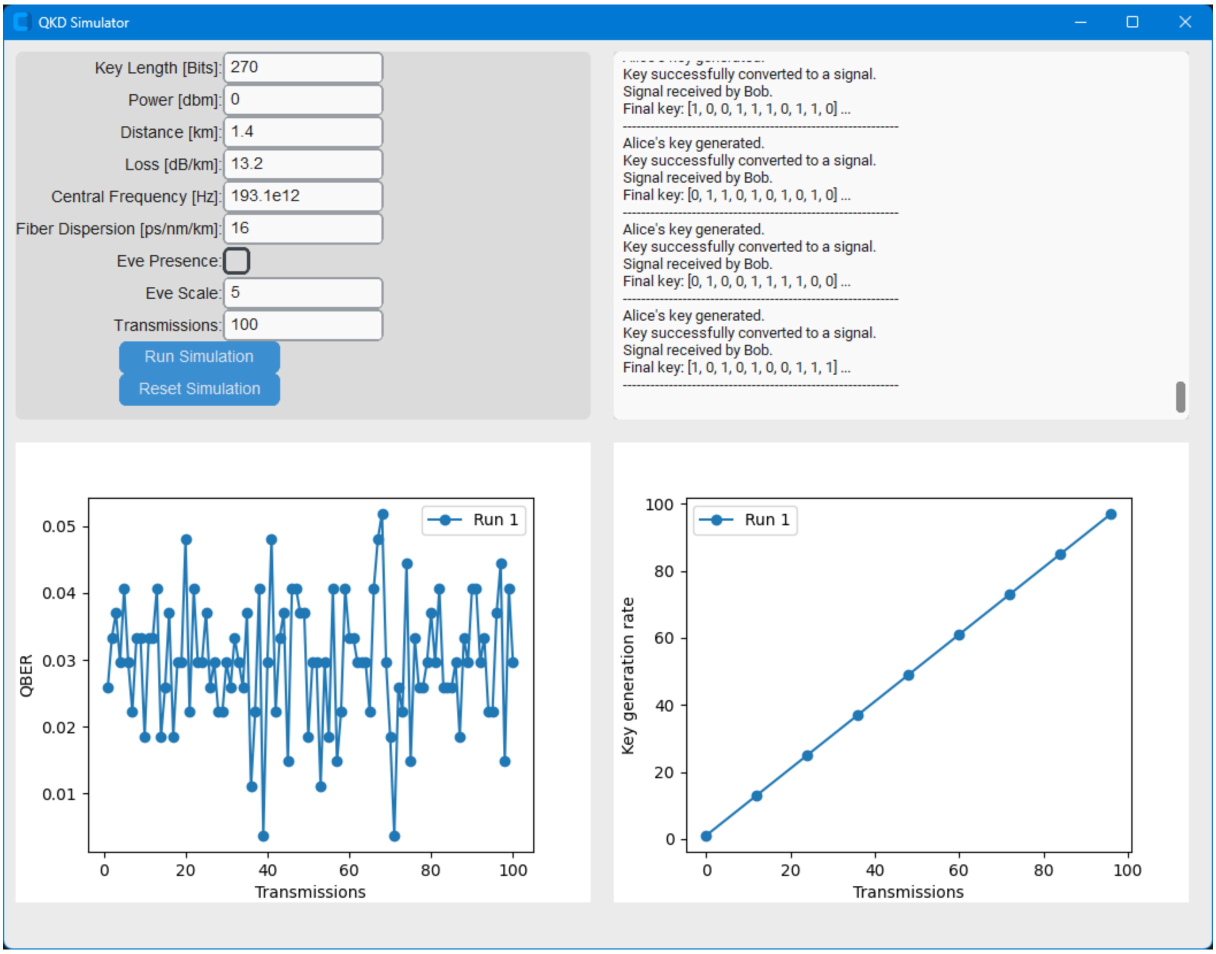
Finally, the B92 protocol reduces the number of polarization states needed for encoding, hence simplifying the quantum key distribution process. Based on the fact that B92 relies on a single quantum alphabet instead of two, it lowers the general complexity and size of the system while still ensuring a safe key exchange. This protocol keeps the fundamental ideas of quantum cryptography despite its simplicity, which lets Alice and Bob create a shared key with strong security against eavesdropping.

Numerical results

In this section, we provide numerical results of three distinct QKD protocols, namely BB84, B92, and COW. Among the most studied protocols in QKD, these provide particular advantages and trade-offs in terms of security, key rate, and resilience against eavesdropping attacks. The purpose of the numerical simulations is to evaluate the performance of each protocol under a variety of conditions. We present significant performance measurements like the QBER and safe key rate, thereby providing a comparative examination of every approach.

DPR-QKD

In this simulation scenario, we explore DPR-QKD, a method that uses continuous variables like quantum states of light to achieve secure key distribution between Alice and Bob. Figure 10 illustrates the graphical UI of the QKD Simulator for the COW protocol.



Key Length [Bits]: 270
Power [dbm]: 0
Distance [km]: 1.4
Loss [dB/km]: 13.2
Central Frequency [Hz]: 193.1e12
Fiber Dispersion [ps/nm/km]: 16
Eve Presence: □
Eve Scale: 5
Transmissions: 100

```
Key successfully converted to a signal.
Signal received by Bob.
Final key: [1, 0, 0, 1, 1, 1, 0, 1, 1, 0] ...
----------------------------------------------------------------
Alice's key generated.
Key successfully converted to a signal.
Signal received by Bob.
Final key: [0, 1, 1, 0, 1, 0, 1, 0, 1, 0] ...
----------------------------------------------------------------
Alice's key generated.
Key successfully converted to a signal.
Signal received by Bob.
Final key: [0, 1, 0, 0, 1, 1, 1, 1, 0, 0] ...
----------------------------------------------------------------
Alice's key generated.
Key successfully converted to a signal.
Signal received by Bob.
Final key: [1, 0, 1, 0, 1, 0, 0, 1, 1, 1] ...
----------------------------------------------------------------
```

Figure 10: Screenshot of the DPR-QKD interface

Parameters

Initially, you are presented with a variety of degrees of freedom that can be modified to customize the simulation to specific requirements:

1. Key Length: The length of the random bit string Alice creates and transmits.
2. Power: The power emitted in the optical fibre of the quantum signal.
3. Distance: The length of the optical fibre between Alice and Bob.
4. Loss: The quantum signal loss over the optical fibre during the transmission.
5. Central Frequency: The central frequency selected for the transmission.
6. Fibre Dispersion: The fibre's optical dispersion.
7. Eve Presence: The existence of an eavesdropper in the quantum channel.
8. Eve Scale: It controls how aggressively the eavesdropper tries to gain information from the quantum signal.
9. Transmissions: This parameter determines the number of transmissions during the simulation.

Metrics

After configuring these parameters, running the simulation will produce various plots of:

- Key Generation Rate**:** This plot shows the rate at which secure keys are generated between Alice and Bob, reflecting the overall effectiveness of the DPR-QKD protocol.
- QBER**:** This plot displays the error rate of the quantum bits transmitted, providing insights into the fidelity of the key distribution and the impact of factors like fibre loss and eavesdropping.

Plots

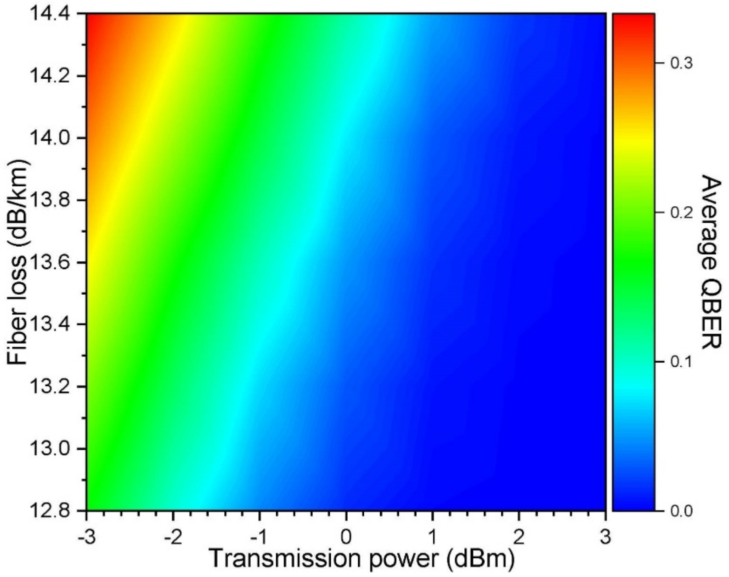


Figure 11: BS2 Average QBER as a function of the fibre loss and transmission power

Figure 11 illustrates the link quality between fibre loss and transmission power in terms of average QBER. The data highlight that increasing transmission power results in poorer QBER values regardless of whether the fibre loss is modest or high. In contrast, reduced transmission power is related to higher average QBER values, with the highest QBER recorded when fibre loss is at its maximum. Furthermore, the plot displays a distinct zone for transmission power higher than 1 dBm in which the average QBER is continuously low, regardless of fibre loss value. In this range. This suggests a threshold, after which power above a certain level reduces the influence of fibre loss on QBER. According to this figure, increasing transmission power is a successful method for reducing QBER in a communication system, even in cases where fibre loss is high. The region where power surpasses 1 dBm indicates a minimum power threshold that must be met in order to keep QBER low for a fibre that is characterized by 14.4 dB/km loss. If we substitute this fibre with a less lossy one, it is possible to attain the same QBER with lower transmission power. Therefore, for best performance and minimal error rates, maintaining transmission power over a certain level depending on the fibre losses is critical.

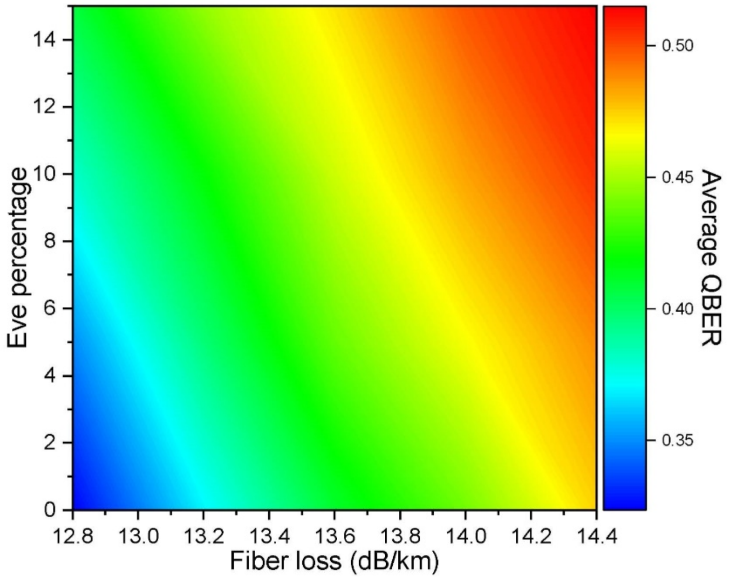


Figure 12: Average QBER as a function of the fibre loss and Eve's percentage

Figure 12 depicts the average QBER for different eavesdropping and fibre loss configurations. As expected, there is a clear relationship between the eavesdropping scale and the average QBER. Specifically, higher eavesdropping values result in higher QBER. When both the eavesdropping percentage and fibre loss are at their peak, QBER attains the highest values as well. For the best fibre loss configuration, the average QBER stays low, even if the scale of eavesdropping is significant. In contrast, at large levels of fibre loss, the QBER rises significantly independent of the eavesdropping scale, demonstrating that fibre loss can have a major impact on the system. This suggests that maintaining low fibre loss is critical for keeping low error rates, even in the face of eavesdropping. Reducing fibre loss should be a top goal in communication system security since it lowers the possibility of QBER escalation, even in the presence of eavesdropping. All in all, we notice that both eavesdropping scale and fibre loss alone lead to a rise in QBER, with their combined effects being extremely damaging to the system's security.

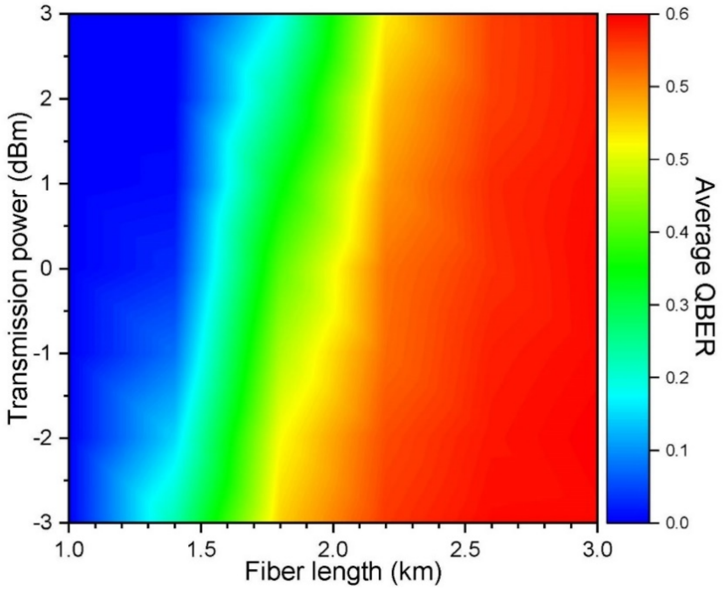


Figure 13: Average QBER as a function of the fibre length and transmission power

Figure 13 presents the average QBER for different values of transmission power and fibre. As anticipated, the plot indicates that the longest fibre lengths correlate with the highest average QBER values. Notably, over a particular fibre length threshold, the average QBER achieves its peak value regardless of the transmission power level. In contrast, when using the shortest fibre lengths, the curve exhibits the lowest QBER values, regardless of power levels. This shows that reducing fibre length is critical for maintaining low QBER. Fibre length plays an important role in determining the average QBER in a communication system. While increasing transmission power can generally aid in reducing QBER, this effect fades once a particular fibre length threshold is reached, after which QBER remains high independent of power modifications. To maintain the best performance and decrease error rates, fibre lengths should be kept to the shortest possible value. Finally, let us stress that measures aimed at reducing or optimizing fibre length may be more beneficial than just boosting transmission power in terms of preserving low QBER levels over extended transmission distances.

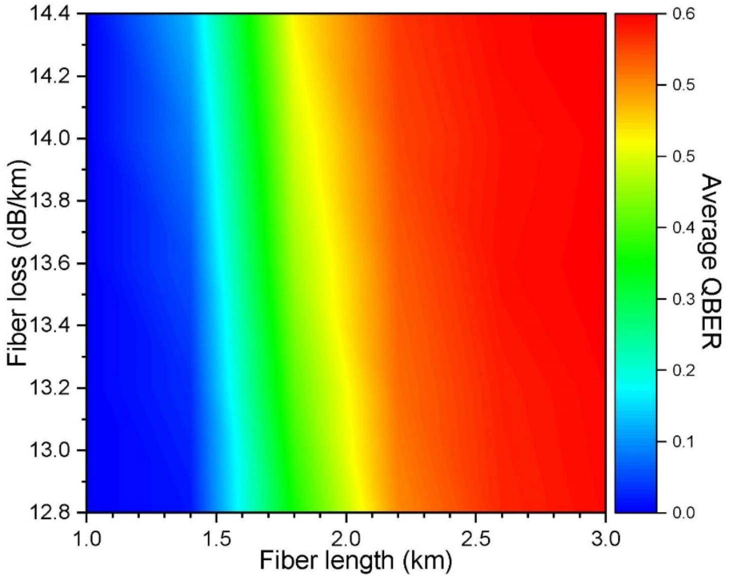


Figure 14: Average QBER as a function of the fibre length and fibre loss

Figure 14 depicts the link quality between fibre loss and fibre length measured through the average QBER. As expected, for longer fibre deployments, the average QBER increases, while for shorter fibres it remains low independent of fibre loss. It is also worth noting that a minor initial increase in fibre length causes a noticeable spike in QBER, which then varies directly with subsequent changes in fibre length across the whole spectrum of fibre loss. All in all, independent of fibre loss degree, low QBER depends on maintaining shorter fibre lengths. Moreover, even a small increase in fibre length can cause a notable change in QBER, thereby underlining the need for precisely regulating and minimizing fibre length to keep low error rates in communication systems.

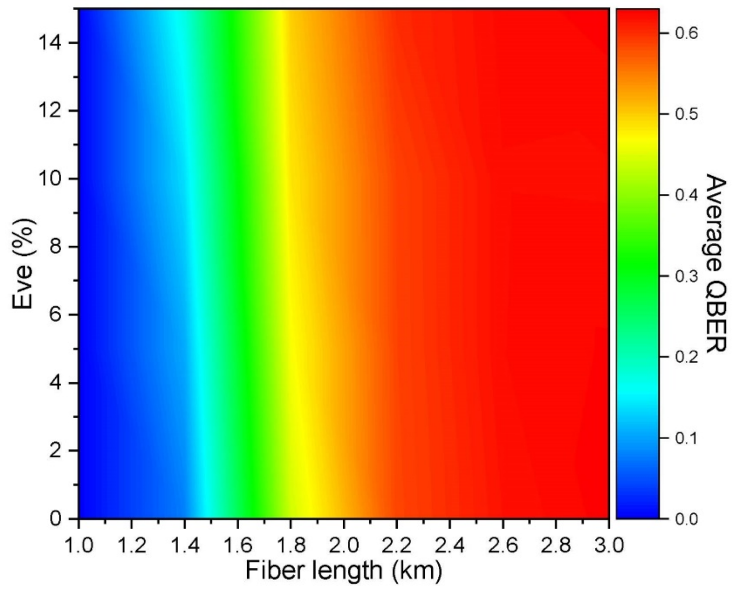
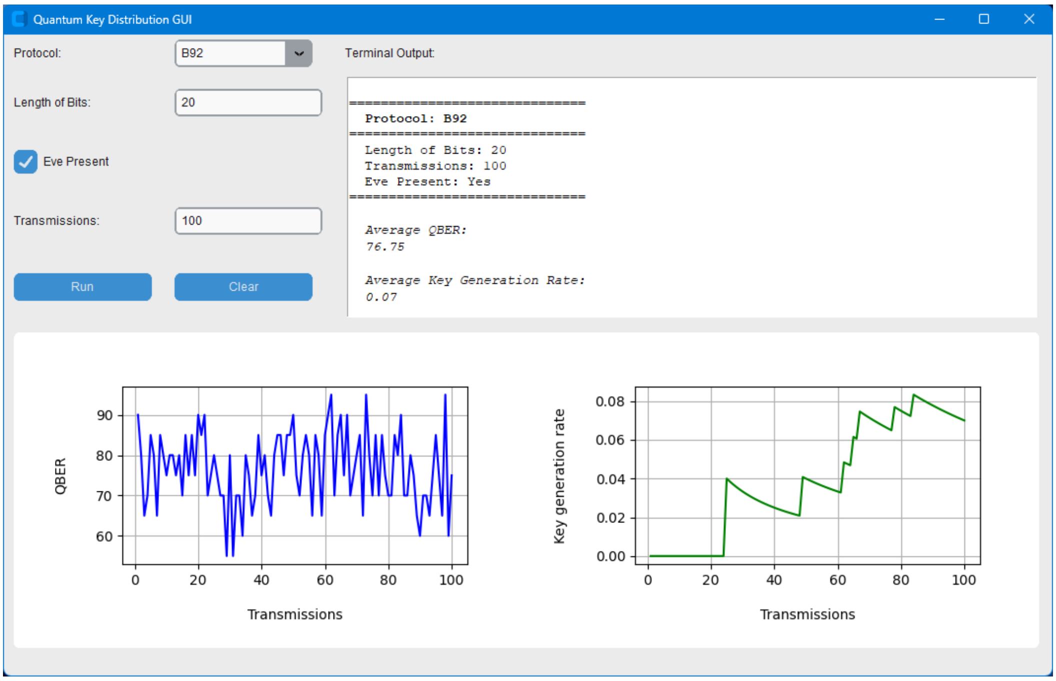


Figure 15: Average QBER as a function of the fibre length and Eve's percentage

In Figure 15 we can observe the average QBER as a function of the eavesdropping scale and fibre length. The findings show that the shortest fibre lengths have the lowest average QBER values, independent of eavesdropping level. Once the fibre length exceeds a threshold of around one-third of its total values, the QBER achieves its maximal value, indicating a critical limit beyond which further increases in fibre length have no effect on error rates. On the other hand, the average QBER remains consistent across the whole range of eavesdropping scales. This illustrates that fibre length is a dominant factor in determining average QBER, outweighing the impacts of the eavesdropping scale.

DV-QKD

This section explores the discrete QKD to simulate a secure key exchange between Bob and Alice. Figure 16 illustrates the graphical UI of the QKD Simulator for the BB84 and B92 protocols.



Protocol: B92

Length of Bits: 20

☑ Eve Present

Transmissions: 100

```
================================
  Protocol: B92
================================
  Length of Bits: 20
  Transmissions: 100
  Eve Present: Yes
================================

  Average QBER:
  76.75

  Average Key Generation Rate:
  0.07
```

Figure 16: Screenshot of the DV-QKD interface

The top toggle selects the QKD protocol for the simulation:

- BB84: Renowned for security and efficiency, the most used QKD protocol.
- B92: Reduced quantum states provide a simplified form of BB84.

Parameters

The degrees of freedom for the selected protocol include:

- Length of Bits: The size of Alice's raw key that is transmitted through the quantum channel.
- Transmissions: The number of total transmissions.
- Eve presence: The presence of an eavesdropper trying to intercept the communication.

Metrics

Running the simulation after defining these settings would not only demonstrate the important generating rate but also offer comprehensive graphs for the QBER and key generation rate.

Numerical results

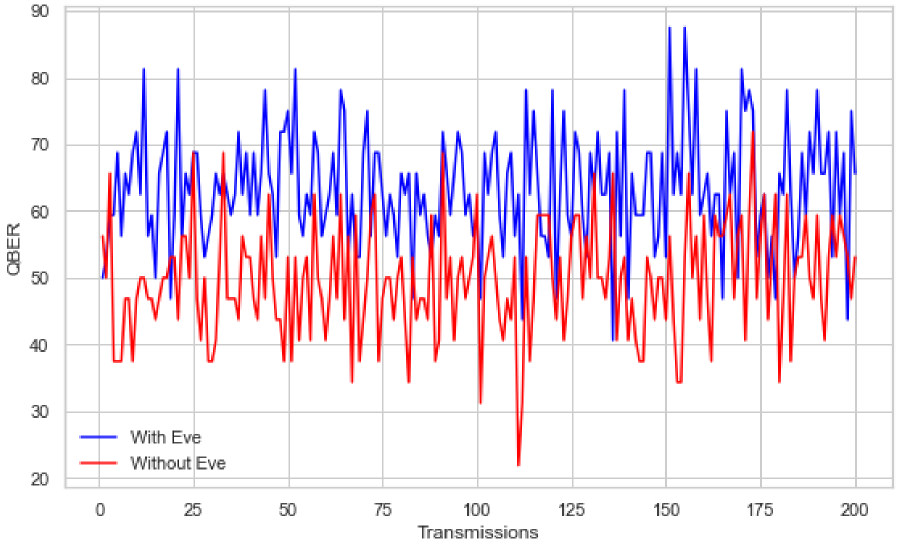


Figure 17: BB84 QBER as a function of the number of transmissions with and without Eve

Figure 17 illustrates the QBER as a function of transmissions in two different scenarios, one with eavesdropping and one without. In both scenarios, QBER follows a similar trend over the range of broadcasts, albeit on different scales. Furthermore, in the case without eavesdropping, the average QBER is around 50, a fact implying constant performance in the absence of outside interference. By comparison, the Eve scenario indicates a significant rise in the QBER, which falls between 60 and 70. This discrepancy demonstrates the influence of eavesdropping, as Eve's presence continuously raises the QBER. Despite this increase, the QBER in both circumstances remains rather stable throughout the course of transmissions, implying that, while Eve's presence reduces performance, the overall QBER behaviour does not change drastically.

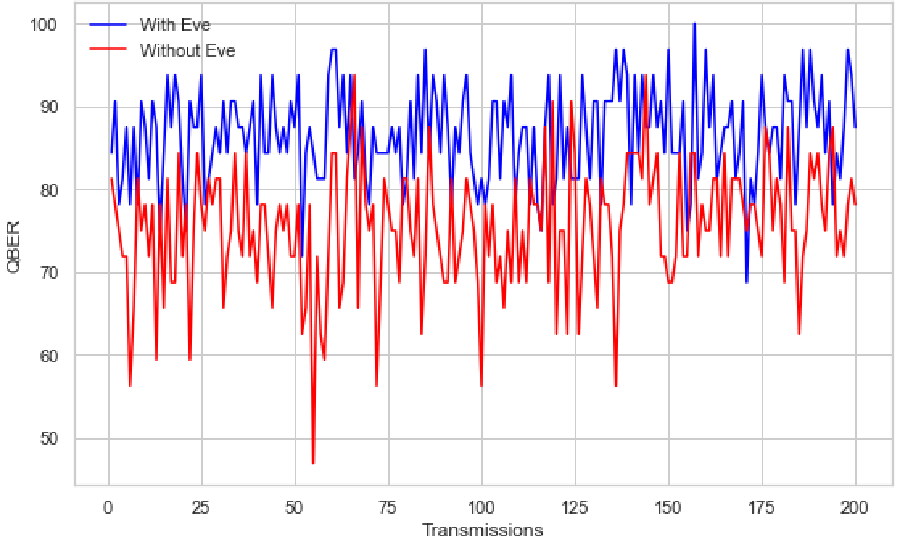


Figure 18: B92 QBER as a function of the number of transmissions with and without Eve

Figure 18 illustrates the QBER as a function of transmissions for the B92 protocol with and without eavesdropping. Without Eve, the average QBER in the scenario is between 70 and 80. Eve's presence greatly increases the QBER with occasional spikes to 100. As anticipated eavesdropping can be identified by observing the QBER of the communication.

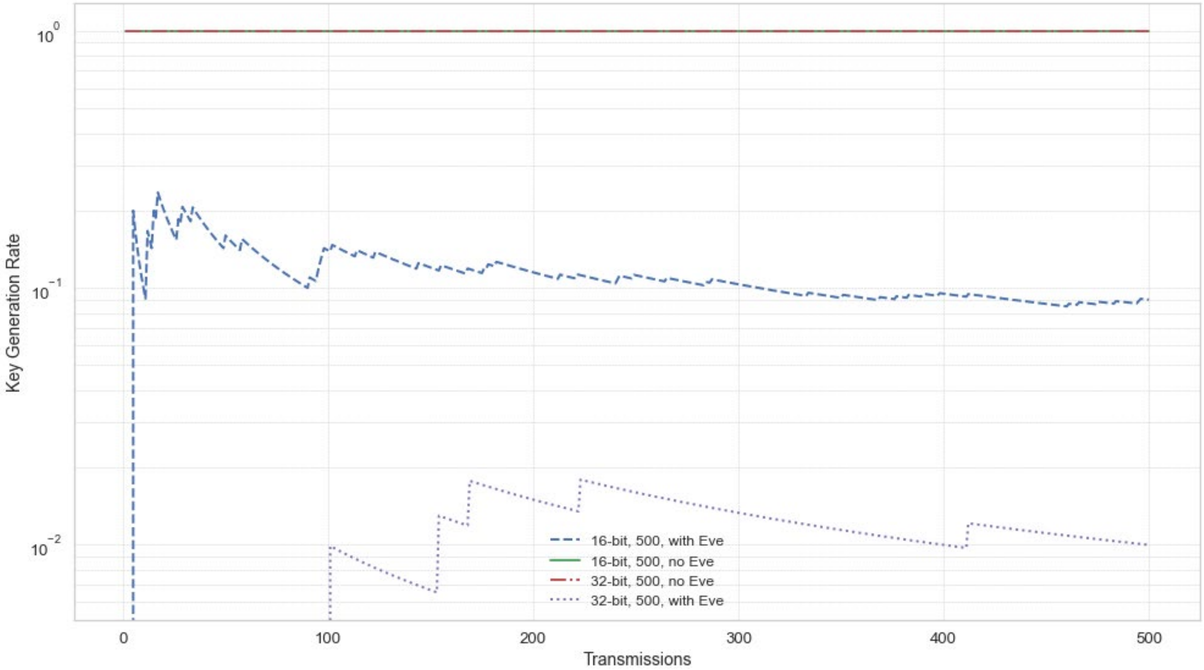


Figure 19: BB84 key generation rate as a function of the number of transmissions with and without Eve for different key lengths

Figure 19 compares the key generation rate to the number of transmissions in a QKD system using the BB84 protocol, by representing four distinct scenarios: (i) 16-bit key length without eavesdropping, (ii) 16-bit key length with eavesdropping, (iii) 32-bit key length without eavesdropping, and (iv) 32-bit key length with eavesdropping. The situations without eavesdropping show the greatest key generation rates, notably for 16-bit and 32-bit key lengths, which both follow the same trend. This closeness implies that the beginning key length has no effect on the generation rate when no eavesdropping is present, enabling the system to create keys effectively regardless of the starting key length. When eavesdropping is present, however, the initial key length has a far greater influence. The 16-bit key length with eavesdropping has a lower key generation rate than its non-eavesdropping counterpart. The 32-bit key length with eavesdropping has the lowest key generation rate, with values around 0, suggesting substantial interruption due to eavesdropping. The trajectories show that when the starting key length grows, the key production rate reduces significantly in the presence of eavesdropping. This shows that larger initial key lengths make the existence of an eavesdropper more visible, affecting key generation performance. Finally, this figure shows how the key length and the existence of eavesdropping affect the key generation rate in QKD systems. Eavesdropping considerably reduces performance, particularly as key length decreases, emphasizing the difficult balance between key size and security in quantum communication.

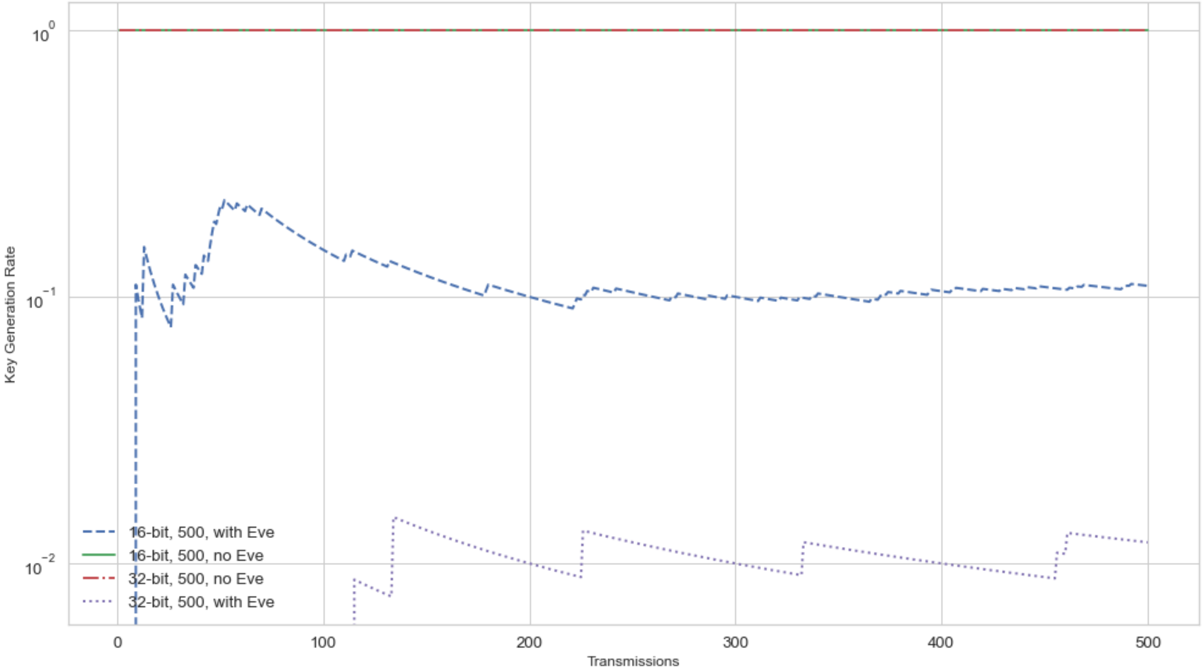


Figure 20: B92 key generation rate as a function of the number of transmissions with and without Eve for different key lengths

Figure 20 shows the key generation rate against the number of transmissions in a QKD system using the B92 protocol, under the same four different scenarios as Figure 19. In a scenario lacking eavesdropping, the highest key generation rates are found, particularly in situations with 16- and 32-bit key lengths. Both cases show similar trends, indicating that when there is no eavesdropper, the starting key length has no major effect on the key generation rate, enabling the system to function effectively across varied key lengths. However, when eavesdropping is present, its influence is far more noticeable. The key generation rate for the 16-bit key length with eavesdropping is considerably lower than in the non-eavesdropping situation. On the other side, the 32-bit key length scenario with eavesdropping has the most effect, demonstrating how disruptive eavesdropping can be when paired with a short starting key length. These findings indicate that raising the first key length greatly reduces the key creation rate under eavesdropping settings. This pattern means that greater key lengths make eavesdropping operations more visible, hence complicating the key generation process. The sharp fall in key production rates, particularly in the 32-bit eavesdropping scenario, demonstrates the increasing sensitivity of larger key lengths to eavesdropping.

3.2. Communication Interfaces

QKD devices are composed of layers. The key distribution is performed over the Quantum Channel, a dark fibre where the devices exchange quantum states on photons with extreme sensitivity to medium imperfections. Two other dark fibres, known as the classic channel, are needed to perform synchronization tasks to support the quantum channel information exchange.

However, the service layer is where the Secure Application Entities are, like the Base Station applications, and they do not directly interact with the Quantum and Classical channels. To manage the cryptographic material generated by QKD and store a buffer to be consumed by classical applications as a service, the QKD devices are packed with an integrated Key Management System. This software layer acts as an intermediate between the key consuming applications and the Quantum information exchange.

Since 2010, ETSI (European Telecommunications Standards Institute) has issued a set of standards for QKD vendors with the aim of unifying all interfaces related to this technology. Among others, there are two specifying how a Secure Application Entity can interact with the classical KMS layer to request a key of a specific length:

- **ETSI GS QKD 004:** Quantum Key Distribution (QKD); Application Interface
- **ETSI GS QKD 014:** Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API

Both interfaces serve the same goal, the main difference is regarding the implementation details. Since ETSI-004 is implementation agnostic and provides the concept of sessions, ETSI-014 is specified as a REST API with Mutual Authentication enabled. In many cases, choosing one or another depends on which is supported by the QKD device vendor. In this case, the equipment used for the experiments is only compatible with ETSI-014. Because of this, the demo applications have been developed following this standard to interact with the hardware.

The complete specification of ETSI-014 API can be accessed publicly in the ETSI repository [25].

The interface is enforced with mutual TLS to protect the KMS from unauthorized users. This means that all Secure Application Entities, like each Base Station, need to own a client certificate issued by the Public Key Infrastructure hosted at QKD modules. Within NANCY, these certificates have been previously deployed as PFX files within the source files of each application in the Docker containers.

In addition to the Quantum and Classical channels, each QKD module must be available via LAN with the Secure Application Entities for which provides service. Within NANCY, each application Docker runs in a machine on the same IP subnet as each QKD module.

3.3. Unit Testing

The QKD experiments carried on within the NANCY project framework must provide useful insights for improving the integration of commercial equipment in actual telecommunication networks. In order to stress the experimentation testbed in a systematic manner, an automated script has been developed to reproduce a continuous communication flow with different parameters of time intervals and total duration.

The script operates as an automation layer that manages the functionality of the application through its REST API, serving as an alternative to manual operation via the graphical user interface. More precisely, the script runs the following actions:

- Init loop N
    - Set an example connection data payload to be securely sent
    - Request BS1 API to encrypt the payload
        - BS1 requests ALICE for an encryption key
        - BS1 encrypts the payload with the key in AES256
        - BS1 returns to the client
            - The KeyID
            - The description
            - The cyphertext
            - The response time of the KMS
    - Request BS2 API to add a new encrypted connection data in its database
    - Request BS2 API to decrypt the previously sent data
        - BS2 fetches the connection data from the database
        - BS2 requests BOB for the decryption key with the associated KeyID in the record
        - BS2 decrypts the payload with the key in AES256 and stores the clear text in the corresponding database record
        - BS2 returns to the client
            - The status (OK, ERROR)
            - The response time of the KMS
    - Request BS2 API to fetch the decrypted record
        - BS2 fetches the record from the database
        - BS2 returns to the client
            - All record columns
    - Show the decrypted connection data by terminal
    - Stores in a log file the timestamp and the KMS response time for analysis
    - Wait s seconds
- End loop

The script will be used as the main tool to operate the infrastructure during the experiments, the execution conditions of the test are specified in Table 5.

Table 5: Execution parameters of the tests

Parameter	Value
Duration	1 hour
Wait seconds between each round	0 seconds
Number of rounds	Unlimited within test duration (1 hour)
Key length	256 bits

A set of parameters described in Table 6 will be recorded along with the execution of the experiments. The values gathered during the experiments will be used to plot them in charts as the outcomes to extract conclusions.

Table 6: Experiment parameters to measure

Layer	Parameter	Description
QUANTUM	QBER	Reports the Quantum Bit error rate measured
	KEYRATE (BITS/S)	Reports the average bit rate for a block
CLASSIC (KMS)	KEY AVAILABILITY TIME (MS)	The time (ms) to retrieve a key from ALICE and BOB from the apps

	KEY BUFFER LEVEL (%)	Reports the key filling percentage of the cryptographic material on each KMS (Alice and Bob)

Test scenarios

The experiments will be carried out on different deployment scenarios, the goal is to find performance deviations in specific setups and protocols supported by the QKD devices.

- Infrastructure
  - Actual commercial QKD equipment
  - Simulator
- QKD Protocols
  - Coherent-One-Way 3 States
  - Coherent-One-Way 4 States
- Fibre deployment (dark fibre)
  - 1 Km (actual deployment in Tecnalia premises)
  - 2 Km (Fibre launch lead in the laboratory)
  - 3,5 Km (1,5 Km + 2 Km fibre launch leads in laboratory)
- Operation
  - Normal
  - With eavesdropping presence

The results of the test performed with the parameters of Table 4 will be compared in order to get insights about the service performance over the different scenarios.

3.4. Deployment

The commercial equipment used for the experiments is the Clavis3 QKD Platform made by IDQuantique. The platform comprises two physical appliances: a sender (ALICE) represented in Figure 21, and a receiver (BOB), represented in Figure 22.

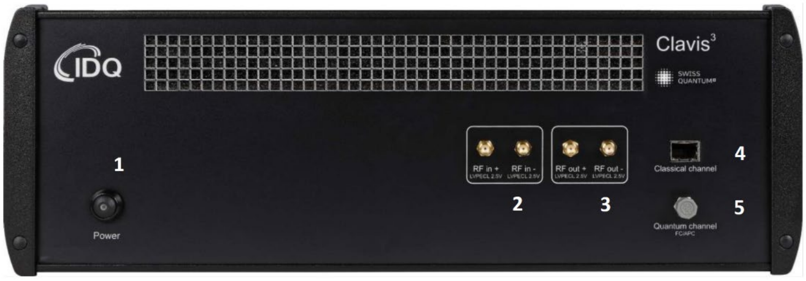


Figure 21: QKD ALICE

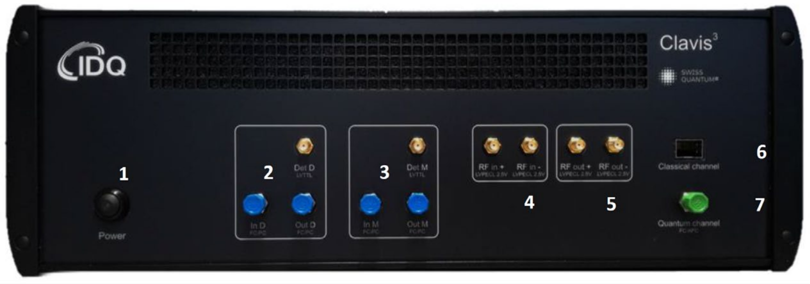


Figure 22: QKD BOB

All technical details are included in Table 7.

Table 7: QKD Equipment technical details

Attribute	Value
Vendor	IDQuantique
Model	Clavis3
Type	Prepare-and-measure
Supported protocols	Coherent-One-Way 3 States Coherent-One-Way 4 States
Power consumption	100-240 V; 50-60 Hz; 4.5 – 2 A, autosensing
Operating temperature	from 20° to 30°C
Dimensions	3.5U, 144 mm (H) x 402 mm (W) x 424 mm (D)
Weight	approx. 15 kg
ALICE Specifications	Inputs: RF in+/RF in -: differential electrical input, LVPECL, 2.5 V. Outputs: • RF out+/RF out -: differential electrical output, LVPECL, 2.5 V, 125MHz clock. • Quantum channel: optical output, FC/APC connector, 1310 or 1550nm, <1mW output power. • RJ45 connector: 100BaseT Ethernet. • USB: for upgrade and copy logs. I/O: Service channel: SFP optical 2.5Gbps transceivers, proprietary protocol.

BOB Specifications	Inputs: • RF in+/RF in -: differential electrical input, LVPECL, 2.5 V. • Quantum channel: optical input, FC/APC connector, 1310 or 1550nm. • Det D: single-end electrical input, LVTTL. • Det M: single-end electrical input, LVTTL. • In D: optical input, FC/PC connector. • In M: optical input, FC/PC connector. Outputs: • RF out+/RF out -: differential electrical output, LVPECL, 2.5 V, 125MHz clock. • Out D: optical output, FC/PC connector, 1310 or 1550nm. • Out M: optical output, FC/PC connector, 1310 or 1550nm. • RJ45 connector: 100BaseT Ethernet. • USB: for upgrade and copy logs. I/O: Service channel: SFP optical 2.5Gbps transceivers, proprietary protocol.

The equipment is deployed in the server rooms of two separate buildings of Tecnalia premises in Derio (Spain). The municipality owns the fibre deployment, and the only requirement was three dark fibres without any active switching to avoid affecting the quantum states. Figure 23 shows a satellite map of the deployment. The blue line in the map represents the approximate fibre optics lines connecting the two buildings.

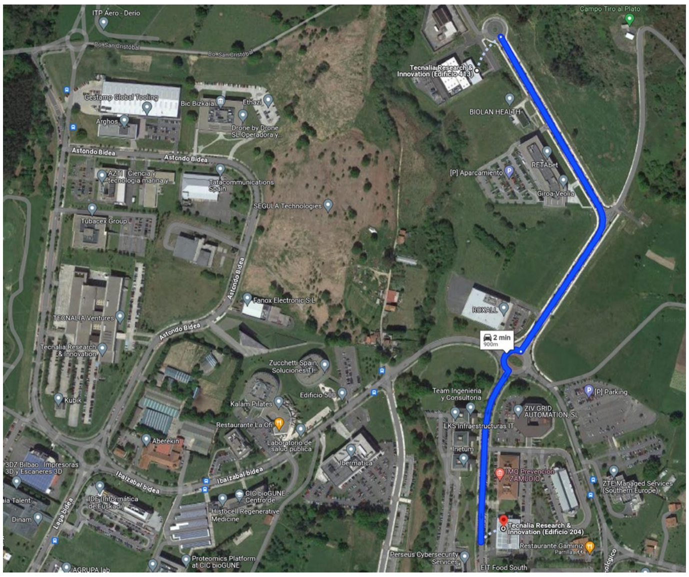

ITP Aero - Derio
Campo Tiro al Plato
Bo San Cristóbal
Tecnalia Research & Innovation (Edificio 413)
Gestamp Global Tooling
Bic Bizkaia
Ethazi
Arghos
Drone by Drone SL Operadora y...
BIOLAN HEALTH
Astondo Bidea
AZTI | Ciencia y tecnología marina y...
Tatacommunications Spain
[P] Aparcamiento
RETAbet
Giroa-Veolia
SEGULA Technologies
Tubacex Group
Fanox Electronic S.L
ROXALL
TECNALIA Ventures
Tecnalia Research & Innovation
2 min
900m
Zucchetti Spain Soluciones TI...
Edificio 500
Team Ingenieria y Consultoria
[P] Parking
Kubik
Kalam Pilates
ZIV GRID AUTOMATION, SL
Restaurante La Ofi
LKS Infraestructuras IT
Aberekin
Laboratorio de salud publica
Inetum
Ibermatica
3DZ Bilbao | Impresoras 3D y Escaneres 3D
Ibaizabal bidea
IMQ Prevencion - ZAMUDIO
ZTE Managed Services (Southern Europe)
IDE (Informática de Euskadi)
CIC bioGUNE - Centro de...
Histocell Regenerative Medicine
Dinam
Laga bidea
Tecnalia Research & Innovation (Edificio 204)
Restaurante Gaminiz Parrilla
Proteomics Platform at CIC bioGUNE
Perseus Cybersecurity Services
EIT Food South
AGRUPA lab

Figure 23: QKD Fibre deployment

To extend the existing fibre deployment to reach larger distances, two fibre launch leads, one 1.5 Km and another 2 Km, were used in the laboratory. The laboratory setup can be seen in Figure 24. To reach the maximum length, both were connected through an optical splitter shown in Figure 25.

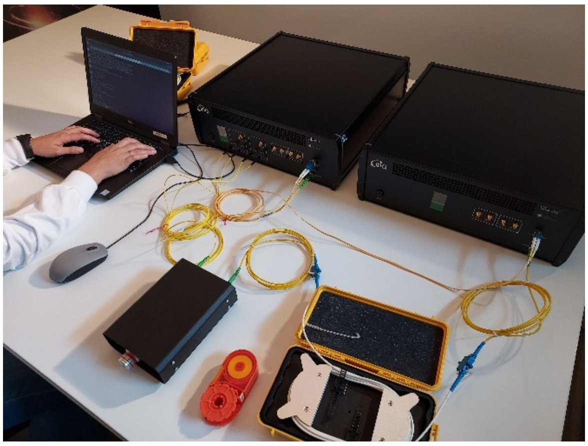


Figure 24: QKD in laboratory

Figure 25: Fibre launch leads with splitter

The equipment reports parameter statistics in real-time to a server running Quantum Management System (QMS), a monitoring software. This server is a virtual machine running the software in Docker containers provided by the vendor. The web application run by the QMS provides a Graphic User Interface to set up the platform and visualize all the parameters measured by the systems (QBER, KeyRate, KMS, etc.). This tool is essential for obtaining the measurements after the experimentation ends.

The QMS only has a connectivity requirement, which is to have IP communication with each device. As both ALICE and BOB have been deployed in Tecnalia premises both are connected by ethernet to a subnet open to the QMS server. A graphical representation of the network can be found in Figure 11.

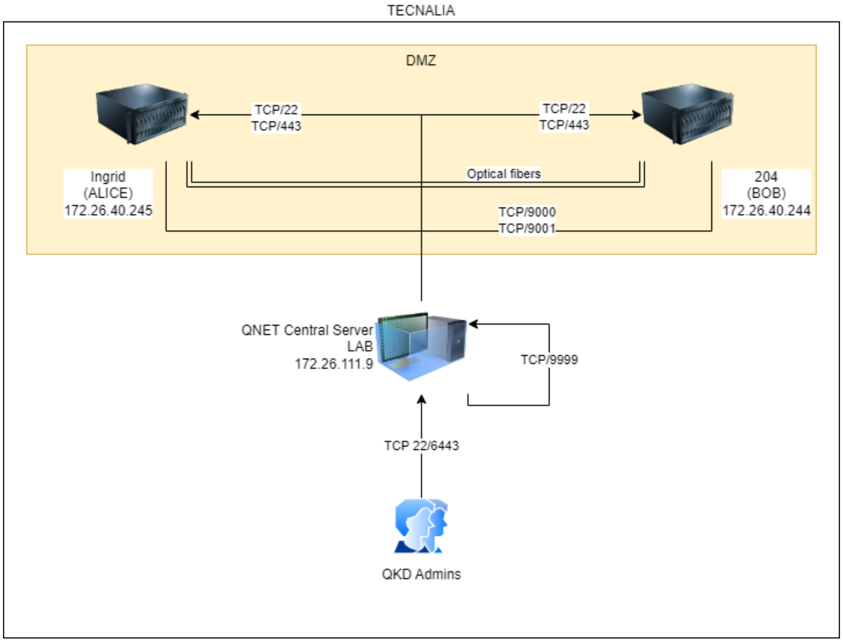


Figure 26: Tecnalia QKD network

Eavesdropping simulator

To perform the eavesdropping tests the vendor IDQuantique provide an additional tool for this purpose. The eavesdropping simulator can be seen in Figure 27.

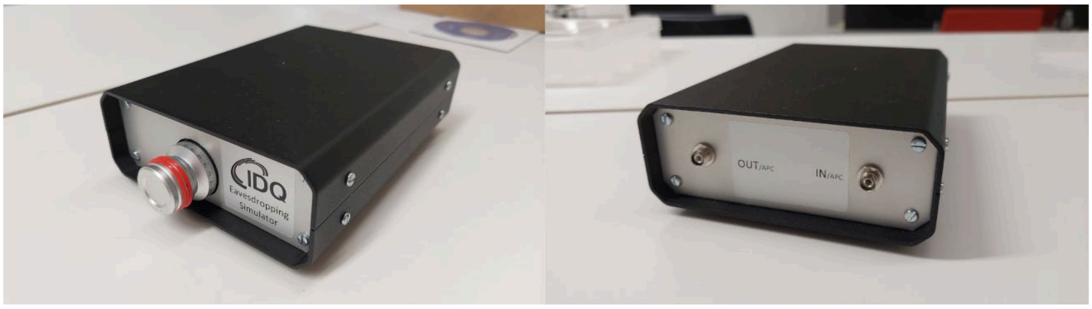


Figure 27: Eavesdropping simulator

The simulator internally works as a tiny mirror that, depending on the regulator knob, reflects a portion of the light of the fibre connected to it. The device simulates a specific kind of attack that can affect QKD, PNS, where an attacker steals a small fraction of the photon pulses sent over the fibre to perform measurements without affecting the rest of the photons to remain undetected.

3.5. Pending Actions and Planning

The experimentation activities carried out under the current task contribute to the state-of-the-art in terms of performance analysis between two of the most common QKD protocols used in commercially available equipment. In addition, the Coherent-One-Way simulator represents an invaluable tool for mocking up the Quantum Channel and classical communication. In further research, the simulator results can be compared with a wider range of experiment topologies, such as different link sizes or WDM multiplexing techniques if they are supported by the hardware. The results achieved in the experiments found the COW3 & 4 protocol behaving in the described testbed in a very similar performance level as the simulator achieved, demonstrating how the protocol modelling fits the actual equipment characteristics.

4. PQC Signature

4.1. Introduction to TDIS PQC Signature Solution

TDIS is developing a PQC Digital Signature Solution composed of:

- **A TDIS PQC Signature Token:** it consists of a smart card integrating a quantum-resistant digital signature algorithm. This token can be used for public key infrastructure (PKI) applications such as identity cards and corporate security (closed user groups). TDIS follows the ongoing initiative from NIST to standardise a set of quantum-resistant algorithms. This feature can be used to ensure the integrity and authentication of any data files. For this purpose, asymmetric key pairs are used: the private keys are stored in the token and used to sign. The signature is internally processed and based on the digest of the data file. On the other hand, the associated public keys, certified by a certification authority to allow the receiver to check the signature and, thus to verify the authenticity and integrity of the file. The certificates may be stored in the token or present on a server in the cloud.
- **A TDIS PQC Signature Middleware:** this middleware (or driver) provides minimal services to the applications for interfacing with the token.

Link from ELECTRON project

As explained in section 3.1.5 of D3.1 “NANCY Architecture Design”, TDIS was participating, from October 2021 to September 2024, in ELECTRON H2020 Project[5] linked with energy management (EPES). As a result, a first version of the PQC Hybrid Digital Signature Solution has been provided.

This initial Solution has validated the following innovations:

- Frugal implementation of PQC Digital Signature on tiny CPU devices environment (32bits CPU, 24kB RAM)
- Selected algorithm: Crystals Dilithium-AES, security level 2
- High secure implementation of the cryptographic algorithm including countermeasures against state-of-the-art attacks (side channel, fault injection attacks)
- Acceptable performance compared to classical cryptography
- Validation of a hybrid concept that consists of a combination of pre-quantum and post-quantum cryptographic algorithms

During this ELECTRON project, TDIS followed closely the NIST PQC standardization process, along with the recommendations from national security agencies (ANSSI, BSI) with the combined objectives of security and interoperability.

During the course of the ELECTRON project, new events took place at the NIST standardization level, such as the adoption of a different variant Crystals Dilithium-SHAKE and the recommendation to target a higher security level (above level 2). This evolution is now formalized under the NIST FIPS 204 “Module-Lattice-Based Digital Signature Standard”. Another event coming from national security agencies is the strong recommendation to implement a capability for Crypto Agility.

To cover these gaps, within the NANCY project, TDIS’s objective is to work on the following innovations on the PQC Digital Signature component. Starting from the existing component, TDIS will develop the Crystals Dilithium SHAKE with Security Level 3 as recommended by NIST.

[5] ELECTRON – project (electron-project.eu)

The new challenges for this innovation will be to keep the same requirement constraints while implementing a much stronger PQC algorithm:

- Frugal implementation
- High secure implementation
- Acceptable performance

Moreover, TDIS targets to bring a novel mechanism for Crypto Agility.

4.2. Architectural Design & Implementation

4.2.1. Architecture Overview of the PQ Digital Signature Solution

As shown in Figure 28, the solution is composed of two elements:

- TDIS PQC Signature Token: smart card implementing the PQC digital signature
- TDIS PQC Signature Middleware: contains the Driver offering easier access to the token from the upper Applications

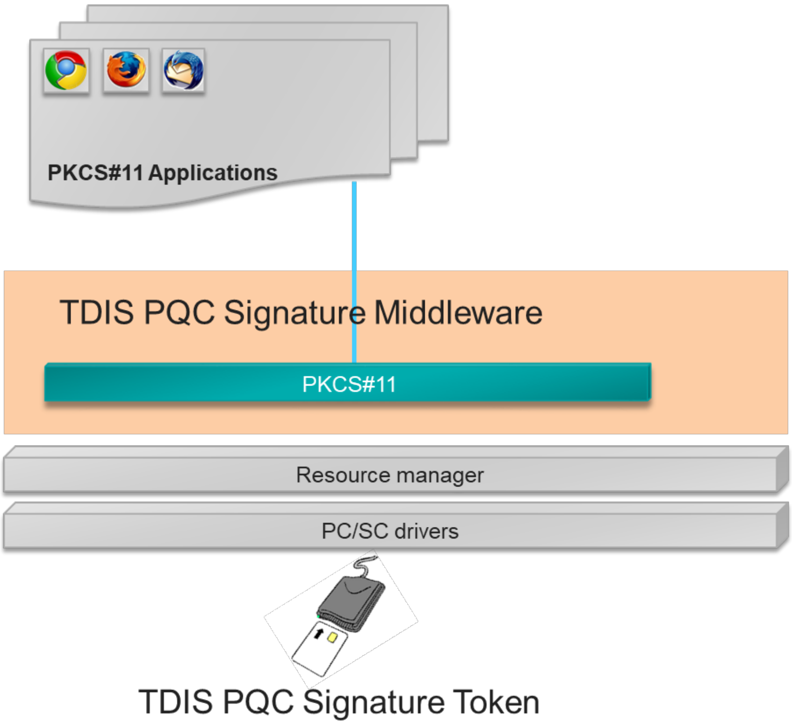


Figure 28: Architecture of the PQC Signature Solution

PQC Signature Middleware

The PQC Signature Middleware published to the Application layer a set of APIs commonly used in PKI systems namely PKSC#11[6] and used for digital signature purposes. The provided PQC Signature Middleware is running on a Linux PC. Resource Manager & PC/SC drivers are the market low-layer drivers to facilitate access to the smart cards. The communication interface with the smart cards can be Contact (ISO/IEC 7816) or Contactless RFID (ISO/IEC 14448).

PQC Signature Token

Figure 29 Architecture of the PQC Signature Token represents the architecture of the PQC signature device.

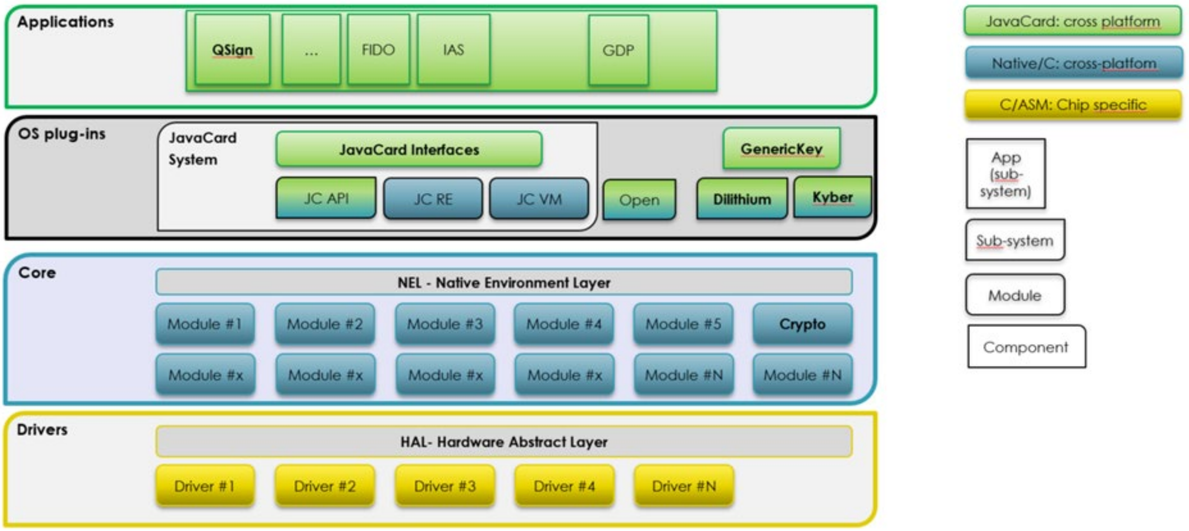


Figure 29: Architecture of the PQC Signature Token

At the top of the diagram, we see the **application layers**, including the target electronic signature application, named **QSign** (Quantum Signature) managing all the PKI operations. This application will rely on **JavaCard APIs** (standard and proprietary), implemented by the upper layers of the **Operating System**.

At the lower layers, we find the **cryptographic primitives** Dilithium and Kyber. Finally, we find the **Hardware Chip** containing the CPU, RAM and NVM memories, Cryptographic hardware accelerations, security sensors, etc. The **communication interface** with the external system can be contacted (ISO/IEC 7816) or contactless RFID (ISO/IEC 14448).

4.2.2. Applicable standards

Several standards in the Smart Cards domain are impacted by the introduction of PQC cryptography.

FIPS 204: Lattice-Based Digital Signatures

FIPS 204[7] covers lattice-based digital signature algorithms, particularly CRYSTALS-Dilithium.

NANCY PQC Signature Token applies for this standard.

[6] Workspace Home - OASIS (oasis-open.org)

[7] FIPS 204, Module-Lattice-Based Digital Signature Standard | CSRC (nist.gov)

PKCS #11

The PKCS #11 standard defines a platform-independent API to cryptographic tokens, such as hardware security modules (HSM) and smart cards.

The API defines the most commonly used cryptographic object types (RSA keys, X.509 certificates, DES/Triple DES keys, etc.) and all the functions needed to use, create/generate, modify and delete those objects.

The existing version of this standard does not address the new PQC cryptographic objects, in particular the hybrid signature concept.

TDIS is registered within OASIS-Open[8] and will contribute to extending this standard.

ISO/IEC 7816

The International Organization for Standardization (ISO) manages jointly with the International Electrotechnical Commission, a set of standards namely ISO/CEN 7816-x related to electronic identification cards with contacts, especially smart cards, and more recently, contactless mobile devices.

A new Ad Hoc group was created to work on the PQC aspect. TDIS led this ad hoc group and provided inputs. Resulting Standard ISO/IEC 7816-8:2021/Amd 1:2023 has been publicly published in November 2023

JavaCard

JavaCard is a software technology that allows Java-based applications (applets) to be run securely on smart cards.

TDIS is registered within JavaCard Forum[9] and contributes to extending this standard to PQC cryptographic Java APIs.

Global Platform

GlobalPlatform[10] has standardized isolated execution environments in different types of devices (such as Secure Elements (SEs) and Trusted Execution Environments (TEEs)), to deliver secure services and trusted storage for diverse industries and stakeholders.

TDIS is registered within Global Platform and contributes to extending this standard to support PQC.

4.2.3. NIST PQC standardization process

Uncertainties due to FIPS 204 standardization work:

In TDIS, the development of the library was anticipated from 2022 (ELECTRON project), before the publication of this FIPS 204 (August 2024). In order to limit the risk of deviation from the future standard draft to be published, the strategy implemented was to scrutinize the PQC forum[11] which centralizes most of the technical discussions around the NIST PQC competition and the various calls for proposal candidacy.

[8] Workspace Home - OASIS (oasis-open.org)

[9] Java Card Forum

[10] GlobalPlatform Homepage - GlobalPlatform

[11] pqc-forum – Google Groups

CRYSTALS-Dilithium was thus renamed ML-DSA. The different associated security levels will now be denoted by ML-DSA-44 (level 2), ML-DSA-65 (level 3) and ML-DSA-77 (level 5). CRYSTALS-Kyber has been renamed ML-KEM. The different associated security levels will now be denoted by ML-KEM-512 (level 1), ML-KEM-768 (level 3) and ML-KEM-1024 (level 5).

TDIS library has therefore been brought into compliance with the draft standard. Unfortunately, the uncertainties cannot be completely resolved because the final version of the standard should not be published until the end of 2024.

NIST “On-ramp process for Signature”

As explained in 2.2, after the 3 first-round selections, NIST has started the “On-ramp process” for signature algorithms[12]. For that, NIST issued a new call for additional signatures with the following scopes:

- NIST is primarily interested in additional general-purpose signature schemes that are not based on structured lattices
- NIST may also be interested in signature schemes with short signatures and fast verification
- Any lattice signature would need to significantly outperform CRYSTALS-Dilithium and FALCON and/or ensure substantial additional security properties.

Within its research activities**, TDIS submitted 2 additional candidate algorithms (VOX and PROV) to the NIST ”On-ramp process”** in early 2023. Since this date, TDIS provided continuous support during the evaluation process through Questions & Answers for those candidates during the whole years 2023 and 2024.

4.2.4. Implementation of PQC Signature

Design of the PQC library

The only initial repository was the open source library made available by the authors of the Dilithium algorithm. This library only provides a functional implementation. The design of the TDIS library imposes industrial quality standards, particularly in terms of coding and testing rules. Furthermore, work has been carried out on the possibility of addressing security level2, level3 and even level5 with the same code. This allows greater agility according to the needs which will be expressed in the final specifications. A new interface has therefore been defined to allow level selection when initializing the Dilithium signing session.

Security and technological monitoring

The major expertise brought by TDIS in NANCY is the secure implementation of the Crystals-Dilithium implementation **against side-channel and fault attacks**.

Analysis of existing attack paths is a continuous activity from the cryptographic research community. TDIS Crypto Experts carry out constant technological monitoring. The subjects of efficient embedded implementations, attack paths and countermeasures algorithms are in constant motion, as it is highly scrutinized by the cryptographic community. In particular, the following publications have been considered: [26] , [27] , [28] , [29] , [30] , [31] , [32] , [33] , [34] , [35] , [36] , [37] , [38] , [39] , [40] , [41] , [42] , [43].

[12] Post-Quantum Cryptography: Digital Signature Schemes | CSRC (nist.gov)

The Crypto Engineering team worked on implementing countermeasures at each step of the algorithm, in order to cover all known attack paths. This implementation of countermeasures was carried out with monitoring of the impact on performance and more particularly on the memory footprint which is already critical for the project. As the performance of the Dilithium algorithm is not deterministic, the impact of countermeasures on performance cannot be easily defined globally. The impact is defined according to the number of internal turns in the algorithm. On our current prototype, the impact of the countermeasures was measured between 33% and 60% for a memory footprint cost of less than 1KB.

Optimization of RAM consumption

On the optimization of RAM consumption, thanks to knowledge sharing and brainstorming work within TDIS development experts, some new tracks of code optimization have been found and implemented.

The first gains in the cryptographic library from this optimization are as follows:

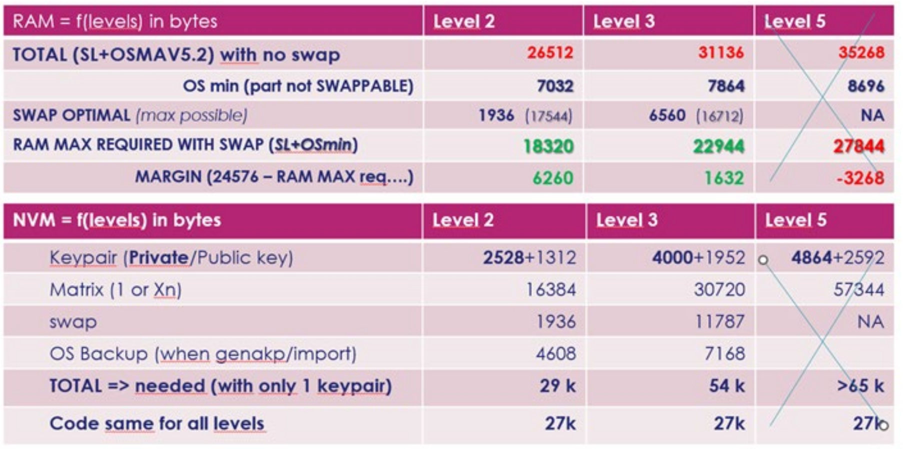
- -3KB for level2
- -5KB for level3
- -7KB for level5

Implementation of ML-DSA-65 (level 3) on real hardware chip:

The optimization described above on the RAM made it possible to integrate a “level 3” Dilithium signature on the current targeted hardware chip; however, “level 5” remains unattainable.

Despite the optimizations, moving to level 3 downgraded NVM consumption and performance. This is due to the larger key sizes and the memory consumption of the matrix which goes from 10kB (level 2) to 30kB (level 3) – in the non-optimized scheme, this matrix is recalculated in RAM. This optimization generates an additional cost of around 700 ms on the first signature (calculation and storage in NVM) or during a key change, the matrix being derived from the rho parameter specific to each key.

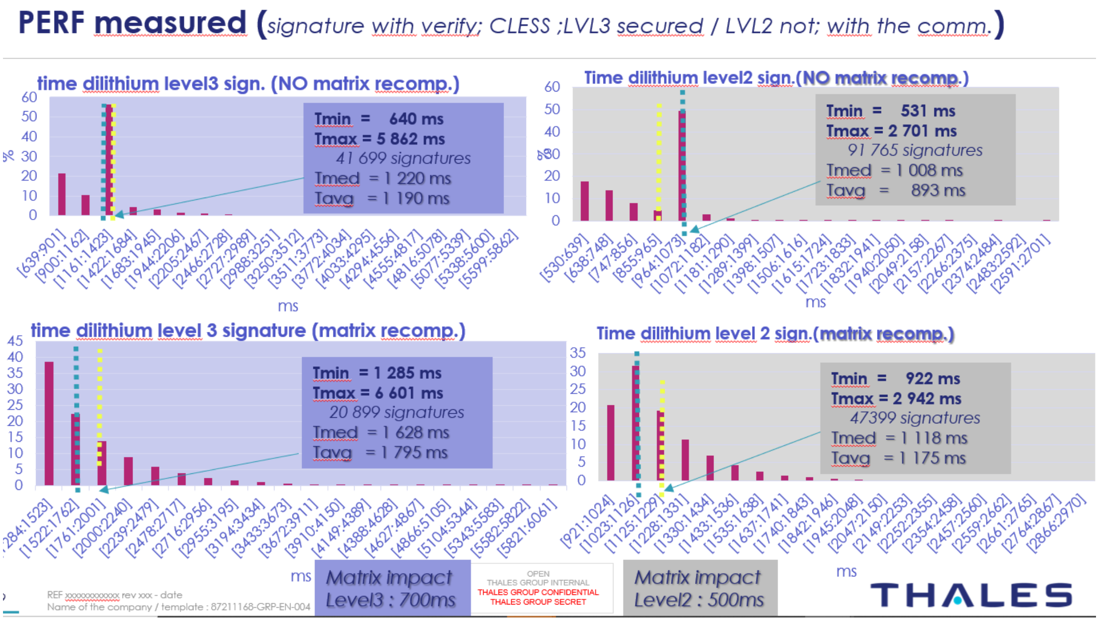
Figure 30 summarizes the measured memory consumption and Figure 30 Signature creation performance.



RAM = f(levels) in bytes	Level 2	Level 3	Level 5
TOTAL (SL+OSMAV5.2) with no swap	**26512**	**31136**	~~**35268**~~
OS min (part not SWAPPABLE)	**7032**	**7864**	~~**8696**~~
SWAP OPTIMAL *(max possible)*	**1936** (17544)	**6560** (16712)	~~**NA**~~
RAM MAX REQUIRED WITH SWAP (*SL+OSmin*)	**18320**	**22944**	~~**27844**~~
MARGIN (24576 – RAM MAX req....)	**6260**	**1632**	~~**-3268**~~

NVM = f(levels) in bytes	**Level 2**	**Level 3**	**Level 5**
Keypair (**Private**/Public key)	**2528**+1312	**4000**+1952	~~**4864**+2592~~
Matrix (1 or Xn)	16384	30720	~~57344~~
swap	1936	11787	~~NA~~
OS Backup (when genakp/import)	4608	7168	
TOTAL => needed (with only 1 keypair)	**29 k**	**54 k**	~~**>65 k**~~
Code same for all levels	**27k**	**27k**	~~**27k**~~

Figure 30: Memory Footprint



PERF measured (signature with verify; CLESS ;LVL3 secured / LVL2 not; with the comm.)

time dilithium level3 sign. (NO matrix recomp.)

Tmin = 640 ms
Tmax = 5 862 ms
41 699 signatures
Tmed = 1 220 ms
Tavg = 1 190 ms

Time dilithium level2 sign.(NO matrix recomp.)

Tmin = 531 ms
Tmax = 2 701 ms
91 765 signatures
Tmed = 1 008 ms
Tavg = 893 ms

time dilithium level 3 signature (matrix recomp.)

Tmin = 1 285 ms
Tmax = 6 601 ms
20 899 signatures
Tmed = 1 628 ms
Tavg = 1 795 ms

Time dilithium level 2 sign.(matrix recomp.)

Tmin = 922 ms
Tmax = 2 942 ms
47399 signatures
Tmed = 1 118 ms
Tavg = 1 175 ms

Matrix impact Level3 : 700ms

Matrix impact Level2 : 500ms

REF xxxxxxxxxxxx rev xxx - date
Name of the company / template : 87211168-GRP-EN-004

OPEN
THALES GROUP INTERNAL
THALES GROUP CONFIDENTIAL
THALES GROUP SECRET

THALES

Figure 31: Signature creation performance

These diagrams illustrate the non-uniform distribution of signature calculation times. For each security level (level 2 or 3), measurements are made with or without recalculation of the matrix:

- NO matrix recomposition: the signatures are made using the same key (and therefore without recalculating the matrix, only the message differs for each signature)
- WITH the matrix recalculated: for each signature, the key used is different and therefore, the matrix is recalculated accordingly.

Values presented are:

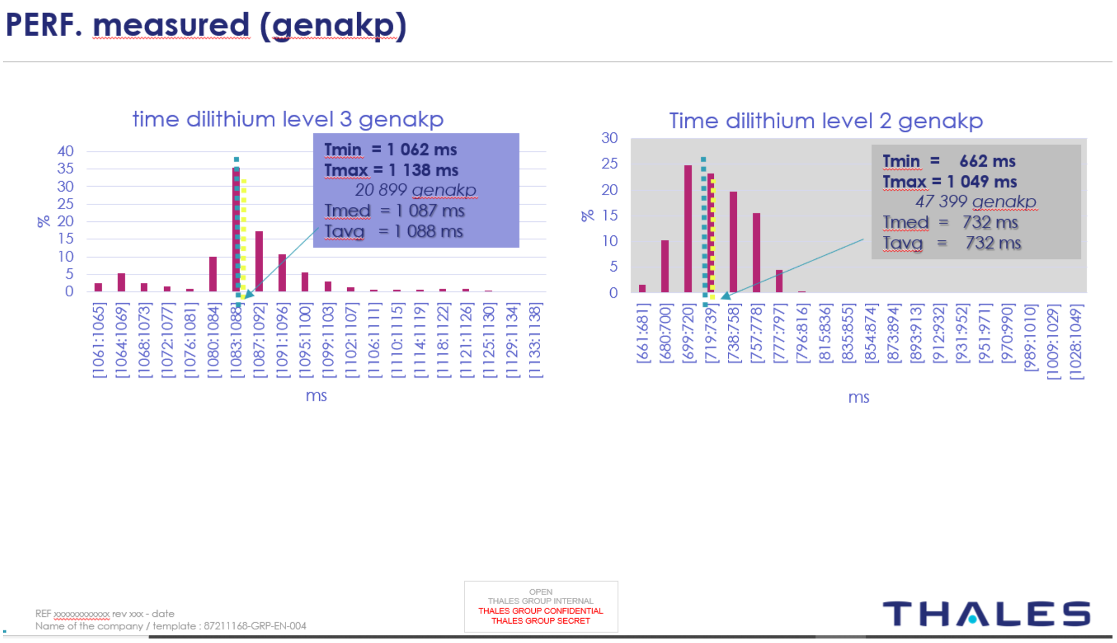
- Tmin: minimum time measured on all N signatures;
- Tmax: maximum time;
- Tmed: median
- Average time tavg. Despite a dispersion, the median shows acceptable performance.

It should be noted that the time slices are slightly different on the diagrams, the measurements having been carried out at different times. On the left “column”, the time ranges are incremented in steps of 262 ms (top part), and 140 ms on the bottom part. On the right “column”, the step is 110 ms (top part) and 103 ms on the bottom part.

The “Matrix Impact” time makes it possible to calculate the difference between the medians Tmed: from NO matrix recomp, compared to matrix recomp.

In addition to the evolution of the specification for “level 3”, the code has been more secure than for “level 2”. We estimate the penalties at 100 ms for these security measures alone, independent of the version of the algorithm used.

We also measured key generation times, shown in Figure 32 below.



PERF. measured (genakp)

REF xxxxxxxxxxxx rev xxx - date
Name of the company / template : 87211168-GRP-EN-004

OPEN
THALES GROUP INTERNAL
THALES GROUP CONFIDENTIAL
THALES GROUP SECRET

THALES

Figure 32: Key Generation performance

The left part uses steps of 5 ms, for 20ms on the right part. We conclude from these measurements that the key generation (genAkp) by the component is of low dispersion.

4.2.5. Implementation of Crypto Agility

Crypto Agility requirements

ANSSI provided a first definition of crypto-agility in a scientific opinion published on March 14, 2022[13], which encourages crypto agility of products. The BSI (German national security agency) had already introduced the concept in other notes[14].

The general principle must make it possible, beyond a simple update of code, to “switch” an issued product based on one type of cryptography, to one or more other algorithms, minimizing impact on existing functional configuration and parameters.

Design considerations

As shown in Figure 32, the architecture of the TDIS Token is based on JavaCard specifications. To date, the specifications offer an interface dedicated to each type of key. Figure 33 and Figure 34 illustrate this point.

[13] Avis de l'ANSSI sur la migration vers la cryptographie post-quantique | ANSSI (cyber.gouv.fr)
[14] Migration to Post Quantum Cryptography (bund.de)

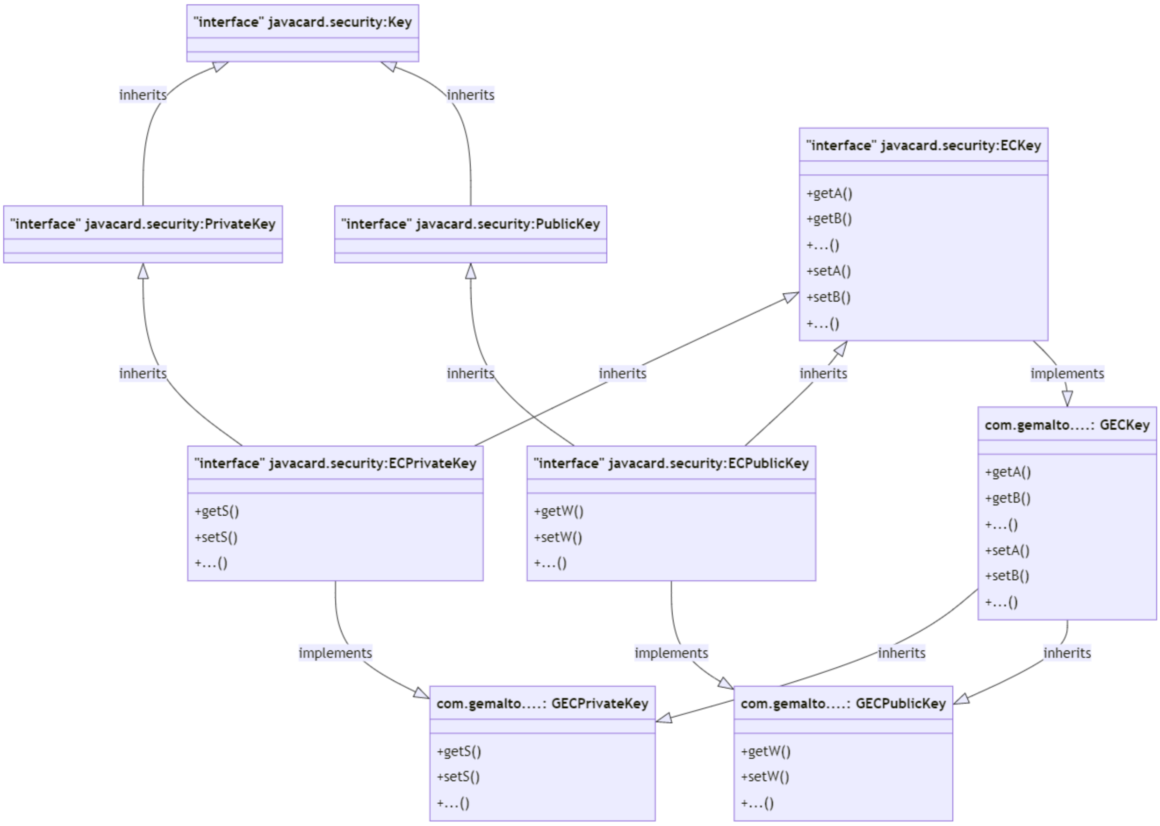

"interface" javacard.security:Key
inherits
inherits
"interface" javacard.security:ECKey
+getA()
+getB()
+...()
+setA()
+setB()
+...()
"interface" javacard.security:PrivateKey
"interface" javacard.security:PublicKey
inherits
inherits
inherits
inherits
implements
com.gemalto....: GECKey
+getA()
+getB()
+...()
+setA()
+setB()
+...()
"interface" javacard.security:ECPrivateKey
+getS()
+setS()
+...()
"interface" javacard.security:ECPublicKey
+getW()
+setW()
+...()
implements
implements
inherits
inherits
com.gemalto....: GECPrivateKey
+getS()
+setS()
+...()
com.gemalto....: GECPublicKey
+getW()
+setW()
+...()

Figure 33: Example of Apis for private and public EC keys by JavaCard

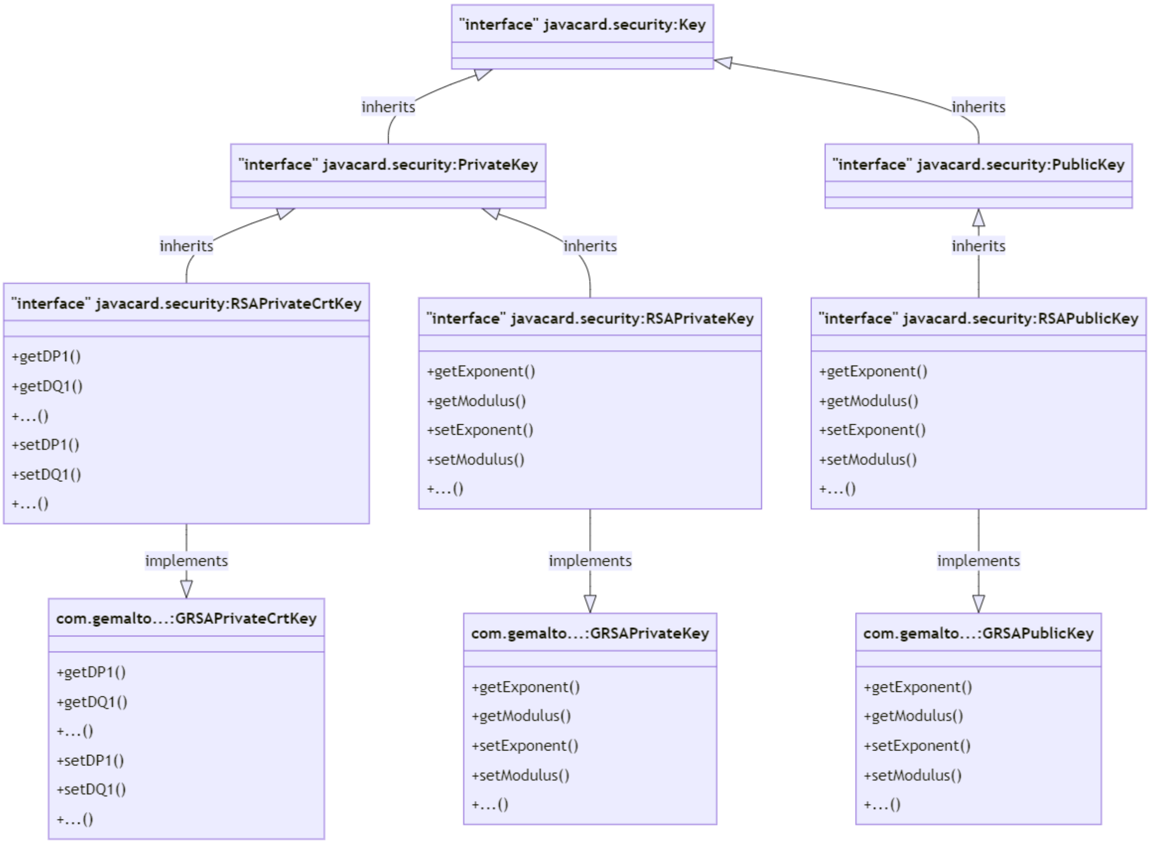


Figure 34: Example of Apis for RSA private and public keys by JavaCard

As shown, each interface (for example ECPrivateKey or RSAPrivateKey) defines its own “accessors” allowing a key element to be read or written (set or get).

This means that the introduction of new key types (e.g., in the case of a new algorithm) requires a modification of the application, and a new delivery of application code, which also implies a re-personalization of the application.

The new **Agility objective** is to address the following constraints: retain user data (avoid complete re-instantiation and re-personalization of applications) and minimize impacts on the overall code (application + operating system).

New Design APIs

As described above, the regular JavaCard APIs for key management are specialized by key types and do not support the new key structures for post-quantum algorithms. In order to facilitate the implementation and make the diffusion of new algorithms sustainable, we have defined an extended programming interface (API) generalizing a generic interface for all key objects.

The application, without code modification, can then manage key updates for future algorithms, with generic parameter access methods.

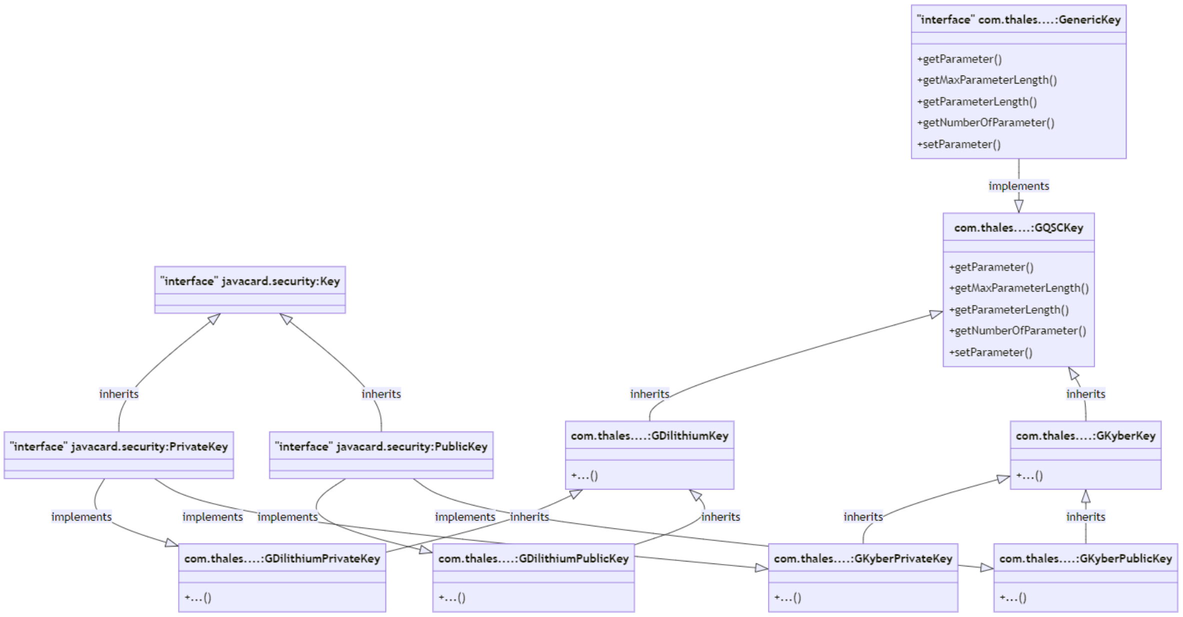


Figure 35: GenericKey interface

The new com.thalesgroup.javacardx.GenericKey interface defines five generic methods.

This work has been promoted to the JavaCard Forum for standardization.

New Architecture to support Crypto-Agility

The restructured new architecture supporting Crypto Agility is shown in Figure 36..

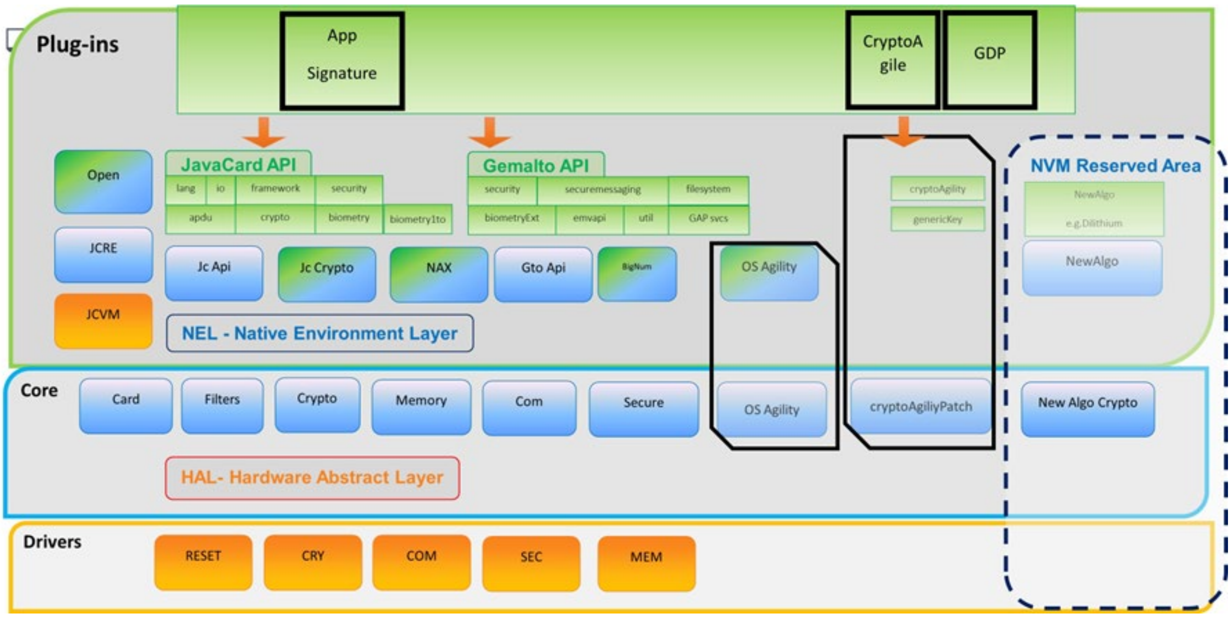


Figure 36: Architecture for Crypto Agility

New Components have been defined:

- **App Signature**: this is the Signature Application of the Token

- **OS Agility**: regroups all the software update functions. Its purpose is to manage the software patches to be downloaded and installed in the Token.
- **Crypto Agile Library**: regroups all the cryptographic algorithms including the new PQC algorithms. The New generic Key Interface described above is implemented in this component.

4.3. Communication Interfaces

4.3.1. PQC Signature Middleware

The PQC Signature Middleware publishes a set of APIs allowing upper Application to:

- Generate a new key pair
- Create a Signature

Figure 37 represents the sequence diagrams for the communication during a key pairs generation. This sequence is not used in NANCY Demonstrators as PQC Signature Tokens are personalized in TDIS factory.

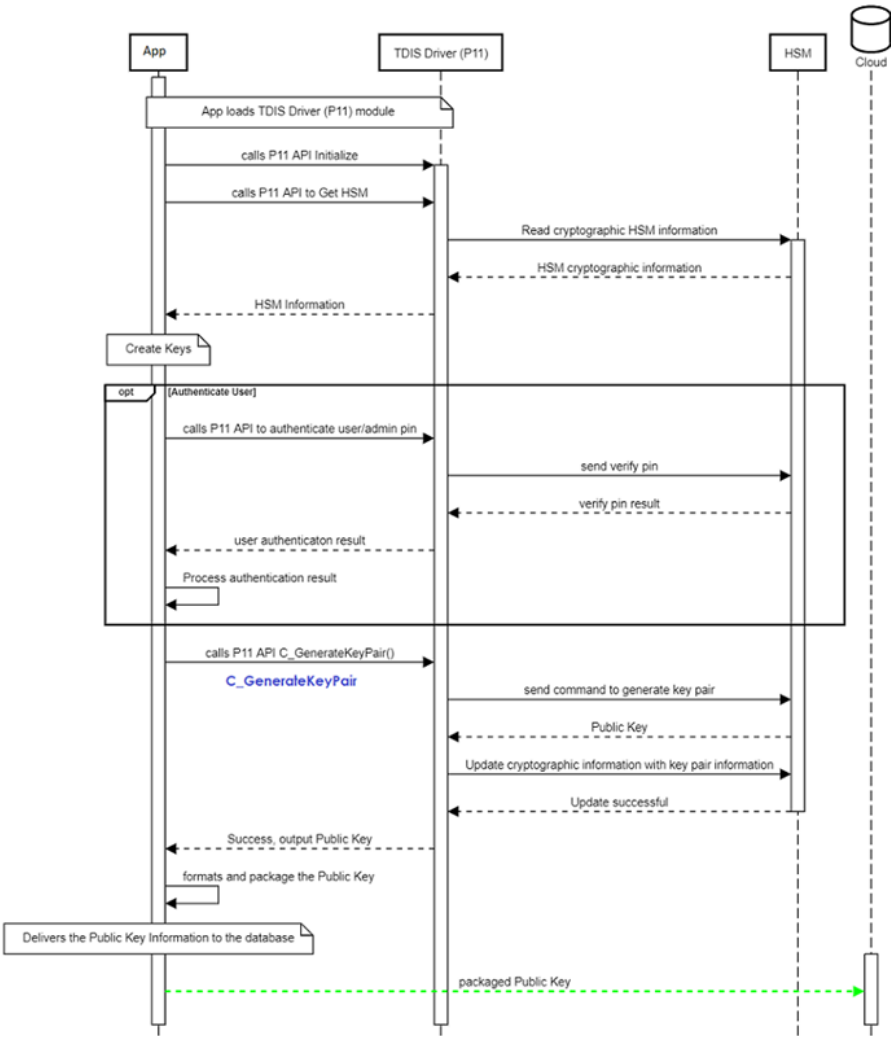

App
TDIS Driver (P11)
HSM
Cloud
App loads TDIS Driver (P11) module
calls P11 API Initialize
calls P11 API to Get HSM
Read cryptographic HSM information
HSM cryptographic information
HSM Information
Create Keys
opt
[Authenticate User]
calls P11 API to authenticate user/admin pin
send verify pin
verify pin result
user authenticaton result
Process authentication result
calls P11 API C_GenerateKeyPair()
C_GenerateKeyPair
send command to generate key pair
Public Key
Update cryptographic information with key pair information
Update successful
Success, output Public Key
formats and package the Public Key
Delivers the Public Key Information to the database
packaged Public Key

Figure 37: Generate Key Pair

Figure 38 represents the sequence diagrams used each time a PQC signature is needed.

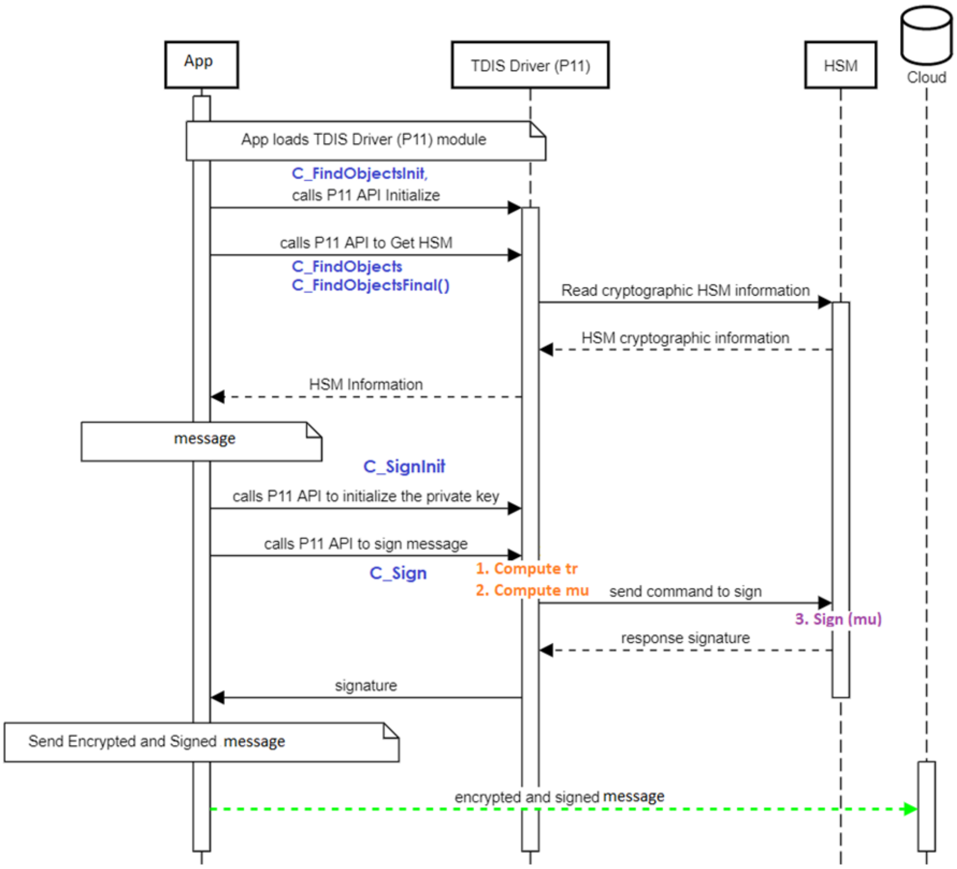


Figure 38: Signature Creation

4.4. Unit Testing

4.4.1. Test Plan

PKCS#11 Test suite that covers major functionalities of PKCS#11 API.

Classes of Tests for dilithium are:

- TC_NE_C_SignInit::testCKM_ML_DSA
- TC_NE_C_Sign::testCKM_ML_DSA
- TC_NE_C_SignUpdate::testCKM_ML_DSA
- TC_NE_C_SignFinal::testCKM_ML_DSA
- TC_NE_C_VerifyInit::testCKM_ML_DSA
- TC_NE_C_Verify::testCKM_ML_DSA
- TC_NE_C_VerifyUpdate::testCKM_ML_DSA
- TC_NE_C_VerifyFinal::testCKM_ML_DSA

4.4.2. Test Results

The test results are summarized in Table 8.

Table 8: Unit test results

Status	Number
Tests	646
FailuresTotal	0
Errors	0
Failures	0

4.5. Deployment

The deployment of the PQC Signature Solution is materialized within 2 environments:

- Integration within the PQC for Secure Communication (described in Section 5)

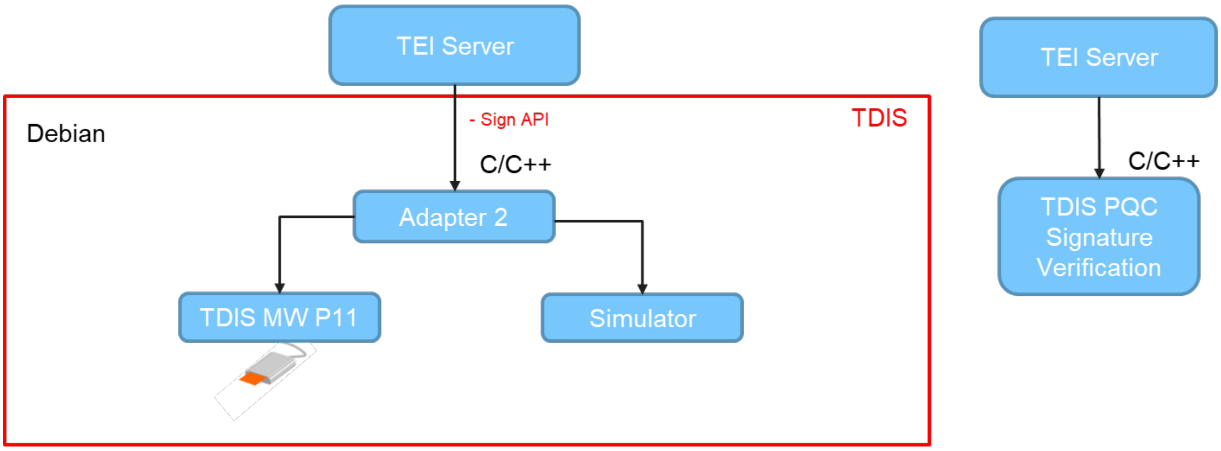


Figure 39: Integration into the PQC for Secure Communications

This Figure shows the integration of the PQC Solution into the PQC for Secure Communications environment. The PQC Signature Solution is shown in red colour. The Operating system is Linux Debian.

- Integration within the Blockchain Wallet

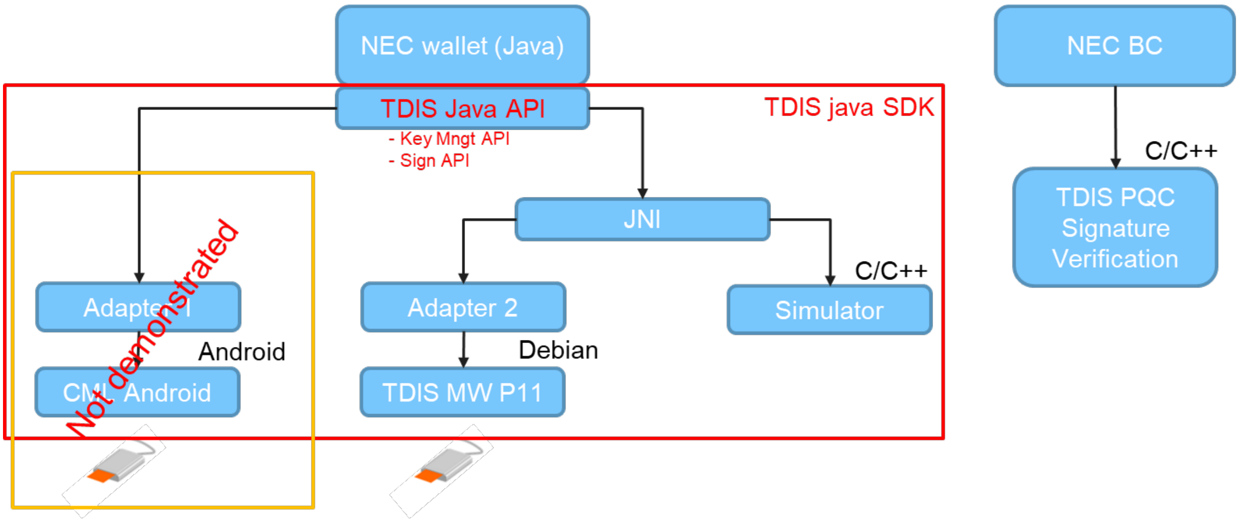


Figure 40: Integration within Blockchain

Figure 40 shows the integration of the PQC Solution into the Blockchain Wallet environment. The PQC Signature Solution is shown in red colour, while the operating system is Linux Debian.

4.6. Pending Actions and Planning

4.6.1. Results

The work carried out as part of the NANCY project allowed TDIS to refine knowledge and optimize existing implementations from the ELECTRON project. This made it possible to obtain results on the following points:

- Optimization of the reference implementation Crystals-Dilithium
    - Security Level 2: RAM memory from 27,6 KB down to 24,7 KB (NVM stable)
    - Security Level 3: RAM memory from 33,9 KB down to 29,2 KB (NVM stable)
- Compliance of the implementation with the FIPS 204 standard currently being published
- Secured implementation of Crystals Dilithium, with the objective of getting a Common Criteria certification at level EAL6+

4.6.2. Deviations

From the beginning of this project, a major risk has been identified and tracked closely. NIST PQC standardisation process still ongoing until the end of 2024 might affect the current implementation of the PQC algorithm.

Status of FIPS 204 standardisation:

- A draft of FIPS 204 was posted on August 24, 2023 on the NIST website[15]
- In the 90 day comment period, 37 commenters gave feedback (80 pages) and lots of PQC-forum discussions
- Final FIPS 204 was published on August 13, 2024

Latest changes adopted from the first draft[16]:

Besides some “EDITORIAL CHANGES”, the following “SUBSTANTIVE CHANGES” are made:

- Change SampleInBall to take all of $c\tilde{c}$, rather than just the first 256 bits
- Change ExpandMask to use SHAKE output from the beginning rather than at an offset
- Fix missing check in HintBitUnpack
- Domain Separated Pure and Pre-hash variants

4.6.3. Next steps

Among the adopted changes, some bring few security enhancements, while others are considered “cleaner”. TDIS considers that adding those required changes would not affect the final result described above. Hence, an update of the PQC Signature Solution will be provided by TDIS, but further to this NANCY project.

[15] FIPS 204, Module-Lattice-Based Digital Signature Standard | CSRC (nist.gov)

[16] FIPS 204 Update (nist.gov)

5. PQC for Secure Communications

5.1. Introduction of PQC for Secure Communications

PQC is critical to ensuring the long-term security of digital communications in the face of future quantum computing capabilities. By replacing used algorithms with quantum-resistant alternatives, PQC provides a safeguard against the risk that quantum computers will pose to modern cryptographic systems.

Quantum computers present a significant threat to current public-key cryptographic algorithms, such as RSA and Elliptic Curve Cryptography (ECC). These algorithms rely on mathematical problems, like integer factorization and discrete logarithms, which quantum computers can solve efficiently using Shor's algorithm. This ability would allow quantum adversaries to break these cryptographic schemes, compromising the security of data encrypted with them.

In contrast, symmetric encryption and hash functions are less vulnerable but not entirely immune to quantum threats. Quantum computers could use Grover's algorithm to achieve a quadratic speedup in brute-force attacks, effectively reducing the security level of symmetric encryption keys and hash functions by half. To counter this, key sizes and hash lengths may need to be doubled to retain their original security strength.

To secure communications, we must protect three critical areas:

- Key Exchange: Establishing a shared secret between communicating parties securely over an insecure channel.
- Digital Signatures: Verifying the identity of the parties involved and ensuring the integrity of the data.
- Encryption: Encrypting data to guarantee confidentiality during transmission.

Post-quantum key exchange methods are designed to resist quantum attacks, providing secure mechanisms for establishing shared secrets. Examples of such algorithms include:

- Lattice-based Cryptography: Algorithms like Kyber and NewHope leverage problems such as Learning With Errors (LWE), which remain hard for both classical and quantum computers to solve.
- Code-based Cryptography: Algorithms based on error-correcting codes, such as the McEliece cryptosystem, are resistant to quantum attacks due to their complex underlying problems.
- Multivariate Polynomial-based Cryptography: This approach involves solving systems of multivariate quadratic equations, which are computationally difficult for quantum computers.

These algorithms enable secure key exchange, even when facing an adversary with access to quantum computing capabilities.

Digital signature schemes also need to be resilient to quantum attacks to ensure data authenticity and integrity. Post-quantum digital signature algorithms include:

- Hash-based Signatures: Algorithms like SPHINCS+ use secure hash functions to create quantum-resistant signatures.
- Lattice-based Signatures: Schemes such as Dilithium and Falcon are based on lattice problems that remain difficult for quantum computers.

- Multivariate Polynomial Signatures: These use complex systems of polynomial equations, which quantum algorithms struggle to solve.

These post-quantum signature methods can replace traditional approaches like RSA and ECC-based signatures, providing quantum-resistant authentication.

Since symmetric encryption can be strengthened by increasing key sizes (e.g., using AES-256 instead of AES-128), dedicated post-quantum cryptographic algorithms are not strictly necessary for symmetric encryption and hash functions. Nonetheless, ensuring longer key lengths and larger hash outputs can help maintain strong security against potential quantum threats.

During the gradual shift to fully adopting post-quantum cryptography, hybrid cryptographic techniques can be employed. These approaches combine classical algorithms with post-quantum algorithms, offering both traditional and quantum-resistant security simultaneously.

Secure communication protocols like Transport Layer Security (TLS) can be updated to support post-quantum algorithms for key exchange and signatures, providing a layered approach to quantum-safe communication. In the following, we will see how to integrate PQC in TLS to secure MQTT communication.

5.2. Architectural Design & Implementation

This section presents the design and implementation of integrating PQC algorithms for key exchange and digital signatures within the TLS protocol, with the ultimate goal of demonstrating their application in an MQTT protocol communication scenario. The architecture is depicted in the following figure.

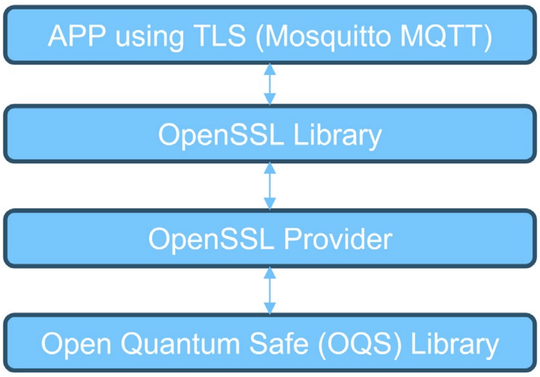

APP using TLS (Mosquitto MQTT)
OpenSSL Library
OpenSSL Provider
Open Quantum Safe (OQS) Library

Figure 41: PQC for secure communications integration architecture

The architecture consists of four main components:

- Application (APP) using TLS (Mosquitto MQTT): This component represents the MQTT-based application that communicates securely over a network. It uses the Mosquitto MQTT broker [44] for messaging and establishes secure communication channels using the TLS protocol. The TLS connection ensures that the data exchanged between MQTT clients and the broker is encrypted and authenticated.
- OpenSSL Library: OpenSSL [45] is used as the underlying cryptographic library to provide the TLS functionality. It handles the encryption, decryption, key exchange, and digital signature

operations required for secure communication. The library is configured to support both classical and post-quantum cryptographic algorithms.

- OpenSSL Provider: The OpenSSL provider is a modular component that allows for the integration of additional cryptographic algorithms into OpenSSL. In this architecture, it is used to integrate post-quantum cryptographic algorithms, making them available for use during the TLS handshake process for key exchange and digital signatures.
- Open Quantum Safe (OQS) Library: The OQS library [46] provides implementations of post-quantum cryptographic algorithms. It supplies the PQC algorithms that are integrated into the OpenSSL provider. These algorithms replace or complement traditional key exchange and signature schemes during the TLS handshake, ensuring quantum-resistant security.

The application sets up an MQTT connection using the Mosquitto MQTT broker. It requests a secure connection via TLS, which triggers the TLS handshake process. During the TLS handshake, the OpenSSL library is used to setup the secure channel. The handshake incorporates PQC algorithms provided by the OQS library through the OpenSSL provider. The key exchange algorithm can be selected from the post-quantum options available, establishing a quantum-resistant shared secret. Digital signatures used for authentication during the handshake also leverage post-quantum algorithms, ensuring the integrity and authenticity of the communication. Once the TLS handshake is successfully completed, a secure communication channel is established between the MQTT client and the broker. The data exchanged is encrypted, using the quantum-resistant session keys negotiated during the handshake. The application can then publish and subscribe to MQTT topics over this secure channel, ensuring the confidentiality and authenticity of the messages.

To enable flexible and modular integration of post-quantum cryptography within an MQTT communication scenario secured by TLS, Docker [47] is utilized to implement the architecture outlined in the previous section. Docker's containerization technology allows for the packaging of all necessary components (such as the MQTT broker, client, and cryptographic libraries) into isolated, lightweight containers. This approach ensures consistency across different deployment environments and simplifies the management of dependencies while allowing separate deployment of the broker and client for versatile testing and production scenarios.

The implementation draws inspiration from the work done within the Open Quantum Safe (OQS) project, particularly the provided demos [48]. Notably, our approach integrates OpenSSL 3 libraries since OpenSSL 1.1.1 (used in the original MQTT demo at the time of writing this deliverable) has been discontinued.

Creating a container image requires a Dockerfile that builds and integrates all necessary components. The Dockerfile performs the essential operations needed to set up the environment, as outlined in the following high-level steps.

- Base Image Selection: Begin with a base image of a Linux distribution, such as Alpine, to ensure compatibility with software packages and minimize image size.
- Install System Dependencies: Update package lists and install essential build tools and dependencies, including build-base, cmake, git, make, and ninja, which are necessary for compiling the software.
- Clone Required Repositories: Clone the repositories for the required libraries and tools using Git. This step includes fetching the Open Quantum Safe libraries, OpenSSL, the OQS provider, and Mosquitto MQTT broker.

  ```
  WORKDIR /opt
  ```

```
RUN git clone --depth 1 --branch ${LIBOQS_TAG} https://github.com/open-
quantum-safe/liboqs && \
git clone --depth 1 --branch master
https://github.com/openssl/openssl.git && \
git clone --depth 1 --branch ${OQSPROVIDER_TAG}
https://github.com/open-quantum-safe/oqs-provider.git && \
git clone --branch master https://github.com/eclipse/mosquitto.git
mosquitto
```

- Compile and Install Libraries and Tools: Build and install the required libraries, such as the Open Quantum Safe library, by configuring the build system and compiling the code.

  ```
  WORKDIR /opt/liboqs
  RUN mkdir build && cd build && \
  cmake -G"Ninja" .. ${LIBOQS_BUILD_DEFINES} -
  DCMAKE_INSTALL_PREFIX=${INSTALLDIR} && \
  ninja install
  ```

- Configure Environment Variables: Set necessary environment variables for the operation of the components, such as TLS_DEFAULT_GROUPS and SIG_ALG (signing algorithm) for TLS operations.
- Expose Ports for MQTT Communication: Make the secure MQTT port (8883) available for external communication by exposing it in the Dockerfile.

  ```
  EXPOSE 8883
  ```

- Run the Application When the Container Starts: Specify the default command or entry point to execute the application automatically when the container is launched.

These high-level steps enable the creation of a Docker container image that integrates post-quantum cryptography libraries with TLS, providing a secure MQTT communication environment.

5.3. Integration with TDIS signature token and middleware

As described in section 4.2 TDIS develop in NANCY a signature token and associated middleware, that securely performs cryptographic signing operations in hardware. Such a hardware-based solution enhances security by providing a protected environment for key management and signature operations, reducing the risk of key exposure or manipulation by malware or other attack vectors.

To integrate the TDIS signing solution in the secure communication scenario, a provider is involved that simplifies the use of the overall solution. The approach is like the integration of Open Quantum Safe libraries described in section[ref]. In Figure 42, the involved components are depicted.

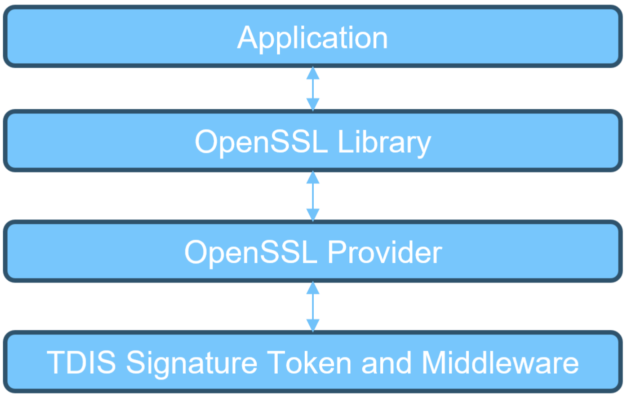
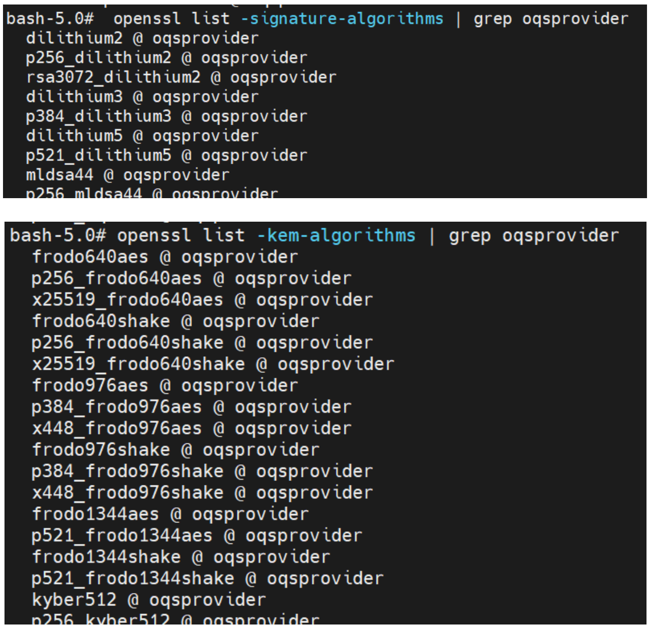


Figure 42: Signature token and middleware integration for secure communications

As described in section [4.2.1], TDIS provides a simulator of the Token and the Middleware which can be used to speed up the integration. The physical integration of the Hardware Token will be performed in Task 6.8 "Italian Massive IoT Testbed" and will be described in deliverables D6.9 “Outdoor Demonstration Planning, Evaluation Methodology and KPIs” and D6.10 “NANCY Pilots' Documentation and Evaluations”.

5.4. Testing

To test the integration, it is possible to run a container and check using OpenSSL command for signature algorithms and kem algorithms as shown (in part) in the following figures.



```
bash-5.0#  openssl list -signature-algorithms | grep oqsprovider
  dilithium2 @ oqsprovider
  p256_dilithium2 @ oqsprovider
  rsa3072_dilithium2 @ oqsprovider
  dilithium3 @ oqsprovider
  p384_dilithium3 @ oqsprovider
  dilithium5 @ oqsprovider
  p521_dilithium5 @ oqsprovider
  mldsa44 @ oqsprovider
  p256_mldsa44 @ oqsprovider
```

```
bash-5.0# openssl list -kem-algorithms | grep oqsprovider
  frodo640aes @ oqsprovider
  p256_frodo640aes @ oqsprovider
  x25519_frodo640aes @ oqsprovider
  frodo640shake @ oqsprovider
  p256_frodo640shake @ oqsprovider
  x25519_frodo640shake @ oqsprovider
  frodo976aes @ oqsprovider
  p384_frodo976aes @ oqsprovider
  x448_frodo976aes @ oqsprovider
  frodo976shake @ oqsprovider
  p384_frodo976shake @ oqsprovider
  x448_frodo976shake @ oqsprovider
  frodo1344aes @ oqsprovider
  p521_frodo1344aes @ oqsprovider
  frodo1344shake @ oqsprovider
  p521_frodo1344shake @ oqsprovider
  kyber512 @ oqsprovider
  p256_kyber512 @ oqsprovider
```

Figure 43: Signature and kem algorithms provided

To functionally test the Docker image with post-quantum cryptography-enabled MQTT communication described in section 5.2, a minimal environment was set up, to test a scenario involving three containers on a Docker network was created. These containers will consist of an MQTT broker, a publisher client, and a subscriber client.

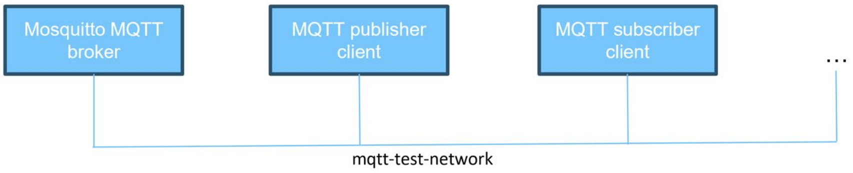
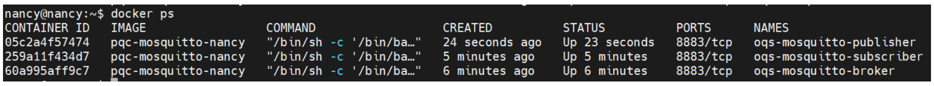


Figure 44: Minimal PQC secure communications test scenario

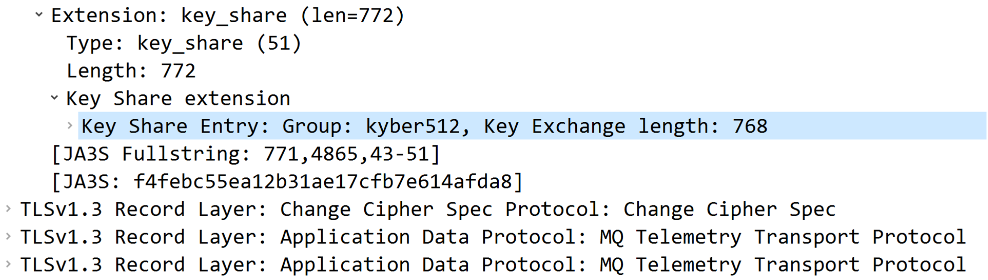
The first step is to create a dedicated Docker network (referred to as `mqtt-test-network` in the figure) to enable communication between the containers. Before starting the clients (publisher and subscriber), the Mosquitto MQTT broker container must be launched to ensure it is ready to accept incoming client connections. The clients will communicate with the broker using TLS on port 8883, utilizing post-quantum cryptography (PQC) algorithms for secure signing and key exchange. In Figure 45, the deployed containers in the test environment are shown.



```
nancy@nancy:~$ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS          PORTS      NAMES
05c2a4f57474   pqc-mosquitto-nancy   "/bin/sh -c '/bin/ba…"   24 seconds ago   Up 23 seconds   8883/tcp   oqs-mosquitto-publisher
259a11f434d7   pqc-mosquitto-nancy   "/bin/sh -c '/bin/ba…"   5 minutes ago    Up 5 minutes    8883/tcp   oqs-mosquitto-subscriber
60a995aff9c7   pqc-mosquitto-nancy   "/bin/sh -c '/bin/ba…"   6 minutes ago    Up 6 minutes    8883/tcp   oqs-mosquitto-broker
```

Figure 45: Docker containers for PQC secure communications test scenario

To verify the effectiveness of the implementation in utilizing PQC algorithms, a packet capture (pcap) trace was performed on the `mqtt-test-network` and analyzed using the Wireshark tool. The analysis of the TLS handshake confirmed the correct integration of PQC algorithms.



```
  ˅ Extension: key_share (len=772)
      Type: key_share (51)
      Length: 772
    ˅ Key Share extension
      › Key Share Entry: Group: kyber512, Key Exchange length: 768
    [JA3S Fullstring: 771,4865,43-51]
    [JA3S: f4febc55ea12b31ae17cfb7e614afda8]
› TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
› TLSv1.3 Record Layer: Application Data Protocol: MQ Telemetry Transport Protocol
› TLSv1.3 Record Layer: Application Data Protocol: MQ Telemetry Transport Protocol
```

Figure 46: PQC kyber512 key share from a captured pcap

In the previous figure, a portion of the Server Hello message from the TLS handshake is shown, with the Kyber512 PQC key share highlighted.

5.5. Pending Actions and Planning

The next steps involve extensively testing the solution in a demonstrator environment to ensure its robustness and functionality under realistic conditions. This will include validating the integration of all components, particularly the Hardware token, which is crucial for performing secure cryptographic operations. Currently, a simulator for the token has been used for development and testing purposes; however, the solution needs to be updated to incorporate the actual signature hardware token and integrate it within the system based on a Raspberry Pi (RPI) platform. This integration and testing will be carried out in the project as part of Task 6.8 "Italian Massive IoT Testbed", a commercial 5G testbed where the demonstrator environment will facilitate real-world testing of the solution's capabilities and allow for fine-tuning the overall system performance, ensuring that the final deployment meets the intended requirements.

6. Conclusion

This project has made significant strides in advancing the state-of-the-art in quantum and post-quantum cryptographic solutions, addressing key objectives through three distinct but complementary approaches: QKD experimentation, the PQC signature solution, and PQC for secure communications.

The **QKD experiment** successfully demonstrated the integration of quantum key distribution into a 5G B-RAN architecture. By experimenting with two of the most common QKD protocols and leveraging the Coherent-One-Way simulator, we provided valuable insights into the performance and viability of QKD technology for secure communication. The experimentation contributes to a deeper understanding of the practical deployment of QKD in real-world telecom environments, setting a foundation for future large-scale implementations.

In the **PQC signature solution**, significant advancements were made in optimizing the implementation of Crystals-Dilithium, reducing memory usage while maintaining compliance with evolving standards such as FIPS 204. The work carried out under this project refined previous developments, producing a highly secure solution on track for Common Criteria certification. Despite the challenges posed by the ongoing NIST PQC standardization process, the adaptability of the solution ensures readiness for integration into critical applications, from secure identity management to corporate security infrastructures.

Finally, the **PQC for secure communications** task laid the groundwork for a robust cryptographic solution that will be tested and fine-tuned in a commercial 5G testbed environment. The integration of hardware tokens with secure cryptographic operations on an RPI platform represents a key achievement in this area, setting the stage for real-world deployment in the Italian Massive IoT Testbed. This phase will validate the system’s performance, resilience, and scalability, ensuring it meets the stringent security requirements needed for future communications.

Together, **these three approaches**—QKD, PQC signatures, and PQC for secure communication—represent a holistic framework for safeguarding communication systems against current and future threats, including those posed by quantum computing. The project’s results not only push the boundaries of what is achievable today but also pave the way for secure, quantum-resistant communications infrastructure in the years to come.

References

[1] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," Systems & Signal Processing, 1984.

[2] A. K. Ekert, "Quantum cryptography based on Bell's theorem," Physical Review Letters, vol. 67, no. 6, pp. 661-663, 1991.

[3] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, F.-Z. Li, X.-W. Chen, L.-H. Sun, J.-J. Jia, J.-C. Wu, X.-J. Jiang, J.-F. Wang, Y.-M. Huang, Q. Wang, Y.-L. Zhou, L. Deng, T. Xi, L. Ma, T. Hu, Q. Zhang, Y.-A. Chen, N.-L. Liu, X.-B. Wang, Z.-C. Zhu, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang and J.-W. Pan, "Satellite-to-ground quantum key distribution," Nature, vol. 549, no. 7670, pp. 43-47, 2017.

[4] A. Aguado, V. Lopez, J. Martinez-Mateo, T. Szyrkowiec, A. Autenrieth, M. Peev, D. Lopez and V. Martin, "Hybrid Conventional and Quantum Security for Software Defined and Virtualized Networks," Journal of Optical Communications and Networking, vol. 9, no. 10, p. 819, 2017.

[5] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dusek, N. Lutkenhaus and M. Peev, "The Security of Practical Quantum Key Distribution," Reviews of Modern Physics, vol. 81, no. 3, pp. 1301-1350, 2008.

[6] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," Physical Review Letters, vol. 68, no. 21, p. 3121, 1992.

[7] F. Grosshans and P. Grangier, "Continuous Variable Quantum Cryptography Using Coherent States," Physical Review Letters, vol. 88, p. 057902, 2002.

[8] Y. Mu, J. Seberry and Y. Zheng, "Shared cryptographic bits via quantized quadrature phase amplitudes of light," Optics Communications, vol. 123, no. 1-3, pp. 344-352, 1996.

[9] D. Stucki, S. Fasel, N. Gisin, Y. Thoma and H. Zbinden, "Coherent one-way quantum key distribution," in Proceedings Volume 6583, Photon Counting Applications, Quantum Optics, and Quantum Cryptography, Prague, 2007.

[10] H. Takesue, T. Honjo, K. Tamaki and Y. Tokura, "Differential phase shift quantum key distribution," in International Telecommunication Union - Proceedings of the 1st ITU-T Kaleidoscope Academic Conference, Innovations in NGN, K-INGN, Geneva, 2008.

[11] M. Mehic, M. Niemiec, S. Rass, J. Ma, M. Peev, A. Aguado, V. Martin, S. Schauer, A. Poppe, C. Pacher and M. Voznak, "Quantum Key Distribution," ACM Computing Surveys, vol. 53, no. 5, pp. 1-41, 2021.

[12] M. Mehic, L. Michalek, E. Dervisevic, P. Burdiak, M. Plakalovic, J. Rozhon, N. Mahovac, F. Richter, E. Kaljic, F. Lauterbach, P. Njemcevic, A. Maric, M. Hamza, P. Fazio and M. Voznak, "Quantum Cryptography in 5G Networks: A Comprehensive Overview," IEEE Communications Surveys and Tutorials, 2023.

[13] A. Ntanos, D. Zavitsanos, G. Giannoulis and H. Avramopoulos, “QKD in Support of Secured P2P and P2MP Key Exchange for Low-Latency 5G Connectivity,” IEEE 3rd 5G World Forum (5GWF), 2020, pp. 157-162.

[14] A. Bahrami, A. Lord and T. Spiller, “Quantum key distribution integration with optical dense wavelength division multiplexing: a review,” IET Quantum Communication, vol. 1, no. 1, pp. 9-15, 2020.

[15] A. Aji, K. Jain and P. Krishnan, “A Survey of Quantum Key Distribution (QKD) Network Simulation Platforms,” 2nd Global Conference for Advancement in Technology (GCAT), 2021, pp. 1-8.

[16] E. Dervisevic and M. Voznak, “Large-scale quantum key distribution network simulator,” Journal of Optical Communications and Networking, vol. 16, no. 4, pp. 449-462, 2024.

[17] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124-134.

[18] W. K. Wootters and W. Zurek, “A single photon cannot be cloned,” Nature, vol. 299, p. 802–803, 1982.

[19] W. T. Buttler, R. J. Hunhes, P. G. Kwiat, S. K. Lamoreaux, G. G. Luther, G. L. Morgan, J. E. Nordhoit, C. G. Peterson and C. M. Simmons, “Practical free-space quantum key distribution over 1 km,” Phys. Rev. Lett. , vol. 81, no. 15, p. 32833286, 1998.

[20] D. Stucki, N. Brunner, N. Gisin, V. Scarani and H. Zbinden, “Fast and simple one-way quantum key distribution,” Applied Physics Letters, vol. 87, no. 19, Nov. 2005.

[21] H. Weier, “European quantum key distribution network,” Tech. Univ. Munich, 2011.

[22] D. Stucki, C. Barreiro, S. Fasel, J. Gautier and O. Gay, “High speed coherent one-way quantum key distribution prototype,” Optics Express, vol. 17, no. 16, p. 13326, Jul. 2009.

[23] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” IEEE International Conference on Computers, Systems, and Signal Processing, 1984.

[24] C. H. Bennet, F. Bessette, G. Brassard, L. Salvail and J. Smolin, “Experimental quantum cryptography,” Journal of Cryptology , vol. 5, pp. 3-38, Jan. 1992.

[25] European Telecommunications Standards Institute, “Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_QKD014v010101p.pdf

[26] V. Q. Ulitzsch, S. Marzougui, M. Tibouchi, and J.-P. Seifert,“Profiling Side-Channel Attacks on Dilithium: A Small Bit-Fiddling Leak Breaks It All,” 29th International Conference Selected Areas in Cryptography (SAC), 2022, pp. 3-33.

[27] M.-J. O. Saarinen, “WrapQ: Side-Channel Secure Key Management for Post-quantum Cryptography,” Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 637–657, 2023.

[28] H. Steffen, G. Land, L. Kogelheide, and T. Güneysu, “Breaking and Protecting the Crystal: Side-Channel Analysis of Dilithium in Hardware,” Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 688–711, 2023.

[29] M. Azouaoui, O. Bronchain, G. Cassiers, C. Hoffmann, Y. Kuzovkova, J. Renes, T. Schneider, M. Schonauer, F.-X. Standaert, and C. van Vredendaal, “Protecting Dilithium against Leakage,” IACR Transactions on Cryptographic Hardware and Embedded Systems. Universitatsbibliothek der Ruhr-Universitat Bochum, pp. 58–79, Aug. 2023.

[30] P. Ravi, B. Yang, S. Bhasin, F. Zhang, and A. Chattopadhyay, “Fiddling the Twiddle Constants - Fault Injection Analysis of the Number Theoretic Transform,” IACR Transactions on Cryptographic Hardware and Embedded Systems. Universitatsbibliothek der Ruhr-Universitat Bochum, pp. 447–481, Mar. 2023.

[31] P. Ravi, A. Chattopadhyay, J. P. D’Anvers, and A. Baksi, “Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results,” ACM Transactions on Embedded Computing Systems, vol. 23, no. 2, pp. 1–54, Mar. 2024.

[32] E. Karabulut, E. Alkim and A. Aysu, "Single-Trace Side-Channel Attacks on ω-Small Polynomial Sampling: With Applications to NTRU, NTRU Prime, and CRYSTALS-DILITHIUM," IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2021, pp. 35-45.

[33] J. W. Bos, J. Renes, and A. Sprenkels, “Dilithium for Memory Constrained Devices,” Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 217–235, 2022.

[34] I.-J. Kim, T. Lee, J. Han, B.-Y. Sim, and D.-G. Han, “Novel single-trace ML profiling attacks on NIST 3 round candidate dilithium.” IACR Cryptology. ePrint Arch., vol. 2020, p. 1383, 2020.

[35] P. Ravi, R. Poussier, S. Bhasin, and A. Chattopadhyay, “On Configurable SCA Countermeasures Against Single Trace Attacks for the NTT,” Lecture Notes in Computer Science. Springer International Publishing, pp. 123–146, 2020.

[36] P. Ravi, A. Chattopadhyay, J. P. D’Anvers, and A. Baksi, “Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results,” ACM Transactions on Embedded Computing Systems, vol. 23, no. 2, pp. 1–54, Mar. 2024.

[37] K. A. Jackson, C. A. Miller, and D. Wang, “Evaluating the Security of CRYSTALS-Dilithium in the Quantum Random Oracle Model,” Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 418–446, 2024.

[38] H. Wang, Y. Gao, Y. Liu, Q. Zhang, and Y. Zhou, “In-depth Correlation Power Analysis Attacks on a Hardware Implementation of CRYSTALS-Dilithium,” Cybersecurity, vol. 7, no. 1, Jun. 2024.

[39] Z Qiao, Y Liu, Y Zhou, M Shao, S Sun, “When NTT Meets SIS: Efficient Side-channel Attacks on Dilithium and Kyber,” Cryptology ePrint Archive, 2023, pp. 1-16.

[40] A. Calle Viera, A. Berzati, and K. Heydemann, “Fault Attacks Sensitivity of Public Parameters in the Dilithium Verification,” Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 62–83, 2024.

[41] O. Bronchain, M. Azouaoui, M. ElGhamrawy, J. Renes, and T. Schneider, “Exploiting Small-Norm Polynomial Multiplication with Physical Attacks,” IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2024, no. 2, pp. 359–383, Mar. 2024.

[42] M. ElGhamrawy et al., “From MLWE to RLWE: A Differential Fault Attack on Randomized & Deterministic Dilithium,” IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 262–286, Aug. 2023.

[43] J.-S. Coron, F. Gérard, M. Trannoy, and R. Zeitoun, “Improved Gadgets for the High-Order Masking of Dilithium,” IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 110-145, Aug. 2023.

[44] “Eclipse Mosquitto - An open source MQTT broker,” [Online]. Available: https://mosquitto.org/

[45] “OpenSSL Library,” [Online]. Available: https://openssl-library.org/

[46] “Open Quantum Safe,” [Online]. Available: https://openquantumsafe.org/

[47] “Docker,” [Online]. Available: https://www.docker.com/

[48] “Open Quantum Safe demos,” [Online]. Available: https://github.com/open-quantum-safe/oqs-demos

[49] CORDIS, “Cordis,” [Online]. Available: https://cordis.europa.eu/project/id/101021936

[50] “European Union Agency for Cybersecurity - ENISA” [Online]. Available: https://www.enisa.europa.eu/

[51] “Agence nationale de la sécurité des systèmes d'information,” [Online]. Available: https://cyber.gouv.fr/en

[52] “BSI - Federal Office for Information Security”, [Online]. Available: https://www.bsi.bund.de/EN/Home/home_node.html