Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 420–428

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

Hybrid QKD-based framework for secure enterprise communication system

Edgars Rencis*, Juris Vīksna, Sergejs Kozlovičs, Edgars Celms, Dāvis Jānis Lāriņš, Krišjānis Petručeņa

Institute of Mathematics and Computer Science, University of Latvia, Raina blvd. 29, Riga, LV-1459, Latvia

Abstract

This research paper proposes a hybrid quantum key distribution (QKD)-based framework for secure enterprise communication using post-quantum cryptography (PQC) and smart cards. The paper discusses the limitations of current PQC implementations and the need for a hybrid approach that combines classical non-PQC algorithms with quantum-resistant PQC algorithms to establish secure communication channels. The proposed framework utilizes point-to-point QKD links between quantum devices to establish a shared secret key, which is then used to encrypt and decrypt data using PQC algorithms. Smart cards are used for authentication and key synchronization to ensure the security of end-user devices. The paper also reviews relevant literature on QKD and PQC and discusses future work. The proposed framework offers a cost-effective and secure way for enterprises to communicate with each other.

Keywords: Quantum Key Distribution; Post-Quantum Cryptography; hybrid digital signatures; hybrid key exchange.

* Corresponding author. Tel.: +371 26462224.
E-mail address: edgars.rencis@lumii.lv

10.1016/j.procs.2024.06.189

1. Introduction

In today's business environment, secure communication between enterprises is crucial. Traditional public-key cryptographic algorithms, such as RSA, Elliptic Curve Cryptography, and Elliptic-curve Diffie–Hellman (ECDH), have been used as shared secret exchange schemes for decades. Traditional public key schemes have also been used to digitally sign messages and certificates (e.g., to authenticate the client and the server). After obtaining the trusted shared secret, it is used as a symmetric secret key to encrypt application data, e.g., using AES encryption in GCM mode. This is how TLS works.

However, with the advent of quantum computers, we have to reconsider the traditional TLS flow. While quantum computers do not significantly impact state-of-the-art symmetric ciphers (AES variants), they do break both RSA and elliptic curve-based cryptosystems [1]. As a result, there is a growing need for quantum-safe cryptographic techniques that can protect sensitive information against attacks from classical and quantum computers. There are two approaches. The first one is post-quantum cryptography (PQC), which provides quantum-resistant public key algorithms (intended for classical computers) for key exchange and digital signatures. The second one is Quantum Key Distribution (QKD), which offers a secure way of distributing cryptographic keys over a quantum communication channel usually implemented as an optical link. The laws of quantum mechanics ensure we can detect whether the keys have been eavesdropped on or altered.

Both PQC and QKD face challenges in widespread adoption. QKD's theoretical security is hindered by the cost and complexity of the required hardware [2]. PQC, undergoing standardization by NIST, lacks agreed-upon aspects and parameters [3]. While PQC's theoretical security is promising, its practical implementations may be vulnerable. Attacks on post-quantum algorithms have already been observed, including the McEliece-Goppa Syndrome Decoding (MGSD) challenge, which tests the security of the McEliece cryptosystem [4]. The McEliece cryptosystem is a public key cryptosystem based on error-correcting codes [5]. Esser et al. broke the 1284-bit version [6], and Bernstein et al. raised the record to 1347 bits [7]. The MGSD challenge facilitates security testing and motivates the development of new schemes.

Vulnerabilities have also been found in the CRYSTALS-Kyber implementation [8]. Dubrova et al. used deep learning to exploit side channels, while Guo et al. broke the same implementation differently [9-11]. Additionally, Guo and co-authors performed the first key-recovery side-channel attack on Classic McEliece [12]. However, workarounds for PQC algorithms are being proposed rapidly, with presentations at the NIST 4th PQC Standardization Conference.

Successful side-channel attacks on the TLS protocol, once considered highly secure, have been reported. Timing analysis was exploited by Nadhem et al. [13], and Merget et al. demonstrated a similar approach [14]. Fortunately, the latest TLS v1.3 appears to be secure against such attacks, with RSA no longer used for key exchange. In this paper, we propose a hybrid QKD-based framework for secure enterprise communication that employs PQC and smart cards (using ECC [†] algorithms) for key distribution. The examples above demonstrate that practical implementations of PQC are still far from being truly secure. Therefore, a hybrid approach is needed to establish solutions that provide the desired security in real-life applications. Our proposed framework uses point-to-point QKD links between quantum devices to establish a shared secret key, which is then used to encrypt and decrypt data using a PQC algorithm. We assume that QKD point-to-point communication is completely secure, and our approach aims to transform it into a shared mode of communication for multiple users without compromising its security.

The first problem we address in this paper is achieving a shared mode of communication without compromising the security of point-to-point communication. The second problem we address is the issue of key distribution. We propose using tamper-resistant data storage devices, such as smart cards, to distribute the keys securely. We combine

[†] elliptic curve cryptograpy

FIPS-compliant classical non-PQC algorithms with less-studied but quantum-resistant PQC algorithms. The proposed approach provides a highly secure and efficient method for communication between multiple users in an enterprise setting.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the QKD as a Service architecture that we use in our framework. In Section 3, we describe the proposed framework in detail, including the integration of smart cards for authentication and key synchronization. In Section 4, we review the relevant literature on QKD and PQC. Finally, in Section 5, we conclude the paper and discuss future work.

2. QKD as a Service

QKD enables the secure exchange of cryptographic keys between two parties by utilizing the laws of quantum mechanics to detect any attempts of eavesdropping. We have developed the infrastructure called QKD as a Service (QaaS), which allows end users to access our QKD link as a shared resource by connecting to it via classical links. This service is particularly useful for enterprises, such as government agencies, financial institutions, and healthcare providers, that require strong cryptographic security for their communications. By utilizing QaaS, these organizations can enjoy the security benefits of QKD without the need to invest in and maintain their own QKD infrastructure. We have implemented a prototype of the protocol and submitted a paper describing it [15].

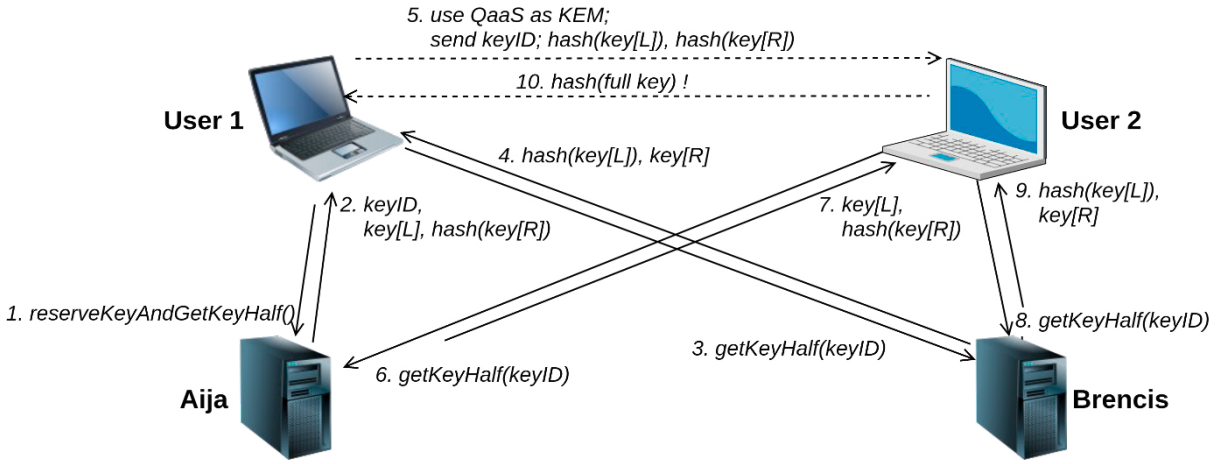

Fig. 1. The QaaS (QKD as a service) architecture and message flow.

Figure 1 depicts the architecture of QaaS. User 1 initializes a connection to User 2. During the TLS handshake, a symmetric encryption key has to be exchanged. However, instead of using a traditional key encapsulation mechanism (KEM) such as ECDH or any post-quantum KEM, a virtual QaaS KEM is used. In QaaS KEM, both users connect to the key distribution centers, KDCs, named Aija and Brencis (we use different server names to distinguish them from QKD hardware devices that are named "Alice" and "Bob" and are linked to "Aija" and "Brencis" respectively), which are directly connected to the corresponding endpoints of the QKD link.

From the QKD point of view, the classical links between end users and KDCs are the weakest part of the key distribution process. To sustain active attacks on any single link, we split the quantumly shared key into halves, sent via two different classical channels. A successful man-in-the-middle attack would now require compromising two independent TLS communication links. Aija returns the first half of the quantumly distributed key and the hash of the second half. Brencis, in its turn, returns the second half of the key and the hash of the first half. Thus, both users can validate key halves against each other. For Users 1 and 2, we provide a library implementing QaaS KEM.

From the end-user point of view, QaaS is a set of protocols for obtaining a quantumly shared key from remote KDCs. In the current implementation, KDCs are connected with a direct quantum channel. In the future, KDCs can be connected via intermediate trusted nodes, but end users can still use the same QaaS protocols. Developing a bridged QKD network is a high-ambitious and non-trivial task. Building the current single-link QKD was already a challenge since we had to generate and validate PQC certificates for each endpoint and CAs. Besides, we had to implement the Butterfly protocol and the controlling server with the ability to re-initialize QaaS in case of failure.

To provide the highest level of security, we have incorporated encryption (via session keys) and authentication schemas into the QaaS process.

Encryption with session keys. While the link between User 1 and User 2 uses QaaS KEM, other links rely on the traditional KEM approach (embedded into TLS v1.3) for obtaining symmetric session keys.

To establish a secure link between endpoints **A** (the client) and **B** (the server), KEM launches a three-stage process for generating the shared session key:

1. The client generates their ephemeral public/private key pair (`pubk/privk`) and sends `pubk` to the server.
2. The server generates the shared secret (`ss`), encrypts it with `pubk`, and sends the ciphertext (`enc(ss,pubk)`) back to the client.
3. The client decrypts the ciphertext with their private key (`privk`) to obtain `ss`.

For key exchange between Users 1/2 and servers Aija/Brencis, we utilize FrodoKEM-640, a post-quantum algorithm that has not been successfully attacked so far and has been implemented in the two most popular libraries supporting PQC cryptography – OpenQuantumSafe and BouncyCastle.

For QaaS KEM between Users 1 and 2, we use the modified three-stage KEM process, where the shared key is not transmitted at all (even encrypted) between Users 1 and 2. Instead, the key is obtained by combining key halves from Aija and Brencis. Thus, our QaaS architecture is sustainable against active attacks on any single link.

Authentication. We use traditional TLS v1.3 handshake flow and X.509 certificates for authentication. We utilize the SPHINCS+ post-quantum algorithm, which is one of the algorithms selected for standardization by NIST. Besides we have also proposed a hash-based server (User 2) authentication in our paper [15].

3. Hybrid QKD-based framework

To mitigate the risks of post-quantum algorithms becoming vulnerable in the future (after more research is done), we propose using a hybrid approach that combines traditional and post-quantum encryption algorithms for both KEMs and authentication. PQC algorithms still have to withstand the test of time – new attacks are constantly emerging, and the NIST standardization process is not yet finished. A hybrid ECC+PQC approach ensures that even if underlying PQC algorithms are broken, QaaS classical links remain secure as long as ECC is secure (e.g. until a powerful-enough quantum computer is built). Besides, storing ECC keys on chip cards ensures additional security against cloning (currently, chip cards do not support PQC keys in write-only memory).

One option is to use hybrid algorithms already implemented by the Open Quantum Safe (OQS) project, which supports developing and prototyping quantum-resistant cryptography [16]. Another option is to combine classical and PQC algorithms manually.

Hybrid KEMs. Figure 2 depicts the hybrid KEM three-stage process. Actually, KEM is performed twice, and two ephemeral shared secrets are being exchanged – one using the classical ECDH approach (we do not use RSA since it is not supported as a KEM in TLS v1.3) and the other using PQC KEM (FrodoKEM in our experiments). Two shared secrets are combined using the bitwise XOR operation. Thus, compromising any one of the secrets does not reveal the combined key.

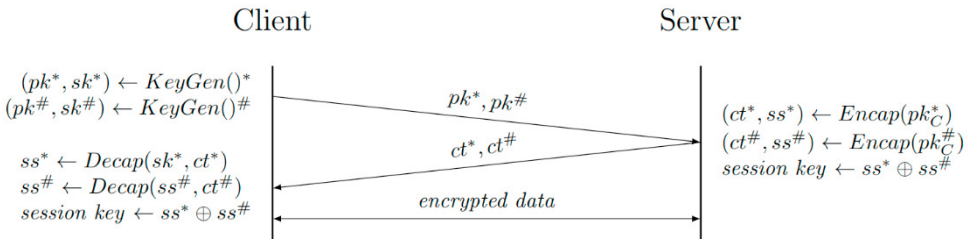

Fig. 2. Classical and PQC key pairs. Asterisk (*) denotes the classical KEM, and dash (#) denotes the PQC KEM.

Hybrid authentication. Figure 3 depicts the authentication process. For authentication, we propose using smart cards as tamper-resistant devices on which the client-side certificates and private keys would be stored. The keys would be protected by a password, thereby introducing the two-factor authentication schema and obtaining an even higher level of security. However, smart cards have some limitations. For example, we cannot use post-quantum or hybrid algorithms for signing and verifying messages due to the unavailability of smart cards on which such algorithms are implemented. It takes time to develop and study balanced side-channel attack-resistant implementations. Therefore, we propose a scheme where there are two certificates on the client side – one PQC (SPHINCS+) certificate and the other generated by a traditional algorithm (we use the P-256 elliptic curve in our experiments). The PQC (e.g., SPHINCS+) certificate would be generated by the LibOQS library (invoked as OpenSSL provider) and stored on a smart card (in a read-write memory) along with its private key. The other certificate (e.g. ECC P-256) would be generated by traditional means (e.g. by OpenSSL, BoringSSL, Java KeyPairGenerator class, etc.) and also stored on the smart card in the write-only memory (the JavaCard technology ensures that the classical ECC key cannot be extracted from the card). Each PQC and ECC public key is signed with the corresponding CA (PQC or ECC) private key. The client sends both PQC and ECC certificates (long-lived) to the server, which authenticates the client if and only if both certificates are signed with the corresponding trusted CA and the client can sign the handshake with both corresponding private keys. To validate the client that has two (PQC and ECC) keys stored in two different memories, we need to develop a new signature method (called QaaS Hybrid Signature Method, QaaS HSM) that uses LibOQS for working with PQC keys in read-write memory and JavaCard technology for working with ECC keys in write-only memory. After performing the signing process, QaaS HSM concatenates the PQC and ECC certificates. The validation process de-concatenates them and validates them separately. We utilize our original TLS injection mechanism (used in the default QaaS implementation) to inject QaaS HSM into the TLS v1.3 flow [15].

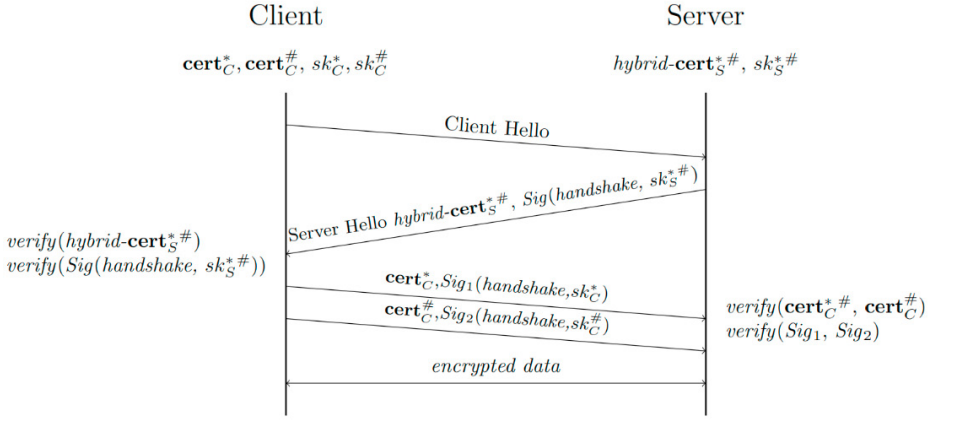

Fig. 3. Hybrid authentication process. Asterisk and dash (*#) denote a LibOQS built-in hybrid algorithm such as p256-SPHINCS+.

On the server side, we do not need to use smart cards, since we assume that the server is physically secured. Smart card computations are slow, which would become a bottleneck for the server. Thus, we store the server certificates and private keys in the server file system. Although we can still generate one ECC and one PQC server certificate (and rely on QaaS HSM), we apply a more convenient approach of signing just one hybrid certificate utilizing one of the predefined hybrid algorithms available in LibOQS, e.g., p256-SPHINCS+. The same hybrid algorithm is used for the CA and the server certificate; thus, the client can validate the server using LibOQS directly.

We have conducted experiments using the NXP J3H145 Java Card, which implements JavaCard version 3.0.4 and Global Platform version 2.2.1. The card has a total of 144 kilobytes of EEPROM memory with approximately 110 kilobytes available for user storage. It supports the AES cipher with a key length of up to 256 bits, as well as the RSA cipher and the ECC signature scheme with a maximum key length of 2048 bits. The card complies with the FIPS Publication 140-2, Level 4, and meets the Common Criteria Evaluation Assurance Level 5+. We have implemented a prototype as a JavaCard applet that allows working with cryptographic primitives.

Our experiments have demonstrated that the NXP J3H145 Java Card is capable of storing and interacting with traditional RSA certificates. Additionally, the card's available memory is sufficient, albeit barely, for storing SPHINCS+ certificates. For the SPHINCS+ *SHA256-128f-robust* variant, a PEM (Base64)-encoded client certificate takes around 23 KiB, and the private key takes 213 bytes (the keys were created using OpenSSL v3 with LibOQS provider). A PKCS#12 file containing both the certificate and the private key takes around 34.6 KiB (created using the OpenSSL PKCS#12 export feature).

Our contribution is a double hybrid method that combines classical (ECC) and PQC algorithms. The method uses the physical JavaCard's built-in ECC implementation in combination with a hybrid ECC+PQC KEM and authentication via X.509 certificates according to the X.501 standard. This approach enables secure and efficient cryptographic operations that leverage the strengths of both ECC and PQC algorithms while also benefiting from the trusted environment provided by the JavaCard platform.

4. Related work

QKD has been an active research area in the field of quantum cryptography for several decades. The idea of using quantum mechanics to distribute cryptographic keys securely was first proposed by Bennett and Brassard in 1984 [17]. Since then, numerous researchers have explored various aspects of QKD, ranging from fundamental theoretical questions to practical implementation issues.

Several experimental QKD systems have been developed and demonstrated over the past few decades. These systems typically involve the use of single photons as quantum information carriers and require specialized hardware such as photon detectors and sources, fiber optic cables, and other ancillary equipment. A number of commercial QKD systems are also available today, such as those offered by ID Quantique [18] and Toshiba [19].

However, despite the significant progress made in the field, several challenges remain to the practical implementation of QKD. One major issue is the limited working distance of QKD systems, which is currently limited to a few hundred kilometers over optical fiber [20]. Another major challenge is the issue of practical security, including the susceptibility of QKD systems to side-channel attacks [21]. Several proposals have been made to address these challenges, including using entanglement-based protocols and incorporating error correction codes into the QKD systems [22-23].

Recent research has also focused on using post-quantum cryptography in QKD systems. As the field of quantum computing continues to advance, it is becoming increasingly important to develop quantum-resistant cryptographic algorithms that can withstand attacks by quantum computers. Several PQC algorithms have been proposed for use in QKD systems, such as the post-quantum public key encryption schemes like CRYSTALS-Kyber [8] and the McEliece cryptosystem [5] as well as the post-quantum key exchange protocols like Frodo [24] and BIKE [25].

There have also been attempts to explore the benefits of Software-Defined Networking (SDN) in the context of QKD networks. Aguado et al. propose an SDN-based architecture for QKD networks that simplifies the design,

deployment, and management of QKD networks [26]. Zhang et al. present a prototype QKD network based on SDN and demonstrate the feasibility of using SDN to manage QKD resources in real-time, such as controlling the distribution of quantum keys across different paths in the network [27].

Overall, while significant progress has been made in the field of QKD, several challenges remain that must be addressed to implement practical, scalable QKD systems that can be deployed in real-world environments and that would benefit enterprises. A valuable insight into the performance and limitations of QKD protocols is presented in [28], where the author provides a comprehensive review of QKD protocols for secure communication. One possible way of addressing these issues is to use the hybrid approach to the key exchange process, where two or more algorithms of different classes are used sequentially to obtain a level of security that is not lower than that of the strongest of these algorithms. Such an approach would mitigate the risk of quantum attacks and preserve common security guarantees, e.g., FIPS compliance [29]. Harnik et al. have coined the concept of "robust combiner," a method for combining multiple candidate schemas to improve security, resilience, and efficiency by taking into account the security assumptions and guarantees of each protocol [30]. Bindel et al. have also introduced such combined algorithms for hybrid digital signatures [31]. Another hybrid example is introduced by Bos et al., who construct cipher suites for the TLS protocol by tying together lattice-based key exchange with traditional authentication using RSA or elliptic curve digital signatures [32].

5. Conclusions

The world is currently at the forefront of a quantum computing breakthrough. While we are not yet there, we must begin to consider making our encryption quantum-safe. In this context, every advancement towards a common goal of designing and improving quantum key distribution (QKD) networks is essential. Several attempts have been made, such as those proposed in [26] and [27]. However, while [26] proposes an SDN-based architecture for QKD networks, it does not provide a detailed implementation or evaluation of the proposed solution. On the other hand, [27] presents a prototype QKD network based on SDN, but the evaluation is limited to a small-scale scenario. Future work could explore the scalability and performance of the proposed solution in larger and more complex networks.

In this paper, we have proposed a hybrid approach toward achieving strong cryptographic security that is also practical for enterprises. Our framework exploits the benefits of existing post-quantum algorithms while maintaining a level of security that is no less than that of traditional algorithms that have been around for several decades. We believe that our approach is a significant step forward in the quest for quantum-safe encryption. Further research can explore the potential of our framework in real-world scenarios, including its scalability and performance, to validate its effectiveness and practicality. Therefore, our contribution to the field invites researchers to continue this effort toward a secure and quantum-safe future.

Acknowledgments

Research is supported by the European Regional Development Fund, project No. 1.1.1.1/20/A/106 "Applications of quantum cryptography devices and software solutions in computational infrastructure framework in Latvia" and by the Latvian Quantum Initiative under European Union Recovery and Resilience Facility project No. 2.3.1.1.i.0/1/22/I/CFLA/001.

References

[1] Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K. (2017) "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms", in: Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science, vol 10625. Springer, Cham. https://doi.org/10.1007/978-3-319-70697-9_9.

[2] Jacak, M., Jacak, J., Jóźwiak, P., Jóźwiak, I. (2016) "Quantum cryptography: Theoretical protocols for quantum key distribution and tests of selected commercial QKD sys-tems in commercial fiber networks", in International Journal of Quantum Information 14 (02) 1630002.

[3] Chowdhury, S., Covic, A., Acharya, R.Y. et al. (2022) "Physical security in the post-quantum era." J Cryptogr Eng 12, 267–303. https://doi.org/10.1007/s13389-021-00255-w.

[4] Aragon, N., Lavauzelle, J., Lesquesne, M. (2019). Decoding Challenge. Available online: http://decodingchallenge.org.

[5] Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R. et al. (2022) "Classic McEliece: conservative code-based cryptography." https://classic.mceliece.org/nist/mceliece-20201010.pdf, October 2020.

[6] Esser, A., May, A., Zweydinger, F. (2022) "McEliece Needs a Break – Solving McEliece-1284 and Quasi-Cyclic-2918 with Modern ISD", in Dunkelman, O., Dziembowski, S. (eds) Advances in Cryptology – EUROCRYPT 2022. EUROCRYPT 2022. Lecture Notes in Computer Science, vol 13277. Springer, Cham. https://doi.org/10.1007/978-3-031-07082-2_16.

[7] Bernstein, D.J., Lange, T., Peters, C. (2008) "Attacking and Defending the McEliece Cryptosystem", in Buchmann, J., Ding, J. (eds) Post-Quantum Cryptography. PQCrypto 2008. Lecture Notes in Computer Science, vol 5299. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88403-3_3.

[8] Bos, J. et al., (2018) "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM." IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 2018, pp. 353-367. https://doi.org/10.1109/EuroSP.2018.00032.

[9] Dubrova, E., Ngo, K., & Gärtner, J. (2022) "Breaking a Fifth-Order Masked Implementation of CRYSTALS-Kyber by Copy-Paste." Cryptology ePrint Archive 2022/1713.

[10] Wang, H., & Dubrova, E. (2021) "Tandem Deep Learning Side-Channel Attack on FPGA Implementation of AES." SN Computer Science, 2, 1–12. https://doi.org/10.1007/s42979-021-00755-w.

[11] Guo, Q., Nabokov, D., Nilsson., A., & Johansson., T. (2023) "SCA-LDPC: A Code-Based Framework for Key-Recovery Side-Channel Attacks on Post-Quantum Encryption Schemes." Cryptology ePrint Archive.

[12] Guo, Q., Johansson, A., & Johansson, T. (2022) "A Key-Recovery Side-Channel Attack on Classic McEliece Implementations." IACR Transactions on Cryptographic Hardware and Embedded Systems 2022 (4): 800–827. https://doi.org/10.46586/tches.v2022.i4.800-827.

[13] Fardan, N.J. & Paterson, K.G.. (2013) "Lucky thirteen: Breaking the TLS and DTLS record protocols." 526-540. https://doi.org/10.1109/SP.2013.42.

[14] Merget, R., Brinkmann, M., Aviram, N., Somorovsky, J., Mittmann, J., & Schwenk, J. (2020) "Raccoon Attack: Finding and Exploiting Most-Significant-Bit-Oracles in TLS-DH(E)." Cryptology ePrint Archive, Paper 2020/1151.

[15] Kozlovičs, S., Petručeņa, K., Lāriņš, D., Vīksna, J. (2023) "QKD as a Service and Its Injection into TLS." Proceedings of the The 18th International Conference on Information Security Practice and Experience (ISPEC 2023), August 24-25, 2023. Under review.

[16] Open Quantum Safe (2023), https://openquantumsafe.org.

[17] Bennett, C.H., and Brassard, G. (1984) "Quantum Cryptography: Public Key Distribution and Coin Tossing." Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, 10-12 December 1984, 175-179. https://doi.org/10.48550/arXiv.2003.06557.

[18] IDQ: Redefining Security: Clavis XG QKD System (2022) https://www.idquantique.com/quantum-safe-security/products/clavis-xg-qkd-system.

[19] Toshiba Quantum Key Distribution (2023), https://www.global.toshiba/ww/products-solutions/security-ict/qkd.html.

[20] Huttner, B., Alléaume, R., Diamanti, E. et al. (2022) "Long-range QKD without trusted nodes is not possible with current technology." npj Quantum Inf 8, 108. https://doi.org/10.1038/s41534-022-00613-4.

[21] Arteaga-Díaz, P., Cano, D., and Fernandez, V. (2022) "Practical Side-Channel Attack on Free-Space QKD Systems With Misaligned Sources and Countermeasures" in IEEE Access, vol. 10, pp. 82697-82705 https://doi.org/10.1109/ACCESS.2022.3196677.

[22] Neagu, M.; Miclea, L.; Manich, S. (2015) "On the use of error detecting and correcting codes to boost security in caches against side channel attacks." A: Workshop on Trustworthy Manufacturing and Utilization of Secure Devices." Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition: 9-13 March 2015, Grenoble, France". Grenoble: 2015, p. 1-6.

[23] Jia, W., Feng, B., Yu, H., and Bian, Y. (2019) "Quantum key distribution protocol based on CSS error correcting codes", in Proceedings of the ACM Turing Celebration Conference - China (ACM TURC '19). Association for Computing Machinery, New York, NY, USA, Article 152, 1–8. https://doi.org/10.1145/3321408.3326680.

[24] Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., and Stebila, D. (2016) "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE." 1006-1018. https://doi.org/10.1145/2976749.2978425.

[25] "BIKE - Bit Flipping Key Encapsulation" (2022) https://bikesuite.org.

[26] Aguado, A. et al. (2019) "The Engineering of Software-Defined Quantum Key Distribution Networks", in IEEE Communications Magazine, vol. 57, no. 7, pp. 20-26, July 2019, https://doi.org/10.48550/arXiv.1907.00174.

[27] Zhang, H., Quan, D., Zhu, C., Li, Z. (2018) "A Quantum Cryptography Communication Network Based on Software Defined Network", in ITM Web Conf https://doi.org/10.1051/itmconf/20181701008.

[28] Kong, P.Y. (2022) "A Review of Quantum Key Distribution Protocols in the Perspective of Smart Grid Communication Security", in IEEE Systems Journal, vol. 16, no. 1, pp. 41-54, March 2022 https://doi.org/10.1109/JSYST.2020.3024956.

[29] Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., Stebila, D. (2019) "Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange", in Proceedings of the 10th International Conference on Post-Quantum Cryptography PQCrypto 2019. Lecture Notes in Computer Science, vol. 11505, 206-226. https://doi.org/10.1007/978-3-030-25510-7_12.

[30] Harnik, D., Kilian, J., Naor, M., Reingold, O., Rosen, A. (2005) "On robust combiners for oblivious transfer and other primitives", in Proceedings of EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, 96-113. https://doi.org/10.1007/11426639_6.

[31] Bindel, N., Herath, U., McKague, M., Stebila, D. (2017) "Transitioning to a Quantum-Resistant Public Key Infrastructure", in Proceedings of the 8th International Conference on Post-Quantum Cryptography PQCrypto 2017. Lecture Notes in Computer Science, vol. 10346, 384-405. https://doi.org/10.1007/978-3-319-59879-6_22.

[32] Bos, J.W., Costello, C., Naehrig, M., Stebila, D. (2015) "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem.", in 2015 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2015, 553-570. https://doi.org/10.1109/SP.2015.40.