Post-Quantum Secure Protocols for Communication Between Data Processing Units.

A. Cano Aguilera,[1,2] C. Rubio Garcia,[1] ,D.C. Lawo, [1,2] R. Frantz, [1,2] I. Tafur Monroy,[1] J.L. Imaña[3] and J.J Vegas Olmos[1,2]

[1] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, Netherlands,
[2] Software Architecture, NVIDIA Corporation, Ofer Industrial Park Yokneam, Israel,
[3] Universidad Complutense de Madrid, Madrid, Spain

The imminent appearance of Quantum Computers poses a threat to the cryptographic algorithms used by public key infrastructure (PKI) of world-wide used communication protocols. High performance computing (HPC) data centers, among others entities, being well aware of the catastrophic consequences of quantum attacks to their PKI, are transitioning to Post-Quantum Cryptographic (PQC), despite the considerable associated overhead on handling the incoming network packets. This work helps with the transition to PQC providing an implementation of a quantum-resilient security protocol for communication between data processing units (DPU). Our solution ensures quantum security providing PQ authentication by means of CRYSTALS-Dilithium, PQ authenticated Key exchanges by means of CRYSTALS-Kyber and confidential transmission of data by means of AES-256. Additionally, the implementation on DPUs help reducing the computational load on the HPC data centers .

Introduction

Since the discovery of quantum algorithms like Shor's and Grover's and its associated threats to classical cryptography, cryptographic experts have made significant efforts to develop quantum resistant cryptography. PQC involves schemes designed to handle quantum computer attacks based on computational problems such as lattices or hashes. In 2015, NIST initiated the Post-Quantum Cryptography Standardization Process to select algorithms for Digital Signatures and Key Encapsulation/Exchange. Recently, NIST released three different drafts for standardizing Kyber [1], Dilithium [2], and Sphincs+.

This work proposes a Post-Quantum protocol safeguarding high-speed DPU link communications. Dilithium verifies party authentication, followed by Kyber executing an AES-256 [3] authenticated key exchange, the main novelty of this paper, between sender and receiver. These PQC-exchanged keys protect the data plane.

Classical and Post-Quantum Cryptography pose high workloads for datacenters. Optimizing cryptographic performance, current schemes often use FPGA or application-specific integrated circuits. DPUs emerge as a promising solution due to their tools for cryptographic tasks in real-time communication systems.

This paper explores DPUs creating communication links in large-scale environments. End-to-end post-quantum cryptography communication is showcased through a software stack for Dilithium and Kyber on DPU units, enhancing CPU cycle performance and throughput. The experimental setup details PQC integration into a real communication system handled by DPUs, presenting results and summarizing key novelties.

Experimental set-up

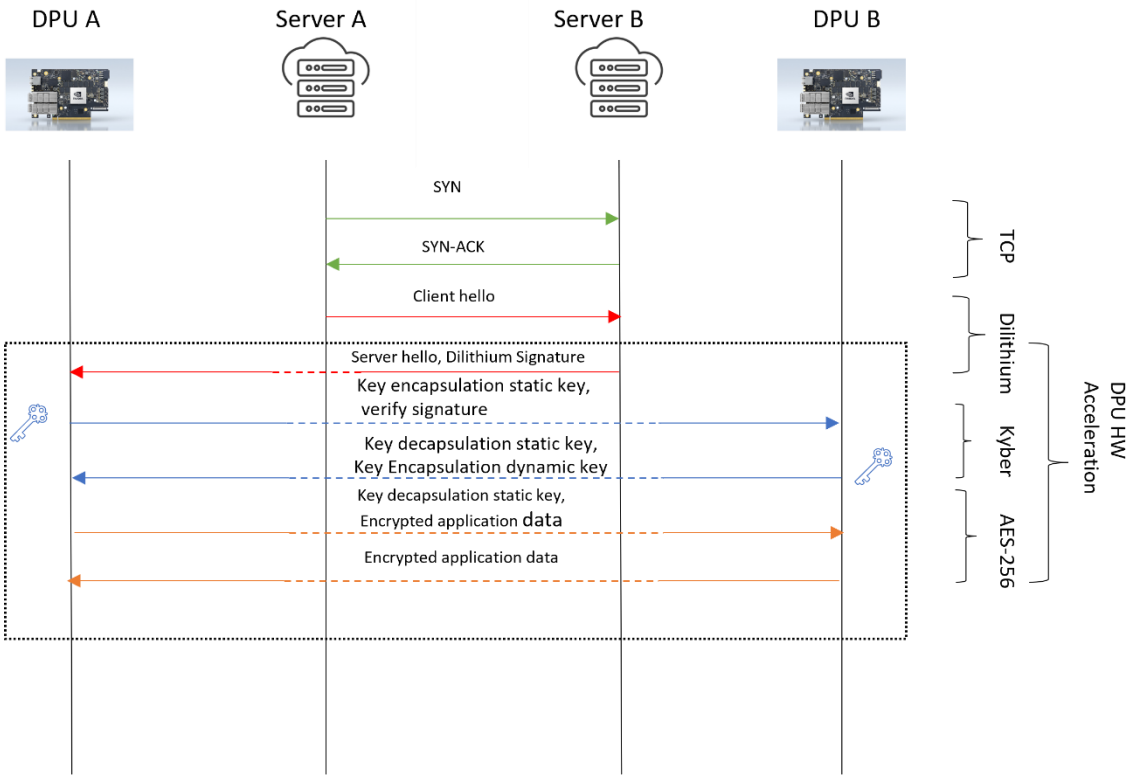

Figure 1: Experimental set-up. Transmitter initiates a quantum-resilient link using Dilithium and Kyber, authenticating the receiver with a Dilithium-generated signature. Upon identity confirmation, a key-exchange protocol ensues, establishing an AES-256 key for encrypted data transmission between receiver and transmitter.

In this section, we describe the experimental configuration required to establish secure post-quantum cryptography (PQC) communications between Data Processing Units (DPUs). The setup, illustrated in Figure 1, consists of two independent servers, each equipped with its own CPU, and two DPUs (model MBF2H516ACEEOT) capable of processing data at rate of 100 Gigabits per second (100G). To connect the servers with the DPUs, Peripheral Component Interconnect Express (PCIe) bridges are employed. The DPUs are interconnected via optical pluggable that use Direct Attached Copper (DAC) cables. These DPUs feature ARMv8 A72 cores dedicated to specific operations.

We initiate channel establishment by sending a client hello to the server. The server responds with a server hello and a Dilithium signature, which is then verified. Subsequently, the key encapsulation payload is exchanged, encapsulating two symmetric keys. One key uses a static public key on the server side for authentication, and the other employs a dynamically generated key by the Kyber Key Generation Algorithm on the client side. After the key exchange, both keys are hashed, enabling communication between the Client and Server using AES-256. These cryptographic operations can be offloaded to the DPU (see Figure 1).

Results

Kyber and Dilithium implementation demands significant processing. Dilithium involves distinct functions, some requiring 10^7 cycles at 2.75 GHz ARM processors, projecting sub-second to above-millisecond performance. Figure 1 shows the performance of both

Kyber and Dilithiunm algorithms when using ARM intrinsics optimizations, where CPU cycles can drastically drop to 10^5 CPU cycles..

PQC schemes, encompassing KeyGeneration and Encapsulation/Signing, utilize sub-routines like Number Theoretic Transform and Montgomery Reductions. Parallelizable functions (e.g., NTT) provide up to 57 times speed improvement on DPUs using ARM instructions, while non-parallelizable functions like SHA3 are not explored.

To achieve millisecond performance, intensive functions are offloaded to a dedicated ARM processor for cryptographic hardware accelerations. Results in Figures 2, where data has been taken from [4] show CPU cycles improvements ranging from 5 to 31 for Dilithium and 8 to 57 for Kyber.

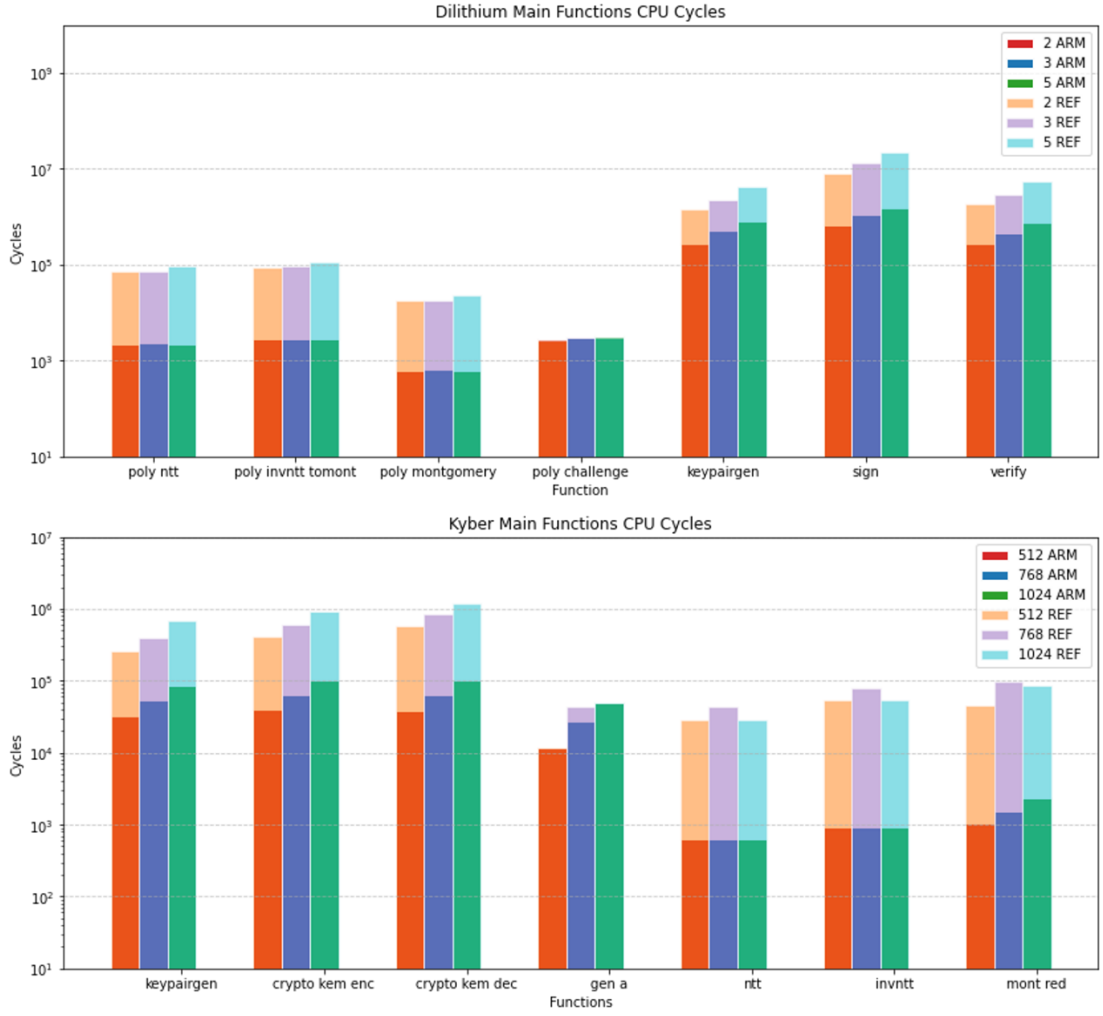

Figure 2: Performance of Dilithium and Kyber main auxiliary functions. A relation is shown between the Reference code and the ARM optimized code

Finally, we show throughput results for our implementation. It can be seen that using the dedicated ARM instructions available on the DPU, the throughput might be improved by a factor of 2.5 in comparison with the reference implementations. We also can see that the throughput offered achieved when using classical cryptography is worse for Elliptic Curve Diffie-Hellman (ECDH) in comparison to its PQ analogous Kyber. On the other hand, Dilithium2 offers a slightly worse performance in comparison to Elliptic Curve Digital Signature Algorithm, showing that migration to Post Quantum Cryptography can be done at almost not cost penalty rather than larger key sizes.

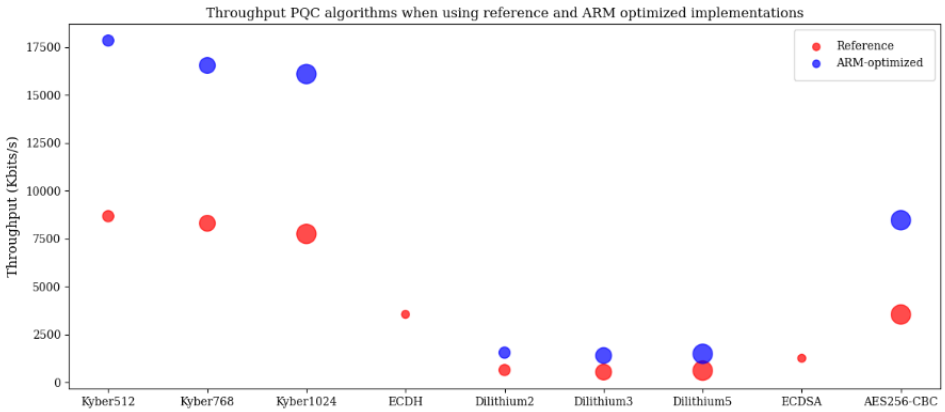

Figure 3: Throughput comparison of Post Quantum Key Exchanges, Digital Signatures and Quantum Resistant symmetric key cryptography.

Conclusion

This work introduced the first end-to-end PQC link established between two server-oriented systems using dedicated DPUs. The DPUs perform all the necessary networking functions to establish a connection with Dilithium and Kyber, combined with a classic AES-256 encryption. We have shown that DPUs HW Acceleration can highly improve the performance of PQC and then meet requirements for high capacity environments.

Furthermore, the letter provides the throughput performance results for a link operating with both reference designs for PQC algorithms and the DPU ARM-optimized algorithms. It demonstrates an increase of over 2.5 times in link capacity when the latter is used, as depicted in Figure 3.

PQC plays a crucial role in ensuring security in a future with quantum computers. This work demonstrates that PQC can be effectively employed in real-world scenarios without substantial latency penalties in the authentication and key exchange phases of communication.

Acknowledgments

This work was partly funded by the QUARC project by the European Union Horizon Europe research and innovation program within the framework of Marie Skłodowska-Curie Actions with grant number 101073355.

References

[1] National Institute of Standards and Technology (2023) Module-Lattice-based KeyEncapsulation Mechanism Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 203 ipd. https://doi.org/10.6028/NIST.FIPS.203.ipd.

[2] National Institute of Standards and Technology (2023) Module-Lattice-Based Digital Signature Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 204 ipd. https://doi.org/10.6028/NIST .FIPS.204.ipd.

[3] Dworkin, M. Et al . (2001), Advanced Encryption Standard (AES), Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.FIPS.197 (Accessed November 13, 2023)

[4] Aguilera, A.C., Clemente, X.A.i., Lawo, D.C., Monroy, I.T. and Olmos, J.J.V. (2023), First end-to-end PQC protected DPU-to-DPU communications. Electron. Lett., 59: e12901. https://doi.org/10.1049/ell2.12901