BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

# Usability of post-quantum cryptography in IoT

Jakub Kratochvíl*

**Abstract**

This work investigates the usability of post-quantum cryptographic algorithms and their resource requirements on constrained devices. For evaluation of the usability of each algorithm, the general parameters on ESP32 with an Xtensa LX6 chip are measured. The second part of this work focuses on network communication sizes of the TLS 1.3 protocol, which uses post-quantum alternatives. Both of these parts include a comparison with current cryptographic algorithms. The results mainly show an increase in resource requirements for post-quantum algorithms in both measurement parts. Post-quantum cryptography is currently usable on devices similar to ESP32, but it is problematic or even unusable on devices with lower resources.

*xkrato67@vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

## 1. Introduction and Overview

**[Motivation]** It is highly possible that within 20 years, the currently used public-key cryptography will be entirely broken by rapidly developing quantum computers [1]. The National Institute of Standards and Technology (NIST) already realized the threat that quantum computers pose to security, and in 2016, proposed a competition for selecting future post-quantum cryptographic algorithms [2].

**[Problem definition]** Since the post-quantum algorithms are in the standardization process, it is important to test them across all possible environments and devices, and even more on devices with limited performance, such as microcontrollers.

**[Related research]** Most of the research regarding post-quantum cryptography has been done in recent years. The most notable works focusing on measurements on constrained devices are the following. First discussed work [3] proposed framework pqm4, created for testing and benchmarking post-quantum cryptographic algorithms on ARM Cortex-M4. Their work has one of the most extensive benchmarks, including almost every post-quantum algorithm proposed by NIST. Another work [4] using ARM Cortex-M4 microcontroller focused on evaluating post-quantum TLS 1.3 on embedded devices. They used pqm4, PQClean, and a customized WolfSSL library for TLS implementation. Lastly, work [5] also focused on post-quantum TLS and evaluating embedded systems. Their work contains measurements for multiple devices, such as LPCXpresso with ARM Cortex M0+, Raspberry Pi 3 Model B+, but most importantly also on ESP32-PICO-KIT V4 with Xtensa LX6 chip, being among the first measurements done on ESP32 microcontrollers. However, their measurements only included two out of four standardized post-quantum algorithms.

**[Proposed solution]** This thesis proposes a two-part measurement of post-quantum cryptographic algorithms in general and real-world use cases. The general measurement focuses on the standalone functions of the post-quantum algorithms on the ESP32-WROOM-32E microcontroller, which uses the Xtensa LX6 chip. This microcontroller was specifically chosen because the Xtensa architecture is significantly less tested than other architectures such as ARM or x86_64. The second part of the testing is focused on a real use case and involves measuring the network communication size of post-quantum TLS 1.3 on different devices. Both of these parts include measurements of classical cryptographic algorithms, against which the post-quantum algorithms are compared.

**[Contributions]** As of our best knowledge, this is the first work that evaluates all post-quantum algorithms selected for standardization on ESP32 with Xtensa LX6 chip. Further, it shows added overhead in performance and memory requirements, as well as in network communication size in TLS 1.3 for post-quantum algorithms.

## 2. Poster Commentary

For better clarity, each section header in the following commentary matches the headers in the poster.

### 2.1 Overview of the measurements

The figure in this section displays the overview of proposed measurements and shows their structure. Two parts of the measurements are shown at the top layer. The one layer under, with purple, green, and yellow colors, shows structuring based on the algorithm types. Finally, measured parameters with the color grey are displayed at the lowest level.

### 2.2 Measured post-quantum algorithms

This section displays four measured post-quantum cryptographic algorithms, which correspond with their color to the mentioned algorithm types. The first post-quantum algorithm in CRYSTALS-Kyber, a key encapsulation mechanism, is thus classified as a key establishment algorithm. The other three post-quantum algorithms, named CRYSTALS-Dilithium, Falcon, and SPHINCS$^+$, are digital signature algorithms. There are also two classical algorithms that were measured that are not displayed. The first one is Elliptic-Curve Diffie-Hellman (X25519) for comparison with the key establishment algorithm, and the second is Edwards-Curve Digital Signature (Ed25519) for digital signatures.

### 2.3 General measurements

The general measurements, presented in orange, are divided based on the measured post-quantum algorithms into the key establishment and digital signature algorithms. For both of these algorithm types, specific measured parameters are used: CPU utilization, runtime memory usage, and static memory usage (binary size). The figure on the left side represents the underlying API functions of the key encapsulation mechanisms and how they are generally used. The figure in the middle shows underlying API functions but for digital signature algorithms. The final results, showcased with graphs at the bottom of the poster, are further divided based on these functions. The last figure in this part, on the right, displays the ESP32-WROOM-32E microcontroller used for this part of the measurements.

### 2.4 Post-quantum TLS 1.3 communication size

The second part, which focuses on the measurements of post-quantum TLS 1.3 communication sizes, is further divided into server-only authentication and mutual authentication. In server-only authentication, only servers send their certificate and signature, whereas in mutual authentication, both the client and server send these messages to authenticate themselves. The upper figure displays messages sent in standard TLS 1.3 handshake almost identical to the ones measured. The second figure displays the proposed public-key infrastructure used in the measurements. For each algorithm combination, such as ECDHE-EdDSA, certificates were generated based on the shown PKI. The public-key infrastructure highly affects the sizes of TLS 1.3 handshake since the certificate messages sent in the TLS handshake include chained certificates up until, but not included, root certificate authority.

### 2.5 Sample of the results

The last section of the poster displays samples of the collected results, located at the bottom left. Unfortunately, due to a large amount of measured data, all eight graphs with the results are not displayed. The two graphs on the left, which have matching colors, are results from general measurements. The upper graph showcases the CPU utilization results of key establishment algorithms. The graph underneath displays the runtime memory usage results of digital signature algorithms. The two graphs on the right are the results of TLS 1.3 communication size measurements, specifically showing the size of the messages transferred in the TLS handshake. The upper graph shows the results of server-only authentication, while the graph at the bottom shows the results of the mutual authentication.

## Conclusion

It was found that the lowest available NIST security levels of post-quantum algorithms can run on the ESP32 without problems. The key encapsulation mechanism Kyber512 exceeded ECDHE (X25519) in execution time and was not drastically worse in the other two parameters. The post-quantum digital signature algorithms did not have a clear winner; instead, each algorithm had advantages and disadvantages in individual parameters. However, each post-quantum variant had much worse CPU utilization and runtime memory usage than the classical EdDSA (Ed25519). The results of the second part show that post-quantum cryptography in TLS 1.3 creates a significant overhead in the size of transferred data compared to classical algorithms. The most significant parts of the post-quantum TLS 1.3 handshake consist of transferred certificates and signatures, which are significantly larger in post-quantum variants.

## Acknowledgements

I would like to thank my supervisor Mgr. Kamil Malinka Ph.D. for his excellent help and consultations whenever needed.

## References

[1] Lily Lidong Chen, Stephen Jordan, Yi-Kai Liu, et al. Report on post-quantum cryptography. *NIST Internal Report 8105*, April 2016. https://doi.org/10.6028/NIST.IR.8105.

[2] Lily Lidong Chen. NIST Post-Quantum Cryptography Standardization. A Workshop About Cryptographic Standards (AWACS), 2016. https://www.cryptoexperts.com/awacs2016/slide-awacs2016/AWACS-PQC-2016-05082016.pdf.

[3] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and benchmarking nist pqc on arm cortex-m4. Cryptology ePrint Archive, Paper 2019/844, 2019. https://eprint.iacr.org/2019/844.

[4] George Tasopoulos, Jinhui Li, Apostolos Fournaris, et al. Performance evaluation of post-quantum tls 1.3 on resource-constrained embedded systems. In Chunhua Su, Dimitris Gritzalis, and Vincenzo Piuri, editors, *Information Security Practice and Experience*, pages 432–451, Cham, November 2022. Springer International Publishing.

[5] Kevin Bürstinghaus-Steinbach, Christoph Krauß, Ruben Niederhagen, et al. Post-quantum tls on embedded systems. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 841–852, New York, NY, USA, 2020. Association for Computing Machinery.