Energy Consumption Evaluation of Post-Quantum TLS 1.3 for Resource-Constrained Embedded Devices*

George Tasopoulos
Industrial Systems Institute, R.C. ATHENA
Patras, Greece
g.tasop@isi.gr

Charis Dimopoulos
Industrial Systems Institute, R.C. ATHENA &
Electrical and Computer Engineering Dpt, University of Patras
Patras, Greece
c.dimopoulos@upnet.gr

Apostolos P. Fournaris
Industrial Systems Institute, R.C. ATHENA
Patras, Greece
fournaris@isi.gr

Raymond K. Zhao
CSIRO's Data61
Sydney, Australia
raymond.zhao@data61.csiro.au

Amin Sakzad
Faculty of Information Technology, Monash University
Melbourne, Australia
amin.sakzad@monash.edu

Ron Steinfeld
Faculty of Information Technology, Monash University
Melbourne, Australia
ron.steinfeld@monash.edu

ABSTRACT

Post-Quantum cryptography (PQC), in the past few years, constitutes the main driving force of the quantum resistance transition for security primitives, protocols and tools. TLS is one of the widely used security protocols that needs to be made quantum safe. However, PQC algorithms integration into TLS introduce various implementation overheads compared to traditional TLS that in battery powered embedded devices with constrained resources, cannot be overlooked. While there exist several works, evaluating the PQ TLS execution time overhead in embedded systems there are only a few that explore the PQ TLS energy consumption cost. In this paper, a thorough power/energy consumption evaluation and analysis of PQ TLS 1.3 on embedded systems has been made. A WolfSSL PQ TLS 1.3 custom implementation is used that integrates all the NIST PQC algorithms selected for standardisation as well as 2 out of 3 of those evaluated in NIST Round 4. Also 1 out of 2 of the BSI recommendations have been included. The PQ TLS 1.3 with the various PQC algorithms is deployed in a STM Nucleo evaluation board under a mutual and a unilateral client-server authentication scenario. The power and energy consumption collected results are analyzed in detail. The performed comparisons and overall analysis provide very interesting results indicating that the choice of the PQC algorithms in TLS 1.3 to be deployed on an embedded system may be very different depending on the device use as an authenticated or not authenticated, client or server. Also, the results indicate that in some cases, PQ TLS 1.3 implementations can be equally or more energy consumption efficient compared to traditional TLS 1.3.

CCS CONCEPTS

• **Hardware** → **Power and energy**; • **Security and privacy** → *Embedded systems security*; • **Networks** → Network reliability.

KEYWORDS

post-quantum cryptography, energy consumption, resource-constrained systems, secure communication protocol

1 INTRODUCTION

The development of Post-Quantum Cryptography (PQC) has inevitably gained significant traction after the discovery that the integer factorization and discrete logarithm problems can be solved in polynomial time [20] by powerful and scalable Cryptographically Relevant Quantum Computers (CRQC) [10]. While the vast majority of the existing cryptographic schemes for public key cryptography are dependent on those problems for their security, PQC schemes rely on other arithmetic problems that are not vulnerable to these new quantum attacks.

In an effort to adapt to this new reality, the US National Institute of Standards and Technology (NIST) has currently completed a standardisation process for PQC algorithms that can mathematically resist quantum attacks, ready to be deployed in real-world applications for digital signature schemes or key exchange mechanisms [8]. For the time being, the 4 algorithms that have been selected for standardisation are 3 Digital Signature Schemes and 1 Key Encapsulation Mechanism (KEM), with some KEM candidates being reevaluated in Round 4 of this standardisation process. An additional Call for Proposals was issued, in order to further diversify the selected Digital Signature algorithms since the finalist schemes are mostly based on lattices.

An essential aspect of the PQC adoption process in real applications is not only the requirement that the PQC schemes should be robust, secure and efficient but also how compatible, efficient and flexible they prove to be when adapted into already existing security protocols and applications. One of the most widely used protocols in this category is the Transport Layer Security (TLS) protocol, responsible for the secure communication of the vast majority of today's Internet. It is, thus, of paramount importance that PQC algorithms are correctly integrated with TLS. This integration and its inevitable performance overhead that is being introduced, while trivial for powerful computing systems, is proving to be a

*Accepted for publication at the Malicious Software and Hardware in Internet of Things Workshop, 20th ACM International Conference on Computing Frontiers '23, May 9th-11th 2023, Bologna, Italy.

challenging issue when dealing with resource-constrained environments [23]. These devices must be holistically evaluated for a wide range of PQC performance metrics, like running time, memory, bandwidth, and network requirements.

However, the power/energy consumption of the device in use is one of the key metrics in the IoT domain. Especially, IoT end devices that operate on battery are significantly affected by the energy consumption of pivotal operations since they can be the main source of such consumption and battery usage. The PQC operations' increased computation overhead compared to traditional schemes as well as the significantly higher byte number of keys, digital signatures and certificates (to be exchanged/communicated between IoT server-clients) can leave a non-trivial footprint on an IoT device energy consumption and eventually drain such a device's battery. Thus the ability of a TLS-integrated PQC scheme to offer a small power/energy consumption footprint while being used within the TLS protocol is highly important in the overall design of an IoT device. This highlights the need for a thorough power/energy consumption profiling/evaluation of TLS 1.3 when various PQC schemes are used.

1.1 Related work

The scope of the current state of research for evaluating the performance of Post Quantum Cryptography (PQC) schemes for KEMs and digital signatures is still in its early stages, especially in regards to resource-constrained environments. A performance evaluation of PQ TLS 1.3 was realized in [23], focusing on the integration of the pqm4 and PQClean libraries on resource-constrained devices. In an effort to benchmark TLS for PQC, a framework was introduced in [12], where the network conditions can be customized and their impact on Post-Quantum TLS implementations can be evaluated, while Paul et al. [13] offer a migration strategy for authentication in a PQ era, while concurrently evaluating the performance of TLS 1.3 in various devices. A Raspberry Pi was used in [2] to perform an evaluation of PQ TLS with OQS [21] library. However, a Raspberry Pi is considered a higher-tiered embedded device and does not operate on the Cortex-M4 which is the reference Microcontroller Unit (MCU) for embedded devices proposed by NIST for the PQC standardisation. In [3], an integration of the CRYSTALS-Kyber KEM scheme and the SPHINCS+ digital signature was implemented into the MbedTLS library. Despite different platforms being tested from a performance perspective, no measurements for accurate energy consumption were provided. Various performance metrics were also assessed in [19], with only empirical tests employed for an energy consumption estimation.

For the works focusing in energy requirements, the authors of [17] conducted an evaluation of the energy consumption of PQC candidates on an Intel Core i7-6700 CPU and ranked them based on the results. These algorithms' versions however are based on the Round 2 of the NIST standardisation competition. The work in [18] evaluates the overhead in energy and latency demand that the transition to PQC algorithms will cause, focusing on the TLS handshake. However, the TLS 1.2 version is utilized and the energy consumption comparisons do not include digital signature schemes, while some of them have currently been disqualified from the final results of the NIST competition. Additionally, another energy efficiency comparison for PQC algorithms was carried out in [1]. The differentiating factor in this work is the effort to use pre-quantum hardware accelerators and compare them to a RISC-V software implementation. Despite the speedup, the algorithms tested are also based on the Round 2 of the NIST competition and no full TLS handshake was executed. Lastly, in [11], an extensive energy requirement analysis for the new PQC candidates of the NIST standardisation competition was realized using an STM X-NUCLEO_LPM01A "PowerShield", a programmable power supply for NUCLEO devices that can also measure power consumption. The PQC algorithms were executed on a NUCLEO-F411RE development board and the collected measurements were incorporated into guidelines for correct PQC algorithm usage in IoT. These measurements, however, do not incorporate TLS 1.3 evaluation, but only analysis of PQC primitives. To the authors' knowledge, there is no complete work on real-time energy consumption evaluation for TLS 1.3 using PQC in resource-constrained devices.

1.2 Contribution

In this paper, extending the work in [23], an exhaustive power/energy consumption profiling of TLS 1.3 protocol when integrating various NIST PQC standardisation candidates is made and a comparison between them is performed in order to determine the energy consumption overhead each PQC algorithm combination introduces to the TLS 1.3 protocol. [1] The setup used includes an oscilloscope for accurate power measurements, a Cortex-M4 MCU board, the reference platform for embedded devices for the NIST PQC standardisation process and a laptop that acts as a server. More specifically, the power and energy consumption cost of a PQ TLS 1.3 handshake was evaluated in the two most typical scenarios in the IoT world:

- When a resource-constrained embedded system (an end-node) is connected to another device (an end-node or a more powerful sever) and both are mutually authenticated.
- When a resource-constrained embedded system (an end-node) is connected to another device (usually a more powerful server) and only the latter is authenticated.

Several PQC algorithms were integrated into TLS 1.3: All NIST PQC algorithms selected for standardisation; 2 out of 3 NIST PQC Round 4 KEMs; the KEM that the BSI (Germany's Federal Office for Information Security) recommends in its technical references for quantum resistance. Specifically, CRYSTALS-Dilithium, Falcon, SPHINCS+, CRYSTALS-Kyber, BIKE, HQC and FrodoKEM.

After integrating these PQC algorithms into TLS 1.3, a series of experiments were conducted in the two aforementioned scenarios and power and energy consumption measurements were taken. Additionally, the primitive operations of the PQC algorithms mentioned, were evaluated. Even though the primitive operations measurements does not introduce any novelty over [11], they were included for the sake of completeness, to support the PQ TLS 1.3 results as well as due to the different experimental setup and devices used, which led to different absolute values.

It should also be noted that even though the PowerShield from ST is the most common way to measure the power consumption of

[1]The source code is publicly available as two repositories at
https://gitlab.com/g_tasop/pq-wolfssl-for-embedded
https://gitlab.com/g_tasop/pq-wolfssl-for-pc

a Nucleo board, it seemed inadequate to measure the overall power consumption of a Nucleo board when running a full PQ TLS 1.3 handshake (it exceeded the 50mA current threshold for dynamic measurements [22]). Thus, we decided to use instruments that offer higher accuracy and higher current or voltage limits. The PicoScope 5444B [24] is a flexible, high-performance PC oscilloscope that was used in pair with PicoScope 7 software[2] to take measurements of the overall power consumption of a Nucleo board when running a PQ TLS 1.3 handshake.

The rest of the paper is structured as follows: In Section 2, the necessary background information on TLS 1.3 is described, while in Section 3 the experimental setup used is presented. In Section 4 the results of this evaluation process are presented and lastly, Section 5 concludes the paper.

2 BACKGROUND

2.1 TLS Protocol

The potential existence of a Cryptographically Relevant Quantum Computer (CRQC) [10] poses a threat to real-world security protocols that are utilising public key cryptography as their foundation. Transport Layer Security (TLS) protocol is among the most affected, due to its heavy usage of public key cryptography and its wide adoption. It is used by the majority of the applications wanting to communicate securely over the Internet (email servers [5], website transfer [15], smartphone apps [14]) making it the de facto standard. The latest version (TLS 1.3) [16] is currently at an adequate level of adoption, mainly due to the centralisation of the Internet [6]. In TLS 1.3, a whole round-trip is eliminated, in comparison to the previous version (TLS 1.2) and some potentially insecure primitives are deprecated, making it faster and more secure [16].

TLS consists of two main components: the *handshake protocol* and the *record protocol*. The handshake protocol is in charge of negotiating cryptographic parameters, establishing shared key material and authenticating communicating parties and to do this it is utilising public key cryptography (Diffie Hellman, Elliptic Curve cryptography, RSA). Record protocol is using symmetric key ciphers (AES, ChaCha) to secure the messages that the two parties exchange. While a Quantum Computer could potentially utilize Grover's algorithm [4] to speed up an attack on symmetric ciphers, it is not considered a serious threat. NIST is considering AES 128, which offers 128-bits of security, to be secure against quantum computers for the foreseeable future and a conservative approach would be to double the key length, thus doubling the 'bits' of security [9]. On the other hand, the public key cryptographic primitives of the protocol are seriously vulnerable against a CRQC running Shor's algorithm and need a replacement that would provide security against such attacks. The integration of post-quantum public key algorithms in such a protocol is mitigating this threat.

2.2 TLS implementations

In this paper, the WolfSSL implementation of the TLS protocol is chosen, because it is licensed under a GPLv2 open-source license and it is specifically targeting resource-constrained embedded devices. Also, it has implemented TLS version 1.3, in contrast with other popular open-source libraries (eg. MbedTLS).

In recent versions of WolfSSL, support for CRYSTALS-Dilithium, Falcon, and SPHINCS+ has been added using code from liboqs [25]. Also, CRYSTALS-Kyber has been present in the WolfSSL library but the code is available only after contacting WolfSSL[3]. However, these implementations are not targeting resource-constrained systems and are unoptimized for the Cortex-M4, with the exception of a limited experimental setup using Kyber512 directly from pqm4 [26].

In the work of [23], the WolfSSL library has been extended in order to support all the ARM Cortex-M4 optimised versions of the PQC algorithms. Any implementation that was not present in the pqm4 repository was complemented with the respective implementation from PQClean [7]. More specifically, the authors in [23] used the pqm4 implementation for CRYSTALS-Dilithium, Falcon, CRYSTALS-Kyber, BIKE, FrodoKEM and HQC, the PQClean implementation for SPHINCS+ and HQC. Adopting this paper's approach of this paper, we used the paper's code libraries and used them for our evaluation board, acting as the starting point of our research. The security levels of each of these deployed PQC algorithms are noted in Table 1 and are discussed further in Section 4.

2.3 PQC Modifications

In order to make TLS quantum-safe, architectural changes were made to the WolfSSL implementation and changes were introduced to the protocol itself. Two TLS message fields were altered, called "Extension Fields": "Supported Groups" and "Signature Algorithms". Additionally, changes were made in order to use KEMs instead of the classic Diffie-Hellman Key Exchange methods, as well as for the protocol to use post-quantum authentication both in Digital Signatures and in the Certificates employed. These changes are discussed in detail in [23] and are displayed in Figure 1.

3 EXPERIMENTAL SETUP

In our work, the power/energy consumption of the PQC TLS 1.3 was evaluated in the two most typical IoT scenarios:

Scenario 1 (mutual authentication): When a low-end, resource-constrained embedded device (the end-node) is connected to another device, another embedded device or maybe a more powerful device, and they are mutually authenticated, e.g. in an MQTT configuration where a number of devices need to communicate in a secure way to exchange data, before transmitting them to a server.

Scenario 2 (unilateral authentication): When our end-node is connected to a more powerful device (the server) and only the latter is authenticated, e.g when a sensor connects to a cloud server to transmit the data it has collected. In this scenario, the end-node, which acts as a TLS client, performs a TLS handshake with the server and only the server is authenticated.

In our experimental setup, the following devices were used: A Nucleo-F439ZI, which is equipped with a 32-bit ARM Cortex-M4 running at 180 MHz, with 192 KB of usable SRAM (plus 64 KB of CCM RAM that is not utilised) and 2 MB of Flash memory. It also provides Ethernet connectivity and a series of GPIO pins available

[2] Available at https://www.picotech.com/downloads

[3] https://github.com/wolfSSL/wolfssl/blob/master/wolfcrypt/src/wc_kyber.c#L25

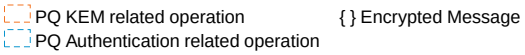

Figure 1: Post-quantum TLS 1.3 handshake messages.

to the user. This is a typical device for IoT end-nodes in such con- gurations and it additionally incorporates a Cortex-M4 core that NIST has selected as their reference platform for embedded devices in the PQC process. The second device is a laptop that is running Ubuntu 20.04 in x86_64 architecture and it is equipped with an Intel i7-1165G7 with 8 cores running at 2.8 GHz. This might not be a server-grade machine, but it is adequately more powerful than the embedded device and can act as a cloud server.

Both machines are connected to the same access point through the Ethernet interface, with a mean Round-Trip Time (RTT) of 0.493 ms, without any intermediate gateways (note that the Nucleo board is equipped with an Ethernet connector).

For the power consumption measurements, we used the Pico-Scope 5444B from the PicoScope 5000 Series [24], able to capture traces with up to 1 GS/s real-time sampling. In our experiment, we used a sampling rate of 5 MS/s, which is adequate to capture power consumption measurements and at the same time provides a large enough time window (5 seconds) to execute the operations under evaluation many times. In some exceptions, like a TLS handshake with SPHINCS+, where the total capture time should be a lot more than 5 seconds, a lower sampling rate was used, 833 kS/s, and a larger time window, 100 seconds. Two probes were used in a dif- ferential setup, to capture the Voltage trace across a small shunt" resistor (1.5 Ohm) that was inserted in the IDD jumper of the board. This jumper (IDD) is speci cally used to measure the current, with an Ammeter, that the 3.3V voltage regulator supplies to the whole circuit of the board, meaning the MCU, the peripherals, etc. In other words, we measured the realistic energy consumption of the whole resource-constrained embedded system in the two most typical scenarios in an IoT network, while being Quantum-Safe.

In Fig. 2, the power/energy consumption measurement setup can be seen. Two probes are connected to the shunt" resistor, which

Figure 2: The Nucleo board and the PicoScope in the energy consumption con guration.

is inserted in the IDD jumper through some jumper cables and a third probe to a GPIO pin, through a plain wire. This is the D7 pin of the microcontroller and it is used as the "trigger" signal for the PicoScope. All the probes have their ground connected to the extra ground pins in the perimeter of the Nucleo board, an Ethernet cable is connected to the local access point and a USB cable is connected to a PC to power, program and control the board via a serial port.

A measurement capture is completed with the following steps:

- The rst device (either the end-node or the server) runs as the TLS 1.3 server.
- The second device (either the server or the end-node re- spectively) runs as the TLS 1.3 client and connects to the TLS 1.3 server.
- When the two devices are connected at the TCP level and before the TLS 1.3 handshake begins, the end-node triggers the D7 pin and the PicoScope begins to measure the voltage of the probes in the di erential format.
- The measurement continues as the two devices continu- ously perform the following: a complete TLS 1.3 handshake and an exchange of 2048 bytes of data in a loop until the 5 seconds window has passed.
- When the measurements are complete, the PicoScope soft- ware utility is used to compute the average power consump- tion, knowing the voltage trace across the "shunt" resistor and the value of the resistor.
- The average energy consumption can be computed by mul- tiplying the average power consumption that we computed, with the average time measurement of the TLS 1.3 hand- shake that it was captured as well.

The process is the same for both experimental Scenarios, either when the board acts as a server or as a client. The only exception is TLS 1.3 with SPHINCS+ where a lower sampling rate is used and thus a larger time window to be able to t its timely operations.

4 MEASUREMENTS, EVALUATION AND RESULTS

In this section, the power and energy consumption measurements regarding the PQC algorithm primitive operations and the PQ TLS 1.3 are presented and analyzed in depth. It should be noted that in all collected measurements, in order to evaluate the PQC algorithms and TLS 1.3 in real conditions, we have included the average power