Energy Consumption Analysis of NIST Selected Post-Quantum Digital Signature Algorithms for Quantum Secure Communications

Atinderpal Singh Lakhan
Carleton University
Ottawa, Canada
atinderpalsinghlakh@cmail.carleton.ca

Mostafa Taha
Carleton University
Ottawa, Canada
mostafa.taha@carleton.ca

***Abstract*—This paper conducts an analysis of energy consumption in NIST selected digital signature algorithms, a pivotal stride in the preparation for post-quantum cryptography and the expansion of the IoT-connected secure ecosystem. Given the significance of energy-efficient cryptographic solutions for battery-operated devices' performance and operational longevity, this study offers a comprehensive assessment of energy usage in key generation, signing, and verification processes. The algorithms scrutinized encompass the NIST selected algorithms like Sphincs+, Falcon, Crystals-Dilithium, Picnic, Rainbow, and also consider MPPK/DS as a candidate. Notably, this study employs the Apple M1 chip for analysis, distinct from previous studies employing Intel's Skylake processor. Results highlight Picnic's key generation excellence (0.424 ms, 11.04 W), closely trailed by MPPK/DS (0.567 ms, 13.12 W). Falcon and Crystals-Dilithium emerge as energy-efficient in signing (2.962 mJ, 4.472 mJ), while Crystals-Dilithium and Rainbow excel in verification (0.647 mJ, 1.143 mJ). Falcon and MPPK/DS also exhibit commendable efficiency. Picnic and SPHINCS+256S display moderate energy usage due to extended execution times. This study's utilization of the M1 chip establishes a benchmark for energy efficiency analysis in post-quantum cryptography. Future endeavors will extend its scope to encompass a broader platform exploration and the inclusion of other NIST-selected encryption/decryption algorithms.**

***Index Terms*—NIST, PQC, Cryptography, Sphincs+, Falcon, Crystals-Dilithium, Picnic, Rainbow, and MPPK/DS**

I. Introduction

For battery-operated devices, energy efficiency is a paramount consideration. The execution of cryptographic algorithms in resource-constrained environments can significantly impact device performance and operational lifespan. By exploring the energy consumption of post-quantum cryptographic algorithms, we can identify energy-efficient solutions that extend the battery life of devices and optimize resource utilization

In addition to the submitted post-quantum cryptographic algorithms, This research aims to provide a comprehensive evaluation of the energy consumption of post-quantum cryptographic algorithms, focusing on MPPK/DS [1] and other prominent candidates. Our analysis spans a range of algorithms, including SPHINCS+256S [2], Falcon [3], Crystals-Dilithium [4], Rainbow [5], Picnic [6] which are the part of lattice-based cryptosystems, code-based cryptosystems, and hash-based signatures. By comparing their energy consumption on different platforms, we can gain valuable insights into their suitability for real-world applications.

By measuring energy efficiency on both microprocessors, we aim to facilitate informed decision-making in deploying energy-efficient cryptographic solutions, particularly in the context of IoT and resource-constrained environments. The results of this investigation contribute valuable knowledge to empower future applications and ensure security in the face of quantum computing challenges. This paper is the continuation of the previously published research study in which we produced the results of network performance analysis and key correlation with semi-covariance analysis for the various PQC digital signature algorithms [7]. By subjecting different algorithms to energy consumption tests, we seek to identify the algorithm that operates with optimal energy efficiency. Energy consumption is a critical factor, particularly in resource-constrained environments, and selecting an algorithm that minimizes energy usage is vital for practical implementations. Alongside evaluating the NIST selected algorithms, our focus extends to a new algorithm, the MPPK/DS (Multivariate Polynomial Public Key Digital Signature), as a promising candidate for the PQC competition.

Additionally, to ensure transparency and reproducibility, all codes and artifacts developed for this study, including the Python implementation of the MPPK/DS algorithm, are made available online on GitHub. By providing open access to the research materials, other researchers and practitioners can validate and build upon the findings, fostering collaboration and advancing the field of post-quantum cryptography. The results of this comprehensive research contribute in benchmarking various algorithms while ensuring the integrity and confidentiality of sensitive data in the face of emerging quantum computing threats.

A. Contribution

The objective of this study is to evaluate NIST selected digital signature algorithm in terms of energy consumption. In particular, the contributions are as follows:

- Generating bash script in order to log the power consumption of the algorithms in test
- Energy consumption analysis of all the key-generation algorithms of different digital signatures
- Energy consumption analysis of all the signing algorithms of different digital signatures
- Energy consumption analysis of all the verifying algorithms of different digital signatures
- Considering MPPK/DS as one of the additional algorithm in test and benchmarked with NIST selected signatures.
- Promoting reproducible results by making all codes and artifacts developed for this study available online on Github[1].

II. Related Work

Classical cryptographic schemes, which rely on the difficulty of certain number theoretic problems, have been widely used to ensure security in various applications. However, the advent of large-scale quantum computers poses a significant threat to the security of these schemes. As a result, the National Institute of Standards and Technology (NIST) has initiated the Post-Quantum Cryptography (PQC) standardization process, evaluating candidate algorithms that can withstand quantum computing threats.

The evaluation of post-quantum cryptographic algorithms goes beyond security and performance metrics; it also considers the energy consumption of these algorithms, especially in the context of the increasing use of battery-operated devices and growing interest in green computing [8]. Understanding the energy consumption of different cryptographic algorithms is crucial for optimizing their implementation and making informed decisions about their suitability for resource-constrained devices and energy-sensitive environments.

In this literature review, we explore the research works that have experimentally measured the energy consumption of various post-quantum cryptographic algorithms, focusing specifically on digital signature schemes. We will discuss the findings from research papers which shed light on the energy efficiency of different cryptographic algorithms and provide valuable insights into their performance on energy-constrained platforms.

The study [9]by Beckwith et al. investigates the FPGA energy consumption of post-quantum cryptographic algorithms, particularly key encapsulation mechanisms (KEMs) and digital signature schemes [10]. The study compares the energy consumption of lattice-based algorithms, such as Saber and BIKE, to code-based algorithms like Streamlined NTRU Prime. It demonstrates that lattice-based algorithms exhibit significantly lower power requirements than code-based algorithms, making them more energy-efficient in practice. Additionally, the paper highlights the potential of FPGA implementations to reduce energy consumption of these cryptographic operations.

In another study by Farahmand et al., the energy consumption of NIST PQC Round 2 candidates is experimentally measured, including both Optimized C Implementations and Assembly Optimized Implementations [11]. The study ranks the candidates based on their energy consumption, revealing that lattice-based schemes, such as Dilithium and Falcon, tend to be highly efficient in practice. It also shows that multivariate-based schemes are competitive with lattice-based counterparts, especially when assembly instructions are utilized. The paper emphasizes the need for a holistic approach in selecting algorithms based on various evaluation criteria, including energy consumption.

Furthermore another study presents an evaluation of PQC candidates' energy consumption on an Intel Core i7-6700 CPU, analysing both Optimized C Implementations and Assembly Optimized Implementations [12]. It provides insights into the energy consumption profile of different cryptographic functions and proposed security levels. Notably, lattice-based algorithms demonstrate lower median energy consumption across all security levels for key encapsulation and digital signatures. This study emphasizes the importance of considering energy efficiency when selecting post-quantum cryptographic algorithms.

Overall, There are only a few number of studies which collectively contribute to the understanding of energy consumption in post-quantum cryptographic algorithms, specifically for digital signature schemes. They demonstrate the advantages of lattice-based algorithms in terms of energy efficiency and parallelizability. These findings highlight the potential benefits of utilizing FPGA implementations and vectorized instructions for further energy savings. The comparison between different cryptographic families and their energy consumption characteristics assists in pinpointing potential areas for optimization.

Furthermore, this study aims to address the void in energy consumption analysis of post-quantum cryptographic algorithms. While existing research has explored the energy efficiency of various algorithms on different platforms, our study will uniquely contribute by conducting energy consumption analysis on the M1 processor, which is becoming increasingly relevant in the domain of resource-constrained devices and IoT applications. By benchmarking the energy performance of the MPPK/DS algorithm on the M1 processor, this study aim to provide valuable insights into the algorithm's suitability for energy-sensitive scenarios. Through this research, the study aim to aid in the selection and optimization of post-quantum cryptographic schemes, ensuring secure and energy-efficient implementations on cutting-edge platforms like ARM-based systems-on-a-chip processor M1.

III. Methodology:

In this section, we outline the methodology adopted for evaluating the energy consumption of various post-quantum cryptographic algorithms. Our analysis encompasses a selection of submitted post-quantum cryptographic algorithms,

[1]Github.com/AtinderPalSingh/Analysis-on-MPPK-DS-algorithm

including SPHINCS, Falcon, Crystals-Dilithium, MPPK/DS, Picnic and Rainbow which are the subset of lattice-based cryptosystems, code-based cryptosystems, and hash-based signatures. By leveraging the Mx power gadget, an open-source software designed for measuring power consumption on Apple's M1 chip, we aim to gain valuable insights into the energy efficiency of these algorithms. The methodology is structured as follows:

Selection of Algorithms and Platform

- Analyze a range of post-quantum cryptographic algorithms submitted to NIST.
- Also selected MPPK/DS for comparison, implemented in python and benchmarked with other NIST selected algorithms
- Perform evaluations on the Apple M1 chip as a representative microprocessor platform.

Data Collection and Implementations

- Download and utilize code files and implementations from the NIST submissions.
- Opt for 'optimized implementations' to ensure a fair comparison, considering C implementations without vectorization.
- Account for encryption/encapsulation submissions with specific security considerations.

Power Consumption Logging

- Prepared a bash script to enable logging of power consumption on the Mx power gadget. MX Power Gadget is a software tool specifically designed for monitoring power consumption on Apple's M1 chip. Similar to its counterpart, Intel Power Gadget, which supports Windows OS, MX Power Gadget provides access to the processor energy counter, allowing calculation of power usage in watts. The tool records power consumption data at a user-defined sampling rate and logs it into a .csv file. As MX Power Gadget is tailored to the M1 chip, it provides valuable insights into the energy consumption of cryptographic algorithms executed on this specific microprocessor.
- Preparing the script in order to collect the correct recordings when the algorithm executes.
- Start logging when the corresponding signature algorithm is executed.
- Stop logging once the algorithm completes the generation of signatures and verification.

Data Analysis

- Extract power consumption and timestamps from the collected data.
- Focus on the package power in watts as the desired metric for energy consumption.
- Correlate power and time data to derive energy consumption in millijoules.

Results Presentation

- Present the analysis findings graphically to facilitate algorithm comparisons.

```
#!/bin/bash
# Function to measure power consumption using powermetrics and capture the
process ID
measure_power_consumption() {
  mx_power_gadget -l mx_power_gadget_log.csv &
  sudo powermetrics --samplers smc -i 1000 -n $1 > $2 2>&1 &
  powermetrics_pid=$!
  wait $powermetrics_pid
}
# Function to execute the PQC algorithm
execute_algorithm() {
  echo "Executing the  PQC algorithm..."|
/path/to/your/algorithm_executable
  ./kat512int
  echo "Falcon PQC algorithm executed."
}
# Main function
main() {
  # Set the duration for power measurement in milliseconds
  duration=1300
  # Set the output file for powermetrics data
  power_metrics_output="updated_power_metrics.csv"
  # Measure power consumption and capture the process ID of powermetrics
  measure_power_consumption $duration $power_metrics_output

  # Execute the  PQC algorithm
  execute_algorithm

  # Create a CSV file with header for power metrics data
  echo "Timestamp,Processor_Watt,GPU_Watt" > updated_power_metrics.csv

  # Parse the power metrics data and extract relevant values
  power_metrics_data=$(grep "^[0-9]*,[0-9]*,[0-9]*" $power_metrics_output)

  # Process each line of power metrics data
  while read -r line; do
    # Extract timestamp, processor power, and GPU power
    timestamp=$(echo $line | cut -d ',' -f 1)
    processor_power=$(echo $line | cut -d ',' -f 2)
    gpu_power=$(echo $line | cut -d ',' -f 3)

    # Append the data to the CSV file
    echo "$timestamp,$processor_power,$gpu_power" >>
updated_power_metrics.csv
  done <<< "$power_metrics_data"
  echo "Power consumption data logged in updated_power_metrics.csv."
}

# Run the main function
main
```

Fig. 1: Bash script for logging power consumption

- Report results in tabular format for a comprehensive overview.

This methodology enables us to assess the energy consumption of post-quantum cryptographic algorithms, providing valuable insights into their suitability for real-world applications, particularly in energy-constrained environments like IoT. The results of this analysis contribute to the field of post-quantum cryptography and aid in selecting energy-efficient cryptographic solutions.

A. Bash script:

The following set of commands are used in conjunction in order to log data into.csv file from where the further results are correlated and graphical data analysis is performed . The following script is an example of key generation algorithm of FALCON algorithm, where the algorithm is run while the power consumption and time stamps are recorded in the "updatedpowermetrics.csv" file.

IV. Results:

In this section , the energy consumption results for various post-quantum cryptographic (PQC) algorithms are presented. The investigation focuses on three key aspects: key generation, signing algorithm, and verification algorithm energy consumption. The PQC algorithms considered for this study encompass the most prominent digital signature schemes, which have been selected as finalists by the National Institute of Standards and Technology (NIST). The selected algorithms include Falcon, SPHINCS+256S, Rainbow, Picnic, MPPK/DS, and Crystals-Dilithium.

For each PQC algorithm, the power consumption in watts and execution time in milliseconds during key generation, signing, and verification processes were measured. The average power consumed was then multiplied by the total execution time to obtain the total energy consumption in millijoules. By analyzing the energy consumption across these different cryptographic operations, insights into the computational demands and resource requirements of each algorithm can be gained.

A. Key Generation Energy analysis

SPHINCS+256S[[2]]- SPHINCS+256S is a post-quantum cryptographic algorithm known for its compact key sizes and large signature size. It utilizes public key sizes of 64 bytes, private key sizes of 128 bytes, and signature sizes of 29,792 bytes. This algorithm offers two variants: SPHINCS+-'robust' and SPHINCS+-'simple. The 'simple' instantiations are designed with a focus on better speed, providing a more efficient implementation. However, their security argument applies only in the random oracle model, making them less robust against potential attacks. On the other hand, the 'robust' instantiations have a more conservative security argument, ensuring stronger resistance against various types of attacks. However, this increased security comes at the cost of slower computation speed compared to the 'simple' instantiations. Given the trade-offs between speed and security, SPHINCS+256s is considered in this study to provide a basis for comparison with other algorithms. The compact key sizes and large signature size of SPHINCS+256s are essential factors to evaluate its suitability for real-world applications and its efficiency in terms of energy consumption. Analyzing the energy consumption of SPHINCS+256s will contribute valuable insights into the algorithm's practicality and effectiveness, aiding in the broader understanding of post-quantum cryptographic algorithms' adoption and performance for which the power consumption of SPHINCS+256S key generation algorithm is as depicted in figure 2 below.

Total execution time: 263.34 milliseconds and Average power consumption: 14.06 watts

Note: The execution of the actual algorithm starts from the second peak, as shown in the graph, and the corresponding execution time with power consumption in watts is recorded. However, it is important to note that the first peak represents other miscellaneous commands used to produce the executable file for the algorithm in use. These commands may include makefile procedures, but they are not considered as additive to the time and power consumed by the algorithm itself. Only the execution time and power consumption during the actual algorithm run are taken into account for analysis and same applies for all the algorithms in test.

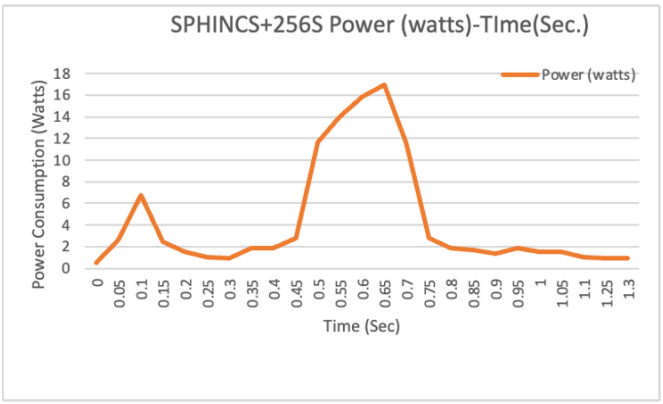

Fig. 2: SPHINCS+ 256S key generation power consumption

FALCON[[3]] :- Falcon, a lattice-based post-quantum cryptographic algorithm, prioritizes resistance to quantum computers while aiming for reasonable efficiency in the current non-quantum world. It offers compact signature sizes, efficient key generation, and fast signing operations. The study evaluates Falcon's energy consumption in terms of power (watts) versus time (milliseconds) as in Figure 3, shedding light on its practicality and effectiveness in real-world applications. These results contribute valuable insights to post-quantum cryptographic algorithms' adoption and performance. **Total Execution time: 11.07 milliseconds and Average power consumed in watts: 12.48 watts**

MPPK/DS[[1]] is a post-quantum cryptographic algorithm that leverages several optimizations to enhance its security and efficiency. One notable feature of MPPK/DS is its ability to provide an additional security level of 384 bits of entropy within a 128-bit field. Further for the implementation in test, this study uses key generation algorithm of MPPK/DS to formulate the results below depicted in figure 3. The algorithm achieves this by employing advanced cryptographic techniques, making it suitable for secure communications in the post-quantum era **Total execution time: 0.567 milliseconds and Average power consumption: 13.12 watts**

PICNIC[[6]] Picnic is a post-quantum cryptographic algorithm that combines non-interactive zero-knowledge proofs of knowledge and Multiparty Computation (MPC). This approach enables fast and efficient computations while ensuring privacy and security. The key generation execution time and power consumption is depicted in figure 3. Picnic's speed and secure design make it a compelling choice for secure communications in the post-quantum era. **Total execution time: 0.424 milliseconds and Average power**

consumption: 11.04 watts

CRYSTALS-DILITHIUM[[4]] : Crystals-Dilithium is an efficient post-quantum cryptographic algorithm known for its carefully optimized signatures based on the hardness of ideal or module lattice problems. One of its key features is the reduction of the public key size by approximately 2.5 times, resulting in significant space savings of over 2KB. This optimization comes at the cost of adding an additional 100 bytes to the signature size. Despite the larger signature size, Crystals-Dilithium strikes a balance between small signatures and public keys, fast signing and verification speeds, and simple constant-time implementations. The key generation of crystals-Dilithium consumes power as depicted in Figure 3. These attributes make it a highly efficient signature scheme, making it a strong contender for secure communications in the post-quantum era. **Total execution time: 2.465 milliseconds and Average power consumption: 14.71 watts**

RAINBOW[[5]] Rainbow is a post-quantum cryptographic algorithm known for its compact signature length, making it suitable for bandwidth-constrained environments. In its classic form, Rainbow signatures are only 66 bytes, providing efficient data transmission and storage. However, one of the trade-offs of Rainbow lies in its key sizes. The public key size is relatively large, at 161,600 bytes, and the private key size is 103,648 bytes. These large key sizes can have implications for memory usage and storage requirements. The key generation power consumption for Rainbow is depicted in Figure 4. Analyzing the energy consumption during key generation is crucial for assessing the algorithm's practicality and efficiency in real-world applications. Given its compact signature size and large key sizes, Rainbow presents interesting characteristics for consideration in this study. Evaluating its energy consumption and efficiency will provide valuable insights into its suitability for various use cases and its performance relative to other post-quantum cryptographic algorithms. **Total execution time: 353.45 milliseconds and Average power consumption: 15.30 watts**

Comparison of the key generation algorithms: Among the algorithms considered for key generation, Picnic stands out as the most efficient. It demonstrates the shortest key generation time of 0.424 milliseconds and the lowest average power consumption of 11.04 watts. This showcases superior speed and power efficiency during the key generation process. MPPK/DS also shows competitive performance in key generation, with a total execution time of 0.567 milliseconds and an average power consumption of 13.12 watts. While slightly slower than Picnic, MPPK/DS remains a highly efficient algorithm for key generation. Crystals-Dilithium, SPHINCS+256S, and Rainbow exhibit relatively higher key generation times. Crystals-Dilithium has a total execution time of 2.465 milliseconds and an average power consumption of 14.71 watts. SPHINCS+256S shows a total execution time of 263.34 milliseconds and an average power consumption of 14.06 watts. Rainbow demonstrates the longest key generation time among the algorithms, with a total execution time of

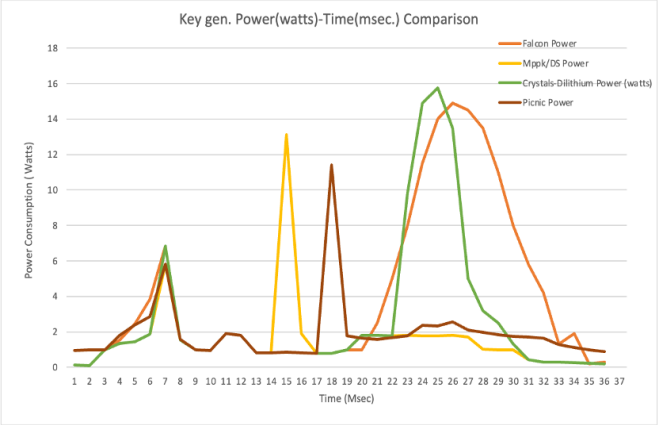

Fig. 3: Key-generation algorithm comparison for power consumption-Falcon,MPPK/DS, Crystals-Dilithium, and Picnic

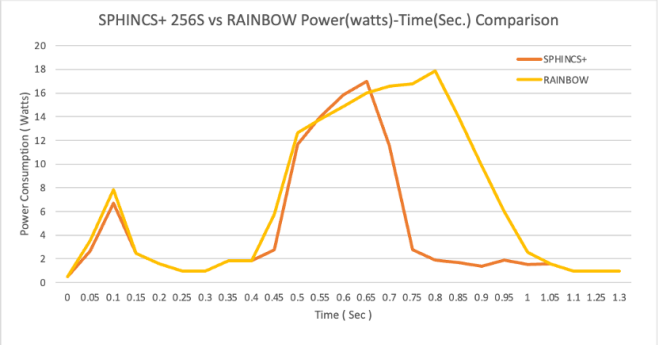

Fig. 4: Key-generation algorithm comparison for power consumption- SPHINCS+256S and RAINBOW

353.45 milliseconds and an average power consumption of 15.30 watts. In conclusion, Picnic and MPPK/DS demonstrate the best key generation efficiency among the algorithms considered. Picnic's remarkably short execution time and low power consumption make it the most efficient choice for key generation, followed closely by MPPK/DS. For applications that prioritize rapid and power-efficient key generation, Picnic would be the preferred algorithm. The same is depicted in Figure 3 and figure 4 Below, where figure 3 compares Falcon, MPPK/DS, Picnic and Crystals-Dilithium as the time of execution is comparable among them in terms of milliseconds however, Rainbow and SPHINCS+256S are compared and depicted in Figure 4 with much higher execution time.

B. Energy Consumption for Key-generation algorithms:

In this section, we will delve into the energy consumption of the post-quantum cryptographic algorithms considered in our study. The energy consumed by a cryptographic operation can be calculated by integrating the power consumption over time. The formula for energy consumption can be represented as follows:

$$E = \int p \cdot (t)\, dt$$

Key-gen algorithm	Execution time (Milliseconds)	Average Power consumed (Watts)	Energy consumed (Millijoules)
FALCON	11.07	12.48	**138.1536**
MPPK/DS	0.567	13.12	**7.4390**
PICNIC	0.424	11.04	**4.680**
CRYSTALS-DILITHIUM	2.465	14.71	**36.260**
SPHINCS+256S	263.34	14.06	**3702.560**
RAINBOW	353.45	15.30	**5407.785**

TABLE I: Energy Consumption for Key-generation Algorithm of different PQC digital Signature Schemes

where E denotes the energy consumed in joules, P(t) represents the power consumption in watts as a function of time t, and the integral integrates the power consumption over the entire execution time of the cryptographic operation while the time taken by the make file command i.e. building the file is excluded. Now, let's examine the energy consumption of each algorithm based on the provided data. All the values for energy consumption of key-generation algorithms are depicted in the comparison table1 below:

C. Signing and Verifying energy Analysis:

This section analyzes the energy consumption of post-quantum cryptographic signing algorithms. Similar to the key generation analysis, energy consumption during signing is calculated by integrating power consumption over time. We use the same approach as before, leveraging average power consumption and execution time data to compute the energy consumption for each algorithm's signing operation. By comparing these values, we gain insights into the efficiency of power consumption during signing, contributing to a comprehensive understanding of the energy performance and practical implications of each algorithm's signing process.

Falcon: In Falcon-512, the signature size is a critical metric that reflects its effectiveness in achieving both security and efficiency. In this study, the signature size for Falcon-512 is approximately 666 bytes. The process of generating a key pair involves the selection of two polynomials: f and g. From these polynomials, essential components are computed, such as, $F = fq = f^{(} - 1)$ modq where f and fq are considered the private keys. The public key is defined as h=pfqfmodq. For Falcon-512, which boasts an equivalent security level to RSA-2048, the resulting key sizes are as follows: a public key of 897 bytes, a private key size of 1,281 bytes, and a signature size of 690 bytes. This blend of key sizes demonstrates Falcon-512's meticulous design in striking a balance between security considerations and efficient resource utilization.*Total signing execution time: 0.245 milliseconds, Average signing power consumption: 12.09 watts, Signing Energy Consumption: 0.245 milliseconds * 12.09 watts = 2.962 Millijoules, Total Verifying Execution Time: 0.073 milliseconds, Average verifying Power Consumption: 11.56 watts, Verifying Energy Consumption : 0.073 ×11.56 = 8.438 millijoules*

MPPK/DS: The MPPK/DS scheme presents a streamlined approach to signature sizes, with different levels showcasing varying degrees of compactness. Specifically, for level I, the signature size stands at 80 bytes, expanding to 120 bytes for level III, and further to 160 bytes for level V. This tiered structure allows for flexibility in accommodating different security requirements. Beyond signature sizes, the MPPK/DS scheme distinguishes itself by offering probabilistic procedures for both signing and verification.*Total signing execution time: 0.312 milliseconds, Average signing power consumption: 15.01 watts, Signing Energy Consumption: 0.312 milliseconds * 15.01 watts = 4.683 Millijoules, Total verifying Execution Time: 0.367 milliseconds, Average verifying Power Consumption: 14.16 watts, Verifying Energy Consumption:0.367 ×14.16 = 5.196 millijoules,*

PICNIC: At NIST's L1 security level, Picnic demonstrates its efficiency with impressive sign and verify times. On a 3.6GHz workstation, signing operations take approximately 5 milliseconds, while verification processes are completed in about 4 milliseconds. However, this study analysis will compare the same for M1 microprocessor. These rapid execution times underline Picnic's suitability for time-sensitive applications where quick cryptographic operations are crucial. Moreover, in terms of signature sizes, Picnic maintains a balance between security and data efficiency. At this security level, signatures occupy 12.6KB of space, showcasing a compact representation while ensuring robust security. *Total signing execution time: 2.3 milliseconds Average signing power consumption: 12.42 watts , Signing Energy Consumption: 2.3 Milliseconds * 12.42 Watts= 28.56 Millijoules, Total verifying Execution Time: 2.548 milliseconds, Average Verifying Power Consumption: 13.51 watts, Verifying Energy Consumption: 2.54 ×13.51 = 34.31 millijoules*

Crystals-Dilithium: Crystals-Dilithium introduces distinct signature sizes across its variants, reflecting its versatility in adapting to varying security requirements. Dilithium 2 boasts a signature size of 2,479 bytes (equivalent to 19,832 bits), making it suitable for scenarios where a balance between security and efficiency is sought. Stepping up in security, Dilithium 3 increases the signature size to 3,352 bytes (or 26,816 bits), catering to more stringent security needs. For the highest security level, Dilithium 5 offers a signature size of 4,654 bytes (amounting to 37,232 bits), ensuring maximum robustness against potential attacks.*Total signing execution time: 0.298 milliseconds, Average signing power consumption: 15.01 watts, Signing Energy Consumption: 0.298 Milliseconds * 15.01 Watts = 4.472 Millijoules, Total verifying Execution Time: 0.045 milliseconds, Average verifying Power Consumption: 14.26 watts, Verifying Energy Consumption:0.0454 × 14.26 = 0.647 millijoules*

Rainbow: Rainbow's distinctive feature lies in its signature size, which is remarkably compact at just 66 bytes. This small

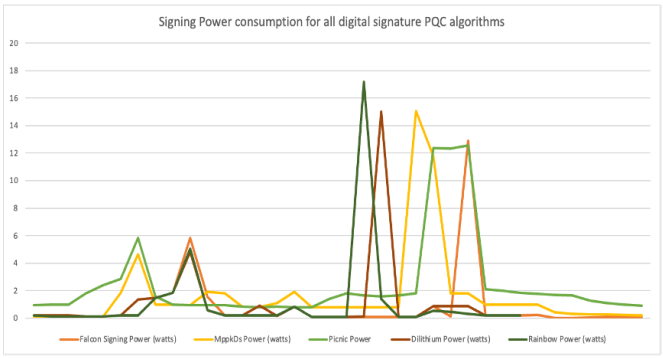

Fig. 5: Power consumption comparison for signing algorithm of different PQC digital signatures

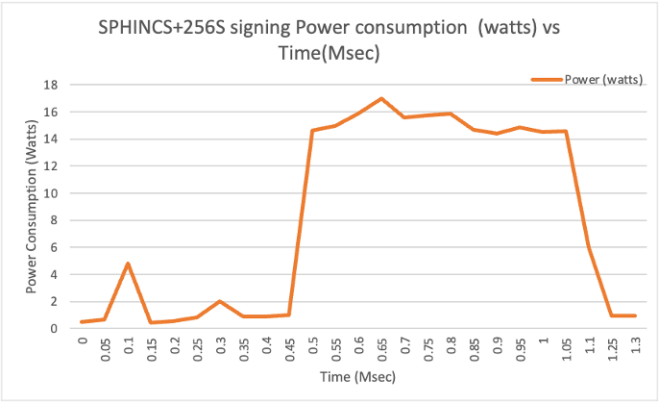

Fig. 6: SPHINCS+256 S Signing Power Consumption

signature size underscores Rainbow's commitment to efficient data representation and transmission. Despite the larger key sizes associated with the algorithm – a public key size of 161,600 bytes and a private key size of 103,648 bytes – the minimal signature size showcases Rainbow's optimization for efficient communication and storage. This balance between key sizes and signature size positions Rainbow as an intriguing solution for scenarios where minimizing signature overhead is a critical consideration, making it a viable choice for applications that require efficient and secure digital signatures.*Total signing execution time: 0.298 milliseconds, Average signing power consumption: 17.02 watts, Signing Energy Consumption: 0.298 Milliseconds * 17.02 watts = 5.071 Millijoules, Total verifying Execution Time: 0.089 milliseconds, Average verifying Power Consumption: 12.85 watts, Verifying Energy Consumption:0.089 × 12.85 = 1.143 millijoules*

SPHINCS+256S: The signature size of 29,792 bytes underscores the algorithm's commitment to providing a comprehensive and secure representation of the digital signature. This larger signature size aligns with SPHINCS+256S's emphasis on achieving a high level of cryptographic strength, making it a suitable choice for applications that prioritize stringent security requirements. The signature algorithm's power consumption is given Figure 6. *Total signing execution time: 1056 milliseconds, Average signing power consumption: 14.51 watts, Signing Energy Consumption: 1056 Milliseconds * 14.51 watts = 15322.56, Millijoules, Total verifying Execution Time: 2.067 milliseconds, Average verifying Power Consumption: 13.87 watts, verifying Energy Consumption: 2.067 × 13.87 = 28.66 millijoules* **Comparison of all the above Signing algorithms:** Comparing the energy consumption of various digital signature algorithms during signing operations reveals insights into their efficiency and suitability for diverse applications. The power consumption of a signing algorithm is a crucial metric for assessing overall effectiveness. Falcon's signing process, with an execution time of 0.245 milliseconds and an average power consumption of 12.09 watts, demonstrates efficient energy usage, resulting in approximately 2.962 millijoules. Crystals-Dilithium also shows notable energy efficiency, with a signing time of 0.298 milliseconds and average power consumption of 15.01 watts, resulting in about 4.472 millijoules. Picnic, with a longer signing time of 2.3 milliseconds, consumes around 28.56 millijoules. MPPK/DS exhibits competitive energy efficiency, with a signing time of 0.312 milliseconds and average power consumption of 15.01 watts, resulting in about 4.683 millijoules. In contrast, SPHINCS+256S, due to its extended signing time of 1056 milliseconds and average power consumption of 14.51 watts, has higher energy consumption totaling about 15322.56 millijoules. Rainbow, with a signing time of 0.298 milliseconds and average power consumption of 17.02 watts, results in an energy consumption of approximately 5.071 millijoules. In summary, Falcon and Crystals-Dilithium are the most energy-efficient signing algorithms, with consumptions of 2.962 and 4.472 millijoules, highlighting their practical viability. Picnic, MPPK/DS, and Rainbow exhibit moderate energy usage, while SPHINCS+256S's extended signing time leads to higher energy consumption. See figure 5 for the same comparison, excluding SPHINCS+256S due to its notably high power consumption during signing.

Comparison of all the above verifying algorithms: Falcon's verification takes 0.073 milliseconds, consuming 8.438 millijoules, and MPPK/DS has a verification time of 0.367 milliseconds with an energy consumption of 5.196 millijoules. Picnic, despite longer verification (2.548 ms), uses 34.31 millijoules. Crystals-Dilithium verifies in 0.045 milliseconds, consuming 0.647 millijoules, while SPHINCS+256S, with a verification time of 2.067 milliseconds, consumes about 28.66 millijoules. Rainbow's verification time is 0.089 milliseconds, using 1.143 millijoules. In summary, Crystals-Dilithium and Rainbow are the most energy-efficient, showcasing consumptions of 0.647 and 1.143 millijoules, respectively. Falcon and MPPK/DS are competitive, while Picnic and SPHINCS+256S have moderate energy consumption. This analysis aids in selecting suitable verification algorithms based on energy considerations. The comparisons are further visualized in the Table-II below, depicting the energy consumption trends across the different algorithms for each process. These analyses collectively aid in algorithm selection based on specific performance criteria. As applications increasingly prioritize energy

Digital Signature Algorithms	Key-generation Energy Consumption (Millijoules)	Signing Energy Consumption (Millijoules)	Verifying Energy Consumption (Millijoules)
Falcon	138.1536	2.962	8.438
MPPK/DS	7.4390	4.683	5.196
Picnic	4.680	28.56	34.31
Crystals-Dilithium	36.260	4.472	0.647
Sphincs+256 S	3702.560	15322.56	28.66
Rainbow	5407.785	5.071	1.143

TABLE II: Energy Consumption for Key-generation Algorithm of different PQC digital Signature Schemes

efficiency, this study's findings offer valuable guidance for choosing the optimal digital signature algorithm for various use cases.

V. Conclusion:

In conclusion, the comprehensive analysis of energy consumption across key generation, signing, and verification processes for various digital signature algorithms provides valuable insights into their performance and suitability for different applications. For key generation, Picnic emerges as the most efficient algorithm, demonstrating the shortest execution time of 0.424 milliseconds and the lowest average power consumption of 11.04 watts. MPPK/DS follows closely with competitive efficiency, showcasing a total execution time of 0.567 milliseconds and an average power consumption of 13.12 watts. Crystals-Dilithium, SPHINCS+256S, and Rainbow exhibit comparatively higher key generation times, making them more suitable for applications where speed is not the primary concern. Moving to signing algorithms, Falcon and Crystals-Dilithium stand out as the most energy-efficient choices, with energy consumptions of 2.962 millijoules and 4.472 millijoules, respectively. Picnic and MPPK/DS display moderate energy usage, while SPHINCS+256S and Rainbow exhibit relatively higher energy consumption due to their longer execution times. In the context of verification, Crystals-Dilithium and Rainbow shine as the most energy-efficient options, with energy consumptions of 0.647 millijoules and 1.143 millijoules, respectively. Falcon and MPPK/DS show competitive energy efficiency, while Picnic and SPHINCS+256S exhibit moderate energy consumption during verification operations.

VI. Future Work:

In future research, we're enhancing the security of the MPPK/DS algorithm for NIST standardization, developing a modified version (MPPK/DS T) to address vulnerabilities and prevent forging attacks. Our efforts extend to analyzing all NIST-selected algorithms, offering insights into optimal digital signature behavior. We plan to broaden the scope to various hardware platforms, particularly microcontrollers in IoT contexts, refining insights into energy efficiency. The study's future trajectory includes expanding to encryption, key encapsulation, and additional digital signature algorithms. Through energy consumption tests, we aim to comprehensively evaluate the suitability and performance of cryptographic methods, contributing to the advancement of secure communication protocols in the digital realm.

Acknowledgment

Authors would like to thank Prof. Jun Steed Huang from Carleton University, Mohammed Abuibaid, and the anonymous reviewers for their profound feedback and suggestions to improve the quality of this work.

References

[1] R. Kuang, M. Perepechaenko, and M. Barbeau, "A new quantum-safe multivariate polynomial public key digital signature algorithm, Scientific Reports, vol. 12, no. 1, p. 13168, 2022.

[2] Zhang, K., Cui, H., Yu, Y. (2022). SPHINCS-α: A Compact Stateless Hash-Based Signature Scheme. Cryptology ePrint Archive

[3] Prest, T., Fouque, P. A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., ... Zhang, Z. (2020). Falcon. Post-Quantum Cryptography Project of NIST.

[4] Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., ... Bai, S. (2020). Crystals-dilithium. Algorithm Specifications and Supporting Documentation.

[5] J. Ding, D. Schmidt, Rainbow a new multivariable polynomial signature scheme, in International Conference on Applied Cryptography and Network Security (Springer, New York, 2005), pp. 164–175

[6] G. Zaverucha, M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, J. Katz, X. Wang, V. Kolesnikov, D. Kales, Picnic. Submission to the NIST Post-Quantum Cryptography Standardization Project (2019).

[7] Lakhan, A. S., Abuibaid, M., Huang, J. S., Taha, M., Wang, Z. (2023, June). Multivariate Polynomial Public Key Digital Signature Algorithm: Semi-covariance Analysis and Performance Test over 5G Networks. In 2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 299-305). IEEE.

[8] Roma, C. A., Tai, C. E. A., Hasan, M. A. (2021). Energy Efficiency Analysis of Post-Quantum Cryptographic Algorithms. IEEE Access, 9, 71295-71317.

[9] Beckwith, L., Kaps, J. P., Gaj, K. FPGA Energy Consumption of Post-Quantum Cryptography.

[10] Dang, V. B., Farahmand, F., Andrzejczak, M., Mohajerani, K., Nguyen, D. T., Gaj, K. (2020). Implementation and benchmarking of round 2 candidates in the NIST post-quantum cryptography standardization process using hardware and software/hardware co-design approaches. Cryptology ePrint Archive: Report 2020/795.

[11] Farahmand, F., Dang, V. B., Andrzejczak, M., Gaj, K. (2019, August). Implementing and benchmarking seven round 2 lattice-based key encapsulation mechanisms using a software/hardware codesign approach. In Proceedings of the Second PQC Standardization Conference, Santa Barbara, CA, USA (pp. 22-24).

[12] Roma, C. A., Tai, C. E. A., Hasan, M. A. (2021). Energy Efficiency Analysis of Post-Quantum Cryptographic Algorithms. IEEE Access, 9, 71295-71317.