

Energy Consumption of Round 2 Submissions for NIST PQC Standards

Crystal Andrea Roma
Electrical and Computer
Engineering
University of Waterloo
caroma@uwaterloo.ca

Chi-En Amy Tai
Management
Sciences
University of Waterloo
catai@edu.uwaterloo.ca

M. Anwar Hasan
Electrical and Computer
Engineering
University of Waterloo
ahasan@uwaterloo.ca

Abstract

Classical cryptographic schemes in use today are based on the difficulty of certain number theoretic problems. Security is guaranteed by the fact that the computational work required to break the core mechanisms of these schemes on a conventional computer is infeasible; however, the difficulty of these problems would not withstand the computational power of a large-scale quantum computer. To this end, the National Institute of Standards and Technology (NIST) has begun the Post-Quantum Cryptography (PQC) standardization process. In addition to the evaluation criteria provided by NIST, the energy consumption of these candidate algorithms is also an important criterion to consider due to the increased usage of battery-operated devices and a growing interest in green computing. In this report, we profile the energy consumption of all Round 2 Optimized C Implementations and Assembly Optimized Implementations. The energy measurements are categorized based on their proposed security level and cryptographic functionality. The results are then further subdivided based on the underlying mechanism used in order to identify the most energy-efficient schemes.

1 Introduction

In today's digital systems, public-key cryptographic techniques are vital in achieving security goals such as confidentiality, data origin authentication, and data integrity. This is made possible by the difficulty of the underlying mathematical relations which make it computationally infeasible to determine one's private key from their public key. Most cryptosystems today rely on problems such as integer factorization and the discrete log problem, two computationally complex problems classical computers cannot efficiently solve. With the increased research on quantum computing, it is possible that a large-scale quantum computer may be realized. Under the quantum paradigm, many mathematical problems which were once deemed intractable may be easily solved. With this, the public-key infrastructure which we so heavily rely upon today will become obsolete. In order to avoid such a catastrophic breach of security, NIST has launched the Post-Quantum Cryptography standardization project [1]. Their aim is to begin the process of developing new quantum-resistant standards similar to the classical digital signature and key establishment schemes published in Federal Information Processing Standards Publication (FIPS) 186 and NIST Special Publications (SP) 800-56 A and B, respectively [2].

The project is currently in the process of evaluating the Round 2 submissions, all of which are publicly available on NIST’s webpage [3]. All Round 2 candidate algorithms are to be evaluated based on a variety of metrics, including correctness, speed, and size of keys, ciphertexts and signatures [4]. Although not an official criterion for evaluation, the energy consumed by each candidate submission is also an important metric to consider. This is due in part by the prevalence of mobile and other battery operated devices. Further, energy consumption can be just as important in non-battery operated devices when considering the idea of green computing, an issue gaining more traction in the research community. This movement strives to achieve more environmentally-friendly IT by emphasizing the importance of energy-efficiency from both a software and hardware standpoint.

Following the work completed in [5] in which the NIST PQC Round 1 submissions’ energy was studied, this report aims to categorize the Round 2 PQC submissions based on their energy consumption. All candidate algorithms are profiled using the IgProf lightweight profiler [6]. The Optimized C Implementations are built on a 64 bit Intel Xeon E3-1270 CPU @ 3.40GHz while the Assembly Optimized Implementations are built on a 64 bit Intel Core i7-6700 CPU @ 3.40GHz both having 8GB of RAM and running Ubuntu 16.04 LTS. The results are subdivided by their proposed security level and cryptographic functionality.

This report is organized as follows. Section 2 gives a brief summary of the Round 2 submissions as well as the NIST PQC submission requirements. Section 3 describes the method by which the energy consumption of each candidate is captured. The energy profiling results of the Optimized C Implementations are given in Section 4 while those of the Assembly Optimized Implementations are provided in Section 5. Section 6 provides an analysis of these findings. Lastly, concluding remarks are provided in Section 7.

2 NIST PQC Round 2

2.1 Summary of NIST PQC Standardization Project

NIST’s PQC Standardization Process officially became public on December 20, 2016 with a Federal Register Notice which announced a formal Call for Proposals for post-quantum cryptographic algorithms [1]. Round 1 submissions were accepted until November 30, 2017. After an initial screening by NIST, 69 submissions were made publicly available. Based on comments by the greater cryptographic community, security proofs, performance, and other evaluation criteria, 26 of the initial 69 Round 1 submissions were selected as Round 2 candidates in January of 2019. The candidates were given until April 2019 to make adjustments to their initial Round 1 submissions. These updated Round 2 submission packages have now been made available here [3]. The 26 contenders target either key encapsulation, public-key encryption, or digital signature operations (or a combination thereof). Each of these cryptographic functions requires a triple of algorithms. NIST has provided an API specification to ensure that the prototypes of these subroutines are consistent across all submissions [7].

Key encapsulation mechanisms (KEM) are primarily concerned with hybrid encryption. Generally, symmetric key encryption is more efficient than its asymmetric counterparts; however, the two parties must have a way of establishing their shared secret beforehand. KEMs provide a means by which this can be accomplished. There are three main subroutines in each proposed KEM:

a) `crypto_kem_keypair(unsigned char *pk, unsigned char *sk)` produces a public key, `pk`, and a corresponding secret key, `sk`.

- b) `crypto_kem_enc(unsigned char *ct, unsigned char *ss, const unsigned char *pk)` takes the public key, `pk`, as input, produces a shared secret, `ss`, and a ciphertext of that shared secret, `ct`.
- c) `crypto_kem_dec(unsigned char *ss, const unsigned char *ct, unsigned char *sk)` takes the ciphertext, `ct` and secret key, `sk`, as input to reproduce the shared secret, `ss`, as output.

Public-key encryption (PKE), as the name suggests, is concerned with encryption of data. It is also given by three subroutines:

- a) `crypto_enc_keypair(unsigned char *pk, unsigned char *sk)` produces a public key, `pk`, and a private key, `sk`.
- b) `crypto_encrypt(unsigned char *c, unsigned long long *clen, const unsigned char *m, unsigned long long mlen, const unsigned char *pk)` performs encryption by taking the public key, `pk`, a message `m`, as well as its length in bytes, `mlen`, as input and produces a ciphertext, `c`, of length `clen`.
- c) `crypto_encrypt_open(unsigned char *m, unsigned long long *mlen, const unsigned char *c, unsigned long long clen, const unsigned char *sk)` achieves decryption by taking the ciphertext, `ct`, its length, `clen`, and the secret key, `sk`, as input to reproduce the message, `m`, of length `mlen`.

Digital Signature algorithms provide a method by which data's origin can be authenticated. They comprise three main functions:

- a) `crypto_sign_keypair(unsigned char *pk, unsigned char *sk)` produces a public key, `pk`, and a private key, `sk`.
- b) `crypto_sign(unsigned char *sm, unsigned long long *smlen, const unsigned char *m, unsigned long long mlen, const unsigned char *sk)` creates a signature by taking the secret key, `sk`, a message `m`, as well as its length in bytes, `mlen`, as input and produces a signed message, `sm`, of length `smlen`.
- c) `crypto_sign_open(unsigned char *m, unsigned long long *mlen, const unsigned char *sm, unsigned long long smlen, const unsigned char *pk)` is a routine which verifies the signed message by taking the signed message, `sm`, its length, `smlen`, and the public key, `pk`, as input and produces the original message, `m`, of length `mlen`.

In order to comply with NIST's guidelines, any of these three cryptographic mechanisms must be demonstrated by a minimum of two implementations: a) a reference implementation for algorithm comprehension and b) an optimized implementation to demonstrate performance [4]. Both must be written in portable ANSI C. In addition to the aforementioned implementations, submissions must also provide supporting documentation which includes a written specification, performance analysis, an analysis of the algorithm against known attacks, the advantages or limitations of the algorithm, and expected security strength description. On the matter of security strength, NIST has defined 5 security levels to which submitted algorithms should adhere, which are listed below. It is not necessary that candidate algorithms target all security levels. In fact, NIST has asked submitters to focus primarily on levels 1-3, while levels 4-5 would be considered for high security applications [8].

- a) Level 1: Algorithm is at least as hard to break as AES128.
- b) Level 2: Algorithm is at least as hard to break as SHA256.
- c) Level 3: Algorithm is at least as hard to break as AES192.
- d) Level 4: Algorithm is at least as hard to break as SHA384.
- e) Level 5: Algorithm is at least as hard to break as AES256.

2.2 Round 2 Submissions

After the Round 2 candidates were selected and announced, the submitters were given time to make tweaks to their submissions. Each Round 2 package contains a summary of these changes. Many submissions have proposed new parameter sets based on less conservative approaches which promote more efficient execution, others have introduced changes to mitigate implementation attacks, as well as other changes specific to each algorithm specification. Further, some have added extra implementations to better showcase their proposal’s performance, like providing implementations using vectorized assembly instructions and versions which use hardware-accelerated symmetric primitives. For complete details regarding these changes and full algorithm specifications, we refer the reader to [3].

Table 1: Categorization of **Key Encapsulation** / **Public-key Encryption** schemes based on the mathematics of the cryptosystem

Scheme	Lattice	Code	Rank	Isogeny
BIKE [9]		✓		
Classic McEliece [10]		✓		
CRYSTALS-Kyber [11]	✓			
FrodoKEM [12]	✓			
HQC [13]		✓		
LAC [14]	✓			
LEDAcrypt [15]		✓		
NewHope [16]	✓			
NTRU [17]	✓			
NTRU Prime [18]	✓			
NTS-KEM [19]		✓		
ROLLO [20]			✓	
Round5 [21]	✓			
RQC [22]			✓	
SABER [23]	✓			
SIKE [24]				✓
Three Bears [25]	✓			

Of the 17 KEM candidates, only LAC and LEDAcrypt have also provided public-key encryption schemes for consideration. Round 2 also includes 9 digital signature algorithms. In Tables 1 and 2, we categorize these submissions based on the post-quantum cryptographic technique on which they have based their algorithm. Tables 3 and 4 show which additional implementations each submission has chosen to include. *SIMD* denotes whether the package includes an implementation which makes use of x86 SIMD instructions, particularly any of the Streaming SIMD Extensions (SSE, SSE2, SSE3 or SSE4) or Advanced Vector Extensions (AVX, AVX2, or AVX-512). *AES-NI* denotes whether extra submissions have made use of the x86 Advanced Encryption Standard instruction set. *Other* indicates whether a submission has used other assembly optimizations not already listed (for instance, those targeting multiplication). *ARM* is indicative of the submission including an implementation targeting the ARM architecture. *FPGA* and *ASIC* identify that the authors have included either of these hardware-based implementations.

This is merely a broad generalization of the additional implementations included in the submission

Table 2: Categorization of **Digital Signature** schemes based on the mathematics of the cryptosystem

Scheme	Lattice	Multi-variate	Hash	Other
CRYSTALS-Dilithium [26]	✓			
Falcon [27]	✓			
GeMSS [28]		✓		
LUOV [29]		✓		
MQDSS [30]		✓		
Picnic [31]				✓
qTESLA [32]	✓			
Rainbow [33]		✓		
SPHINCS+ [34]			✓	

packages. Most candidates have a number of different variants and parameter sets of their algorithms. In many cases, the assembly optimizations may only apply to a subset of these varieties. We refer the reader to each submission’s respective supportive documentation for more information on the specific optimizations applied to each algorithm variant.

Table 3: Additional implementations submitted in Round 2 packages of **Key Encapsulation Mechanisms/ Public-key Encryption**

Schemes	x86 Assembly Optimizations			Other Hardware		
	SIMD	AES-NI	Other	ARM	FPGA	ASIC
BIKE	✓	✓	✓		✓	
Classic McEliece	✓					
CRYSTALS-Kyber	✓	✓				
FRODO	✓	✓		✓		
hqc	✓					
LAC	✓					
LEDAcrypt	✓					
NewHope	✓					
NTRU-HPS	✓					
NTRU Prime						
NTS-KEM	✓		✓			
ROLLO						
Round5	✓					
RQC						
SABER	✓	✓				
SIKE			✓	✓	✓	✓
Three Bears			✓			

Table 4: Additional implemenations submitted in Round 2 packages of **Digital Signature** schemes

Schemes	x86 Assembly Optimizations			Other Hardware		
	SIMD	AES-NI	Other	ARM	FPGA	ASIC
CRYSTALS-Dilithium	✓	✓				
Falcon						
GeMSS	✓		✓			
Luov	✓					
MQDSS	✓					
Picnic	✓			✓		
qTESLA	✓					
Rainbow	✓					
SPHINCS+	✓	✓				

3 Methodology

It is required that a basic portable C implementation be included in each submitter’s package. In order to make a reasonable comparison, we have first chosen to profile these implementations as this would give a fair baseline comparison of all submitted schemes. We refer to this as the *Optimized C Implementation*. Further, some submissions have chosen to provide an implementation applying assembly instructions which target more modern processors. We will refer to these as the *Assembly Optimized Implementations*. As mentioned in the previous sections, there are 5 different security levels for which NIST has solicited algorithms. As it was not required that submitters target all levels, we have grouped the results by security level.

When profiling for public-key encryption schemes and digital signature submissions, the results will change based on the message that is being signed or encrypted. We have thus created a text file containing 100 randomly selected messages to be used by all public-key encryption and digital signature schemes to be tested in order to make a fair comparison. In the case of public-key encryption, we have selected all messages to be 32 bytes in length, which was nearly the largest message length specified in the Known-Answer Test script for public-key encryption schemes provided by NIST. Similarly, all messages to be signed are 3300 bytes in length; this is the largest message length specified in NIST’s Known-Answer Test script for digital signature schemes.

Many submissions have also chosen to include different variants of their algorithms to be considered for submission. We have chosen to profile all of those which have a corresponding security level. This may include variants with different security goals, those with ephemeral keys, those using different hashing functions, and those who have varied sizes of certain internal parameters, to name a few. When reporting results, each submission name specifies the specific implementation tested, following the submitters’ naming conventions. For details about the differences between these variants, we refer the reader to the supporting documentation provided in each candidate’s submission package. In cases where the proposed algorithm includes multiple submissions targeting the same security level, we have reported results for the lowest energy consuming variant.

The profiling of the Optimized C Implementations has been performed on a 64-bit processor Intel Xeon

E3-1270 CPU @ 3.40GHz with 8GB of RAM running Ubuntu 16.04 LTS. As this processor does not support AVX2 instructions, we have chosen to profile the Assembly Optimized Implementations on a 64-bit processor Intel Core i7-6700 CPU @ 3.40GHz with 8GB of RAM running Ubuntu 16.04 LTS. In order to obtain the energy measurements, we have used IgProf [6]. IgProf is an energy profiling tool developed by CERN. It works on the basis of statistical sampling, using PAPI to obtain measurements from the Running Average Power Limit at a fixed interval and attributes the current energy measurement to the current location of execution of the code [35]. In this way, we are able to obtain an estimate of the amount of energy consumed by specific functions within a running program rather than just the energy consumed by the entire process itself. The output of the profiler is composed of a flat cumulative profile, a flat self profile, and a call graph profile. As a result, it is not only possible to determine how much energy is exhausted to run a function, but it will also highlight the amount of energy exhausted by all of its child functions. In this way, one can pinpoint the energy hotspots; this can be helpful in determining which subroutines are the least energy-efficient which can aid in efforts to optimize the code. We have performed our profiling tests for a minimum of 100 iterations. IgProf samples the model specific registers at a rate of 5 milliseconds. Based on the results of [5], we expect many of the candidate algorithms to execute much faster than the sampling rate of the tool. In these cases, we increase the number of iterations of the algorithm we perform. We calculate the power based on the average energy and time.

In the proceeding two sections, the results of the energy profiling experiments are provided. All timing results are reported in milliseconds and all energy measurements in milliJoules. Power is recorded in Watts. In each security level, the most and least energy-consuming functions are emphasized in **bold** font.

4 Energy Consumption Results of Optimized C Implementations

4.1 Key Encapsulation Mechanisms

Table 5: Energy consumption of `crypto_kem_keypair` function for **Keypair Generation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1 CCA	26.79	0.21	5.63	-	-	-	27.4	0.64	17.62	-	-	-	27.96	1.41	39.37
BIKE1 CPA	26.84	0.19	5.02	-	-	-	28.31	0.48	13.62	-	-	-	27.78	0.87	24.25
BIKE2 CCA	26.57	0.45	11.85	-	-	-	25.76	1.49	38.29	-	-	-	26.87	3.2	85.95
BIKE2 CPA	27.29	0.39	10.51	-	-	-	27.22	1.06	28.85	-	-	-	25.29	1.95	49.29
BIKE3 CCA	28.89	0.11	3.29	-	-	-	28.15	0.38	10.67	-	-	-	26.91	0.84	22.66
BIKE3 CPA	28.87	0.11	3.15	-	-	-	27.37	0.29	8.05	-	-	-	28.61	0.64	18.28
Classic McEliece	26.21	243.6	6384.90	-	-	-	26.15	444.9	11632.4	-	-	-	27.55	1388	38234.6
CRYSTALS-Kyber	25.64	0.16	4.15	-	-	-	25.52	0.28	7.2	-	-	-	25.66	0.44	11.29
CRYSTALS-Kyber-90s	26.09	0.26	6.86	-	-	-	25.73	0.49	12.58	-	-	-	26.66	0.79	21.03
FRODO AES	25.05	0.41	10.27	-	-	-	24.75	1.22	30.2	-	-	-	23.01	1.22	28.07
FRODO SHAKE	22.29	0.42	9.36	-	-	-	24.15	1.21	29.22	-	-	-	24.16	1.18	28.51
hqc-1	25.76	0.34	8.76	-	-	-	27.54	0.85	23.41	-	-	-	26.19	1.36	35.62
hqc-2	-	-	-	-	-	-	28.53	0.9	25.68	-	-	-	27.39	1.6	43.83
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	27.98	1.78	49.8
LAC	25.91	0.03	0.88	-	-	-	25.06	0.11	2.63	-	-	-	26.37	0.11	2.8
LEDAcrypt N02	31.81	12.8	407.2	-	-	-	23.44	36.70	860.3	-	-	-	24.34	73.90	1799
LEDAcrypt N03	22.89	4.5	103	-	-	-	24.86	15.6	387.8	-	-	-	23.72	43.2	1024.8
LEDAcrypt N04	23.27	4.40	102.4	-	-	-	23.47	14.2	333.3	-	-	-	24.25	31.8	771
LEDAcrypt LT DFR64	25.07	373.40	9359.50	-	-	-	24.83	1212.50	30104.60	-	-	-	24.73	3899.20	96421.70
LEDAcrypt LT DFRSL	24.21	623.20	15088.90	-	-	-	25.63	2148.60	55077.80	-	-	-	24.27	6403.60	155391.90
NewHope CCA	26.2	0.21	5.42	-	-	-	-	-	-	-	-	-	26.05	0.41	10.76
NewHope CPA	26.01	0.19	4.99	-	-	-	-	-	-	-	-	-	25.36	0.39	9.94
NTRU-HPS	24.45	40.47	989.52	-	-	-	25.36	70.35	1784.32	-	-	-	24.93	107.24	2673.23
NTRU-HRSS	-	-	-	-	-	-	25.33	75.87	1922.05	-	-	-	-	-	-
sNTRU Prime	-	-	-	23.66	67.64	1600.31	22.96	91.6	2102.77	22.8	115.31	2628.83	-	-	-
NTRU LPrime	-	-	-	23.92	7.64	182.74	24.08	10.09	242.94	24.69	12.71	313.77	-	-	-
NTS-KEM	27.31	19.601	535.30	-	-	-	30.2	61.7	1863.4	-	-	-	26.97	107	2886.2
ROLLO-I	18.59	0.66	12.27	-	-	-	23.8	1.10	26.18	-	-	-	25.06	1.27	31.83
ROLLO-II	25.59	4.54	116.18	-	-	-	25.79	4.95	127.65	-	-	-	24.96	5.05	126.03
ROLLO-III	26.54	0.13	3.45	-	-	-	26.78	0.18	4.82	-	-	-	25	0.27	6.75
Round5 Ring	28.65	0.02	0.57	-	-	-	24.34	0.08	1.87	-	-	-	25.33	0.09	2.36
Round5 Ring 5	25.77	0.03	0.67	-	-	-	26.25	0.05	1.26	-	-	-	26.33	0.09	2.29
Round5 Non-Ring	25.14	2.1	52.83	-	-	-	26.01	5.15	133.9	-	-	-	24.58	9.63	236.69
Round5 Long Key	26.93	0.03	0.73	-	-	-	-	-	-	-	-	-	-	-	-
RQC	27.39	0.23	6.3	-	-	-	25.02	0.42	10.51	-	-	-	25.12	0.67	16.83
SABER	27.5	0.04	1.10	-	-	-	22.56	0.09	2.03	-	-	-	27.87	0.15	4.18
SIKE	24.42	23.5	573.80	24.58	36	884.7	25.12	67	1682.9	-	-	-	23.6	121.2	2859.9
SIKE Compressed	24.86	58.8	1461.9	23.89	88	2102.4	23.97	168.3	4034.5	-	-	-	23.71	284.7	6750.8
Three Bears	-	-	-	26.07	0.03	0.73	-	-	-	25.21	0.06	1.41	26.12	0.09	2.35
Three Bears Eph.	-	-	-	25.34	0.03	0.74	-	-	-	25.84	0.06	1.45	29	0.09	2.55

Table 6: Energy consumption of `crypto_kem_enc` function for **Encapsulation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1 CCA	25.75	0.27	6.83	-	-	-	24.21	0.84	20.21	-	-	-	27.79	1.75	48.65
BIKE1 CPA	27.23	0.22	5.94	-	-	-	24.52	0.62	15.2	-	-	-	25.46	1.04	26.53
BIKE2 CCA	22.35	0.12	2.7	-	-	-	26.91	0.33	8.75	-	-	-	30.16	0.71	21.54
BIKE2 CPA	21.04	0.11	2.36	-	-	-	23.15	0.25	5.86	-	-	-	25.13	0.44	11.06
BIKE3 CCA	24.75	0.22	5.55	-	-	-	24.83	0.76	18.82	-	-	-	28.38	1.61	45.7
BIKE3 CPA	24.26	0.2	4.85	-	-	-	23.86	0.58	13.74	-	-	-	28.79	1.21	34.70
Classic McEliece	20.44	0.09	1.84	-	-	-	28.09	0.11	3.09	-	-	-	25.32	0.19	4.81
CRYSTALS-Kyber	27.25	0.22	6	-	-	-	25.64	0.36	9.13	-	-	-	23.78	0.62	14.62
CRYSTALS-Kyber-90s	23.29	0.33	7.62	-	-	-	25.35	0.55	13.94	-	-	-	23.09	0.91	21.1
FRODO AES	25.36	0.58	14.71	-	-	-	27.06	1.19	32.20	-	-	-	29.24	1.64	47.96
FRODO SHAKE	21.83	0.64	13.97	-	-	-	24.51	1.26	30.88	-	-	-	27.73	1.75	48.53
hqc-1	26.7	0.69	18.42	-	-	-	25.55	1.68	42.92	-	-	-	24.3	2.71	65.85
hqc-2	-	-	-	-	-	-	24.31	1.72	41.81	-	-	-	22.91	3.34	76.53
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	24.87	3.52	87.53
LAC	22.03	0.06	1.28	-	-	-	24.21	0.15	3.53	-	-	-	23.49	0.2	4.58
LEDAcrypt N02	24.44	0.82	20.14	-	-	-	24.74	1.9	47	-	-	-	25.59	3.2	81.90
LEDAcrypt N03	24.1	0.66	15.9	-	-	-	29.31	1.3	38.1	-	-	-	26.67	3.3	88
LEDAcrypt N04	24.73	0.84	20.87	-	-	-	22.59	2.20	49.7	-	-	-	24.85	4.10	101.9
LEDAcrypt LT DFR64	20.74	3.40	70.50	-	-	-	26.42	5.20	137.40	-	-	-	25.43	9.20	234.00
LEDAcrypt LT DFRSL	22.21	5.20	115.50	-	-	-	25.30	11.50	291.00	-	-	-	22.69	23.20	526.30
NewHope CCA	24.82	0.33	8.19	-	-	-	-	-	-	-	-	-	25.89	0.64	16.57
NewHope CPA	26.22	0.29	7.47	-	-	-	-	-	-	-	-	-	22.43	0.56	12.56
NTRU-HPS	27.09	0.82	22.21	-	-	-	24.23	1.5	36.35	-	-	-	21.07	2.28	48.03
NTRU-HRSS	-	-	-	-	-	-	24.34	1.38	33.59	-	-	-	-	-	-
sNTRU Prime	-	-	-	24.05	7.38	177.5	24.06	9.85	236.95	23.74	12.65	300.36	-	-	-
NTRU LPrime	-	-	-	22.2	14.89	330.6	23.17	19.46	450.95	24.12	24.32	586.52	-	-	-
NTS-KEM	25.53	0.04	0.92	-	-	-	28.41	0.13	3.55	-	-	-	28.75	0.16	4.66
ROLLO-I	22.69	0.16	3.63	-	-	-	27.42	0.19	5.21	-	-	-	25.46	0.26	6.62
ROLLO-II	24.62	0.76	18.71	-	-	-	29.48	0.85	25.06	-	-	-	21.52	1.06	22.81
ROLLO-III	28.67	0.3	8.6	-	-	-	26.95	0.38	10.24	-	-	-	25.43	0.63	16.02
Round5 Ring	25.08	0.04	0.93	-	-	-	22.75	0.14	3.09	-	-	-	25.52	0.15	3.88
Round5 Ring 5	22.84	0.05	1.14	-	-	-	24.21	0.08	1.86	-	-	-	22.66	0.15	3.38
Round5 Non-Ring	24.82	1.91	47.47	-	-	-	25.39	4.66	118.22	-	-	-	25.71	8.64	222.17
Round5 Long Key	23.98	0.05	1.18	-	-	-	-	-	-	-	-	-	-	-	-
RQC	25.22	0.51	12.86	-	-	-	26.01	0.91	23.67	-	-	-	25.6	1.43	36.61
SABER	28.9	0.66	19.04	-	-	-	24.92	1.19	29.63	-	-	-	24.28	1.91	46.32
SIKE	24.33	38.4	934.1	24.53	58.9	1444.9	24.97	122.4	3056.1	-	-	-	23.48	195.9	4598.90
SIKE Compressed	24.64	70.90	1747.2	24.3	107.8	2619.302	23.9	201.2	4808.90	-	-	-	23.53	360.9	8492
Three Bears	-	-	-	24.53	0.04	0.98	-	-	-	22.7	0.07	1.52	27.42	0.09	2.52
Three Bears Eph.	-	-	-	21.58	0.04	0.93	-	-	-	24.04	0.08	1.83	23.13	0.1	2.41

Table 7: Energy consumption of `crypto_kem_dec` function for **Decapsulation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1 CCA	25.04	2.4	60.1	-	-	-	24.56	5.4	132.6	-	-	-	29.42	12.4	364.8
BIKE1 CPA	22.38	1.3	29.1	-	-	-	24.05	3.7	89	-	-	-	25.46	8.4	213.9
BIKE2 CCA	23.15	2	46.3	-	-	-	26	4.60	119.6	-	-	-	25.3	10.7	270.7
BIKE2 CPA	24.46	1.3	31.8	-	-	-	24.41	3.4	83	-	-	-	22.84	8.1	185
BIKE3 CCA	23.16	2.5	57.9	-	-	-	23.33	6	140	-	-	-	26.54	12.5	331.7
BIKE3 CPA	25.38	1.3	33	-	-	-	27.11	3.6	97.6	-	-	-	24.61	8.80	216.6
Classic McEliece	23.15	25.4	588.1	-	-	-	23.36	64.10	1497.5	-	-	-	22.1	118.8	2625.3
CRYSTALS-Kyber	26.9	0.26	7.05	-	-	-	24.44	0.43	10.51	-	-	-	23.78	0.62	14.62
CRYSTALS-Kyber-90s	25.32	0.37	9.47	-	-	-	25.4	0.63	16.10	-	-	-	23.72	0.98	23.25
FRODO AES	22.74	0.54	12.28	-	-	-	26.64	1.10	29.3	-	-	-	27.47	1.5	41.2
FRODO SHAKE	23.38	0.68	15.9	-	-	-	26.08	1.3	33.9	-	-	-	28.64	1.4	40.1
hqc-1	23.92	1.13	27.03	-	-	-	24.88	2.6	64.69	-	-	-	25.47	3.95	100.6
hqc-2	-	-	-	-	-	-	24.74	2.84	70.26	-	-	-	26.32	4.74	124.78
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	28.48	5.11	145.55
LAC	26.43	0.08	2.14	-	-	-	22.24	0.25	5.56	-	-	-	23.81	0.29	7
LEDAcrypt N02	30.09	3.5	105.3	-	-	-	24.2	9.30	225.1	-	-	-	24.64	15.9	391.7
LEDAcrypt N03	25.33	3.9	98.8	-	-	-	23.55	10.20	240.2	-	-	-	23.54	21.1	496.7
LEDAcrypt N04	25.6	5.8	148.5	-	-	-	24.1	14.7	354.2	-	-	-	24.74	21.8	539.4
LEDAcrypt LT DFR64	28.50	3.40	96.90	-	-	-	23.63	8.30	196.10	-	-	-	23.90	15.90	380.00
LEDAcrypt LT DFRSL	25.12	4.20	105.50	-	-	-	24.76	11.30	279.80	-	-	-	23.63	22.60	534.10
NewHope CCA	23.14	0.35	8.1	-	-	-	-	-	-	-	-	-	24.36	0.77	18.76
NewHope CPA	25.07	0.06	1.45	-	-	-	-	-	-	-	-	-	27.8	0.1	2.78
NTRU-HPS	24.82	2.19	54.36	-	-	-	26.63	3.69	98.28	-	-	-	25.05	5.47	137.03
NTRU-HRSS	-	-	-	-	-	-	24.76	4.06	100.51	-	-	-	-	-	-
sNTRU Prime	-	-	-	23.51	21.61	507.96	24.09	29.19	703.16	23.56	37.22	877	-	-	-
NTRU LPrime	-	-	-	22.43	21.55	483.38	22.93	28.68	657.76	23.55	36.18	851.96	-	-	-
NTS-KEM	24.79	0.26	6.32	-	-	-	27.03	0.47	12.76	-	-	-	26.17	1.06	27.74
ROLLO-I	25.11	0.53	13.31	-	-	-	25.87	0.97	25.09	-	-	-	25.02	1.62	40.53
ROLLO-II	22.74	2.32	52.76	-	-	-	24.37	2.65	64.58	-	-	-	24	3.14	75.37
ROLLO-III	23.83	0.53	12.63	-	-	-	23.44	1.09	25.55	-	-	-	23.83	1.68	40.03
Round5 Ring	26.27	0.02	0.39	-	-	-	25.43	0.07	1.73	-	-	-	24.92	0.09	2.24
Round5 Ring 5	29.91	0.02	0.66	-	-	-	27.39	0.04	1.04	-	-	-	26.52	0.07	1.78
Round5 Non-Ring	24.6	0.12	2.98	-	-	-	26.7	0.18	4.91	-	-	-	27.44	0.07	1.87
Round5 Long Key	24.59	0.02	0.54	-	-	-	-	-	-	-	-	-	-	-	-
RQC	25.96	2.91	75.54	-	-	-	25.61	6.88	176.17	-	-	-	25.15	10.87	273.43
SABER	24.38	0.78	19.04	-	-	-	24.85	1.34	33.22	-	-	-	23.34	2.21	51.68
SIKE	24.43	40.9	999.3	24.62	62.7	1543.4	24.98	124.3	3104.5	-	-	-	23.61	211.1	4983.7
SIKE Compressed	24.9	66.2	1648.1	23.98	100.8	2417.5	24.01	193.1	4636.90	-	-	-	23.54	334.8	7880.2
Three Bears	-	-	-	25.05	0.06	1.4	-	-	-	20.65	0.11	2.25	24.43	0.15	3.54
Three Bears Eph.	-	-	-	29.27	0.02	0.44	-	-	-	34.72	0.02	0.63	35.55	0.02	0.71

4.2 Public-key Encryption

Table 8: Energy consumption of `crypto_encrypt_keypair` function for **Keypair Generation** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	23.76	334.3	7944.2	-	-	-	26.08	1112.2	29007.1	-	-	-	25.31	3367.5	85229.9
LEDAcrypt DFRSL	24.21	493.3	11944.4	-	-	-	24.86	1709.2	42496.4	-	-	-	24.69	5150.3	127181.7
LAC	24.17	0.04	0.85	-	-	-	26.1	0.1	2.61	-	-	-	25.48	0.11	2.83

Table 9: Energy consumption of `crypto_encrypt` function for **Encryption** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	25.53	0.43	10.98	-	-	-	25.7	0.73	18.76	-	-	-	25.56	1.33	33.99
LEDAcrypt DFRSL	26.31	0.58	15.26	-	-	-	28.35	1.3	36.86	-	-	-	25.9	2.23	57.75
LAC	25.54	0.06	1.43	-	-	-	24.23	0.14	3.32	-	-	-	23.88	0.19	4.63

Table 10: Energy consumption of `crypto_encrypt_open` function for **Decryption** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	24.56	1.26	30.95	-	-	-	23.92	2.4	57.4	-	-	-	24.59	4.61	113.34
LEDAcrypt DFRSL	24.67	1.8	44.4	-	-	-	24.14	3.92	94.63	-	-	-	24.3	7.37	179.09
LAC	25.35	0.03	0.66	-	-	-	24.65	0.1	2.39	-	-	-	20.58	0.1	2.12

4.3 Digital Signature

Table 11: Energy consumption of `crypto_sign_keypair` function for **Keypair Generation** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	25.64	0.06	1.51	27.88	0.06	1.65	25.27	0.06	1.49	24.98	0.06	1.47	-	-	-
CRYSTALS-Dilithium AES	26.65	0.12	3.28	25.79	0.21	5.44	26.29	0.33	8.73	24.81	0.46	11.46	-	-	-
Falcon	23.73	8	189.8	-	-	-	-	-	-	-	-	-	24.66	21.8	537.6
GeMSS	25.56	21.6	552.20	-	-	-	25.88	125.8	3255.2	-	-	-	26.25	371	9738.2
BlueGeMSS	26.17	20.6	539.1	-	-	-	26.17	118.7	3106.3	-	-	-	25.68	353.4	9076.6
RedGeMSS	25.74	19.90	512.20	-	-	-	26.25	106.6	2798.6	-	-	-	26.36	323.2	8520.70
Luov Small Sig Chacha	-	-	-	22.51	4.7	105.8	-	-	-	22.92	18	412.5	23.73	31.4	745.2
Luov Small Sig Keccak	-	-	-	22.64	5.6	126.8	-	-	-	24.97	19.90	496.9	23.26	35.6	828.2
Luov Large Sig Chacha	-	-	-	23.9	2.9	69.3	-	-	-	24.04	7.8	187.5	24.64	18.5	455.8
Luov Large Sig Keccak	-	-	-	25.12	3.3	82.9	-	-	-	24.82	8.80	218.4	25.11	21.2	532.30
MQDSS	-	-	-	27.36	1.01	27.63	-	-	-	26.18	2.20	57.6	-	-	-
Picnic UR	24.7	0.01	0.23	-	-	-	24.78	0.01	0.33	-	-	-	25.67	0.02	0.45
Picnic FS	25.91	0.01	0.24	-	-	-	24.64	0.01	0.33	-	-	-	25.23	0.02	0.44
Picnic2 FS	23.83	0.01	0.22	-	-	-	25.01	0.01	0.34	-	-	-	25.72	0.02	0.45
qTESLA	25.1	0.4	10.04	23.92	2.1	50.24	24.45	1.16	28.36	-	-	-	23.86	6.12	146.03
qTESLA-s	25.63	0.4	10.25	25.4	2.01	51.06	25.7	1.12	28.78	-	-	-	25.5	5.99	152.73
qTESLA-p	25.42	2.5	63.55	-	-	-	23.74	9.54	226.51	-	-	-	-	-	-
qTESLA-size	-	-	-	-	-	-	-	-	-	-	-	-	25.7	8.88	228.22
qTESLA-size-s	-	-	-	-	-	-	-	-	-	-	-	-	24.11	8.69	209.51
Rainbow Classic	25.79	8.9	229.5	-	-	-	25.41	132.30	3361.8	-	-	-	25.01	336.3	8411.5
Rainbow Compressed/Cyclic	25.35	10	253.5	-	-	-	24.68	150.5	3714.7	-	-	-	23.26	387.4	9010.5
Rainbow Cyclic	26.75	10.20	272.8	-	-	-	26.86	151.1	4058.5	-	-	-	24.57	379.4	9320
SPHINCS+ SHA256 s simple	27.05	65.10	1761	-	-	-	27.08	93.2	2524.20	-	-	-	28.21	124.1	3501.4
SPHINCS+ SHA256 s robust	27.71	126.9	3516.9	-	-	-	28.04	188.9	5296	-	-	-	26.51	361.6	9586.1
SPHINCS+ SHA256 f simple	25.32	1.9	48.1	-	-	-	26.16	3.1	81.10	-	-	-	26.5	8	212
SPHINCS+ SHA256 f robust	27.18	4	108.7	-	-	-	25.73	5.9	151.801	-	-	-	28.62	21.5	615.4
SPHINCS+ SHAKE256 s simple	28.2	107.1	3019.9	-	-	-	25.66	157.6	4044.1	-	-	-	26.08	208.8	5446.2
SPHINCS+ SHAKE256 s robust	25.99	199.1	5174	-	-	-	25.91	293.60	7608.6	-	-	-	27.26	394.5	10754.8
SPHINCS+ SHAKE256 f simple	26.19	3.2	83.8	-	-	-	25.22	4.90	123.6	-	-	-	27.16	12.6	342.2
SPHINCS+ SHAKE256 f robust	27.27	6.3	171.8	-	-	-	25.58	9.30	237.9	-	-	-	26.94	24.2	652
SPHINCS+ HARAKA s simple	25.86	183.5	4745.7	25.35	266.60	6759.6	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA s robust	26.23	265.60	6965.7	26.15	399.4	10444.4	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f simple	25.45	5.65	143.80	26.2	8.20	214.8	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f robust	26.23	8.36	219.3	26.02	12.3	320.10	-	-	-	-	-	-	-	-	-

Table 12: Energy consumption of `crypto_sign` function for **Signing** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	24.08	0.33	7.92	26.47	0.33	8.63	24.52	0.33	8.09	24.09	0.33	8.02	-	-	-
CRYSTALS-Dilithium AES	23.59	0.5	11.82	24.98	0.78	19.56	24.68	1.25	30.75	25.27	1.16	29.34	-	-	-
Falcon	22.07	0.29	6.4	-	-	-	-	-	-	-	-	-	24.07	0.69	16.61
GeMSS	26.29	677.2	17800.40	-	-	-	26.12	2650.6	69223.60	-	-	-	26.34	5398.7	142192.80
BlueGeMSS	26.16	104.5	2733.5	-	-	-	26.28	397.6	10449.80	-	-	-	25.79	690.5	17805.8
RedGeMSS	22.83	2.9	66.2	-	-	-	24.78	10.3	255.2	-	-	-	27.04	18.40	497.6
Luov Small Sig Chacha	-	-	-	26.53	1.5	39.80	-	-	-	23.95	4.10	98.2	25.17	7	176.2
Luov Small Sig Keccak	-	-	-	28.91	2.30	66.5	-	-	-	28.1	6.1	171.4	25.76	10.4	267.90
Luov Large Sig Chacha	-	-	-	24.99	8.30	207.4	-	-	-	22.63	25.1	568	25.18	51.1	1286.90
Luov Large Sig Keccak	-	-	-	25.88	8.6	222.6	-	-	-	25.02	24.9	623.1	26.75	52.6	1406.9
MQDSS	-	-	-	25.35	48.1	1219.10	-	-	-	24.9	152.6	3800.2	-	-	-
Picnic UR	26.37	5.2	137.1	-	-	-	27.1	13.5	365.9	-	-	-	29.27	23	673.1
Picnic FS	23.59	4.40	103.8	-	-	-	25.26	10.4	262.7	-	-	-	25.37	18.8	477
Picnic2 FS	26.2	147.5	3865	-	-	-	26.12	436.9	11413.5	-	-	-	26.42	924.3	24415.60
qTESLA	24.28	0.21	5.12	24	0.57	13.68	25.19	0.34	8.47	-	-	-	25	0.93	23.3
qTESLA-s	24.94	0.22	5.39	24.21	0.6	14.43	24.74	0.35	8.66	-	-	-	25.87	0.94	24.21
qTESLA-p	24.48	1.26	30.77	-	-	-	24.04	3.45	82.96	-	-	-	-	-	-
qTESLA-size	-	-	-	-	-	-	-	-	-	-	-	-	24.51	1.28	31.3
qTESLA-size-s	-	-	-	-	-	-	-	-	-	-	-	-	24.3	1.37	33.17
Rainbow Classic	24.08	0.12	2.89	-	-	-	24.79	1.03	25.53	-	-	-	23.06	2.24	51.66
Rainbow Compressed/Cyclic	25.53	0.15	3.83	-	-	-	24.68	77.10	1902.7	-	-	-	23.25	192.8	4482.5
Rainbow Cyclic	29.27	0.11	3.22	-	-	-	23.57	1.05	24.75	-	-	-	24.54	2.24	54.97
SPHINCS+ SHA256 s simple	27.6	960.1	26494.9	-	-	-	27.51	2314	63668.1	-	-	-	28.42	1608.6	45710.8
SPHINCS+ SHA256 s robust	28.12	1742.9	49013.5	-	-	-	27.93	4305.7	120274.9	-	-	-	26.94	4401	118541.2
SPHINCS+ SHA256 f simple	28.43	63.3	1799.7	-	-	-	28.13	84.2	2368.80	-	-	-	27.43	190.9	5236.8
SPHINCS+ SHA256 f robust	29.56	116.6	3446.7	-	-	-	27.32	161.9	4422.90	-	-	-	28.05	514.30	14426.2
SPHINCS+ SHAKE256 s simple	27.72	1586	43964.3	-	-	-	26.09	3331.8	86931.9	-	-	-	25.82	2542.80	65658.40
SPHINCS+ SHAKE256 s robust	26.12	2807.2	73310.8	-	-	-	26.18	5689	148952	-	-	-	27.21	4476.10	121794.4
SPHINCS+ SHAKE256 f simple	28.22	103.7	2926.1	-	-	-	27.8	130.9	3638.4	-	-	-	27.93	288.2	8049.6
SPHINCS+ SHAKE256 f robust	27.99	187.8	5257.3	-	-	-	27.28	246.2	6715.3	-	-	-	27.57	535.6	14767.7
SPHINCS+ HAKA s simple	25.97	3310.3	85968.1	25.4	7281.8	184990.6	-	-	-	-	-	-	-	-	-
SPHINCS+ HAKA s robust	26.41	4937.10	130367	25.91	12386.6	320977.5	-	-	-	-	-	-	-	-	-
SPHINCS+ HAKA f simple	26.12	205.5	5367	26.19	240.1	6287.4	-	-	-	-	-	-	-	-	-
SPHINCS+ HAKA f robust	26.64	306.8	8171.7	26.23	371.4	9742.30	-	-	-	-	-	-	-	-	-

Table 13: Energy consumption of `crypto_sign_open` function for **Verification** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	25.99	0.09	2.39	21.88	0.1	2.08	28.32	0.09	2.46	26.84	0.09	2.47	-	-	-
CRYSTALS-Dilithium AES	27.22	0.14	3.73	25.69	0.22	5.7	25.67	0.33	8.58	24.68	0.49	12.04	-	-	-
Falcon	23.15	0.06	1.41	-	-	-	-	-	-	-	-	-	24.4	0.11	2.73
GeMSS	28.76	0.06	1.7	-	-	-	28.66	0.14	4.10	-	-	-	26.3	0.32	8.29
BlueGeMSS	24.97	0.06	1.6	-	-	-	27.78	0.14	3.97	-	-	-	27.97	0.3	8.48
RedGeMSS	42.3	0.11	4.48	-	-	-	38.58	0.15	5.94	-	-	-	44.55	0.32	14.08
Luov Small Sig Chacha	-	-	-	26.42	1.2	31.7	-	-	-	24.59	3.2	78.7	25.59	5.4	138.20
Luov Small Sig Keccak	-	-	-	23.05	2	46.1	-	-	-	27.62	5.3	146.4	25.68	9	231.1
Luov Large Sig Chacha	-	-	-	23.51	5.7	134	-	-	-	23.77	15.6	370.8	24.83	26.3	653.1
Luov Large Sig Keccak	-	-	-	24.19	6.2	150	-	-	-	24.77	16.3	403.7	25.51	28.4	724.5
MQDSS	-	-	-	24.89	35.9	893.4	-	-	-	25	113	2825.1	-	-	-
Picnic UR	27.88	4.3	119.9	-	-	-	28	11.1	310.8	-	-	-	26.83	20.10	539.30
Picnic FS	28.79	3.4	97.9	-	-	-	25.57	8.9	227.6	-	-	-	25.7	16.2	416.4
Picnic2 FS	27.46	67.5	1853.8	-	-	-	27.3	157.20	4290.90	-	-	-	27.96	281.2	7862.4
qTESLA	27.27	0.05	1.23	23.74	0.12	2.85	22.31	0.1	2.12	-	-	-	24.43	0.18	4.37
qTESLA-s	24.93	0.05	1.15	24.19	0.12	2.9	24.41	0.1	2.34	-	-	-	25.89	0.17	4.43
qTESLA-p	24.7	0.28	6.94	-	-	-	24.69	0.79	19.46	-	-	-	-	-	-
qTESLA-size	-	-	-	-	-	-	-	-	-	-	-	-	24.2	0.26	6.2
qTESLA-size-s	-	-	-	-	-	-	-	-	-	-	-	-	23.9	0.25	6.07
Rainbow Classic	24.96	0.1	2.37	-	-	-	25.25	1.36	34.30	-	-	-	24.02	2.32	55.79
Rainbow Compressed/Cyclic	27.16	1.16	31.45	-	-	-	26.11	7.23	188.87	-	-	-	25.55	17.10	436.9
Rainbow Cyclic	27.81	1.14	31.62	-	-	-	27.01	6.78	183	-	-	-	27.91	16.5	460.5
SPHINCS+ SHA256 s simple	27.55	1.07	29.48	-	-	-	28.19	1.71	48.2	-	-	-	28.18	2.20	62
SPHINCS+ SHA256 s robust	26.88	2.20	59.13	-	-	-	29.06	3.42	99.37	-	-	-	26.08	7.1	185.2
SPHINCS+ SHA256 f simple	27.37	2.6	71.16	-	-	-	29.06	4.18	121.46	-	-	-	24.63	4.60	113.3
SPHINCS+ SHA256 f robust	28.54	5.3	151.26	-	-	-	27.08	8.77	237.49	-	-	-	27.61	12.8	353.4
SPHINCS+ SHAKE256 s simple	27.67	1.73	47.87	-	-	-	27.79	2.54	70.58	-	-	-	28.41	3.2	90.9
SPHINCS+ SHAKE256 s robust	27.69	3.35	92.77	-	-	-	26.36	5.01	132.04	-	-	-	27.21	6.6	179.6
SPHINCS+ SHAKE256 f simple	27.25	4.22	114.99	-	-	-	27.83	6.71	186.72	-	-	-	26.09	6.6	172.2
SPHINCS+ SHAKE256 f robust	26.32	8.42	221.6	-	-	-	26.74	12.94	346.07	-	-	-	28.32	13.3	376.6
SPHINCS+ HARAKA s simple	22.97	3.9	89.6	26.64	5.5	146.5	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA s robust	22.43	6.1	136.80	26.31	8.80	231.5	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f simple	26.66	8.20	218.6	27.29	13.1	357.5	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f robust	27.11	13.1	355.1	25.67	20.8	533.9	-	-	-	-	-	-	-	-	-

5 Energy Consumption Results of Assembly Optimized Implementations

5.1 Key Encapsulation Mechanisms

Table 14: Energy consumption of `crypto_kem_keypair` function for **Keypair Generation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	3.800	0.080	0.304	-	-	-	3.519	0.262	0.922	-	-	-	3.259	0.656	2.138
BIKE2	4.975	4.530	22.535	-	-	-	4.856	16.905	82.097	-	-	-	4.885	47.180	230.485
BIKE3	3.426	0.054	0.185	-	-	-	3.539	0.167	0.591	-	-	-	3.220	0.427	1.375
Classic McEliece AVX	5.160	45.532	234.940	-	-	-	5.687	150.781	857.444	-	-	-	5.884	281.968	1659.074
Classic McEliece SSE	5.108	90.682	463.182	-	-	-	5.467	275.814	1507.966	-	-	-	5.618	404.898	2274.669
CRYSTALS-Kyber	2.100	0.010	0.021	-	-	-	4.882	0.017	0.083	-	-	-	5.087	0.023	0.117
CRYSTALS-Kyber-90s	2.000	0.007	0.014	-	-	-	2.000	0.009	0.018	-	-	-	2.167	0.012	0.026
FRODO AES	4.152	0.381	1.582	-	-	-	3.759	0.793	2.981	-	-	-	2.637	1.320	3.481
FRODO SHAKE	5.705	1.073	6.121	-	-	-	5.690	2.272	12.927	-	-	-	5.749	3.998	22.985
hqc-1	3.866	0.082	0.317	-	-	-	4.020	0.149	0.599	-	-	-	4.152	0.230	0.955
hqc-2	-	-	-	-	-	-	4.082	0.159	0.649	-	-	-	3.983	0.242	0.964
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	4.004	0.252	1.009
LAC	4.267	0.015	0.064	-	-	-	4.480	0.025	0.112	-	-	-	4.265	0.034	0.145
LEDACrypt N02	4.958	1.161	5.756	-	-	-	5.028	3.517	17.681	-	-	-	4.869	7.578	36.900
LEDACrypt N03	4.952	0.505	2.501	-	-	-	4.981	1.690	8.418	-	-	-	4.952	4.586	22.711
LEDACrypt N04	5.074	0.815	4.135	-	-	-	5.018	2.380	11.942	-	-	-	5.035	4.91	24.722
LEDACrypt LT DFR64	4.966	285.367	1417.029	-	-	-	4.991	937.746	4680.601	-	-	-	4.993	2878.263	14371.200
LEDACrypt LT DFRSL	4.922	424.166	2087.827	-	-	-	5.006	1460.724	7311.765	-	-	-	5.067	4102.157	20787.330
NewHope CCA	4.947	0.019	0.094	-	-	-	-	-	-	-	-	-	5.086	0.035	0.178
NewHope CPA	4.533	0.015	0.068	-	-	-	-	-	-	-	-	-	5.179	0.028	0.145
NTRU-HRSS	-	-	-	-	-	-	2.982	0.109	0.325	-	-	-	-	-	-
NTS-KEM AVX	5.181	14.635	75.823	-	-	-	5.414	44.749	242.256	-	-	-	5.297	81.954	434.092
NTS-KEM SSE	5.138	15.408	79.173	-	-	-	5.246	45.944	241.025	-	-	-	5.115	86.742	443.702
Round5 Ring	4.588	0.017	0.078	-	-	-	5.127	0.055	0.282	-	-	-	5.217	0.069	0.360
Round5 Ring 5	5.136	0.022	0.113	-	-	-	5.343	0.035	0.187	-	-	-	5.092	0.065	0.331
Round5 Non-Ring	5.423	0.156	0.846	-	-	-	5.475	0.278	1.522	-	-	-	5.323	0.705	3.753
Round5 Ring LongKey	6.217	0.023	0.143	-	-	-	-	-	-	-	-	-	-	-	-
SABER	4.000	0.016	0.064	-	-	-	5.347	0.026	0.139	-	-	-	4.950	0.040	0.198
SIKE	5.214	1.781	9.287	5.245	2.493	13.075	5.246	4.206	22.065	-	-	-	5.247	7.068	37.085
SIKE Compressed	5.365	4.617	24.768	5.264	6.455	33.980	5.201	11.204	58.277	-	-	-	5.264	17.150	90.272
Three Bears	-	-	-	5.105	0.019	0.097	-	-	-	5.143	0.035	0.180	5.214	0.056	0.292
Three Bears Eph.	-	-	-	5.158	0.019	0.098	-	-	-	5.250	0.036	0.189	5.086	0.058	0.295

Table 15: Energy consumption of `crypto_kem_enc` function for **Encapsulation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	3.616	0.099	0.358	-	-	-	3.522	0.297	1.046	-	-	-	3.345	0.708	2.368
BIKE2	3.463	0.054	0.187	-	-	-	2.908	0.152	0.442	-	-	-	3.162	0.359	1.135
BIKE3	3.092	0.109	0.337	-	-	-	3.373	0.335	1.130	-	-	-	3.259	0.856	2.790
Classic McEliece AVX	3.105	0.019	0.059	-	-	-	2.846	0.039	0.111	-	-	-	1.554	0.056	0.087
Classic McEliece SSE	2.696	0.023	0.062	-	-	-	2.375	0.048	0.114	-	-	-	1.810	0.079	0.143
CRYSTALS-Kyber	6.923	0.013	0.09	-	-	-	5.143	0.021	0.108	-	-	-	5.000	0.029	0.145
CRYSTALS-Kyber-90s	2.250	0.008	0.018	-	-	-	1.667	0.012	0.020	-	-	-	2.438	0.016	0.039
FRODO AES	3.834	0.500	1.917	-	-	-	3.578	0.971	3.474	-	-	-	3.251	1.624	5.279
FRODO SHAKE	5.706	1.172	6.687	-	-	-	5.772	2.437	14.067	-	-	-	5.721	4.279	24.482
hqc-1	4.229	0.153	0.647	-	-	-	4.234	0.274	1.160	-	-	-	4.080	0.425	1.734
hqc-2	-	-	-	-	-	-	4.188	0.293	1.227	-	-	-	4.230	0.449	1.899
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	4.319	0.464	2.004
LAC	3.783	0.023	0.087	-	-	-	3.946	0.037	0.146	-	-	-	4.604	0.053	0.244
LEDAcrypt N02	5.310	0.0420	0.223	-	-	-	5.154	0.091	0.469	-	-	-	5.188	0.165	0.856
LEDAcrypt N03	5.290	0.031	0.164	-	-	-	5.080	0.075	0.381	-	-	-	5.058	0.171	0.865
LEDAcrypt N04	4.947	0.038	0.188	-	-	-	5.333	0.099	0.528	-	-	-	5.071	0.198	1.004
LEDAcrypt LT DFR64	5.192	0.130	0.675	-	-	-	5.023	0.258	1.296	-	-	-	5.063	0.559	2.830
LEDAcrypt LT DFRSL	4.896	0.164	0.803	-	-	-	5.120	0.541	2.770	-	-	-	5.018	0.843	4.230
NewHope CCA	4.724	0.029	0.137	-	-	-	-	-	-	-	-	-	4.648	0.054	0.251
NewHope CPA	4.435	0.023	0.102	-	-	-	-	-	-	-	-	-	4.619	0.042	0.194
NTRU-HRSS	-	-	-	-	-	-	3.853	0.034	0.131	-	-	-	-	-	-
NTS-KEM AVX	5.194	0.031	0.161	-	-	-	5.636	0.121	0.682	-	-	-	5.614	0.166	0.932
NTS-KEM SSE	5.290	0.031	0.164	-	-	-	5.508	0.124	0.683	-	-	-	5.568	0.169	0.941
Round5 Ring	5.370	0.027	0.145	-	-	-	5.109	0.092	0.470	-	-	-	5.136	0.118	0.606
Round5 Ring 5	5.389	0.036	0.194	-	-	-	5.797	0.059	0.342	-	-	-	5.121	0.107	0.548
Round5 Non-Ring	5.533	0.165	0.913	-	-	-	5.586	0.292	1.631	-	-	-	5.353	0.709	3.795
Round5 Ring LongKey	5.026	0.038	0.191	-	-	-	-	-	-	-	-	-	-	-	-
SABER	6.111	0.018	0.110	-	-	-	5.724	0.029	0.166	-	-	-	5.318	0.044	0.234
SIKE	5.162	2.907	15.006	5.199	4.061	21.113	5.253	7.736	40.641	-	-	-	5.300	11.406	60.455
SIKE Compressed	5.266	5.607	29.527	5.185	7.630	39.560	5.228	13.322	69.651	-	-	-	5.242	21.792	114.244
Three Bears	-	-	-	5.4	0.025	0.135	-	-	-	5.000	0.042	0.210	5.215	0.065	0.339
Three Bears Eph.	-	-	-	4.846	0.026	0.126	-	-	-	4.773	0.044	0.210	5.104	0.067	0.342

Table 16: Energy consumption of `crypto_kem_dec` function for **Decapsulation** of Round 2 **Key Encapsulation Mechanisms**. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	2.344	0.195	0.457	-	-	-	2.043	0.564	1.152	-	-	-	1.979	1.344	2.660
BIKE2	1.660	0.153	0.254	-	-	-	1.696	0.428	0.726	-	-	-	1.506	0.989	1.489
BIKE3	1.956	0.181	0.354	-	-	-	1.716	0.517	0.887	-	-	-	1.682	1.248	2.099
Classic McEliece AVX	5.278	0.036	0.190	-	-	-	2.557	0.070	0.179	-	-	-	3.143	0.084	0.264
Classic McEliece SSE	3.333	0.048	0.160	-	-	-	1.914	0.105	0.201	-	-	-	1.421	0.114	0.162
CRYSTALS-Kyber	4.700	0.010	0.047	-	-	-	4.765	0.017	0.081	-	-	-	5.400	0.025	0.135
CRYSTALS-Kyber-90s	2.333	0.006	0.014	-	-	-	3.111	0.009	0.028	-	-	-	3.308	0.013	0.043
FRODO AES	3.688	0.478	1.763	-	-	-	3.333	0.926	3.086	-	-	-	3.175	1.567	4.975
FRODO SHAKE	5.675	1.148	6.515	-	-	-	5.695	2.393	13.629	-	-	-	5.844	4.231	24.725
hqc-1	4.533	0.289	1.310	-	-	-	4.406	0.458	2.018	-	-	-	4.456	0.676	3.012
hqc-2	-	-	-	-	-	-	4.560	0.480	2.189	-	-	-	4.475	0.714	3.195
hqc-3	-	-	-	-	-	-	-	-	-	-	-	-	4.467	0.741	3.310
LAC	3.962	0.026	0.103	-	-	-	5.157	0.051	0.263	-	-	-	4.848	0.066	0.320
LEDAcrypt N02	4.996	0.251	1.254	-	-	-	4.979	0.630	3.137	-	-	-	5.079	1.176	5.973
LEDAcrypt N03	5.099	0.313	1.596	-	-	-	4.966	0.758	3.764	-	-	-	4.986	1.432	7.140
LEDAcrypt N04	5.113	0.822	4.203	-	-	-	5.009	1.882	9.427	-	-	-	5.046	2.984	15.056
LEDAcrypt LT DFR64	4.963	0.322	1.598	-	-	-	5.048	0.724	3.655	-	-	-	5.083	1.269	6.450
LEDAcrypt LT DFRSL	5.033	0.450	2.265	-	-	-	5.120	0.989	5.064	-	-	-	5.148	1.822	9.380
NewHope CCA	4.867	0.03	0.146	-	-	-	-	-	-	-	-	-	4.518	0.056	0.253
NewHope CPA	4.400	0.005	0.022	-	-	-	-	-	-	-	-	-	3.778	0.009	0.034
NTRU-HRSS	-	-	-	-	-	-	2.650	0.020	0.053	-	-	-	-	-	-
NTS-KEM AVX	6.289	0.128	0.805	-	-	-	6.341	0.232	1.471	-	-	-	6.144	0.425	2.611
NTS-KEM SSE	6.277	0.213	1.337	-	-	-	6.124	0.394	2.413	-	-	-	5.854	0.804	4.707
Round5 Ring	5.500	0.012	0.066	-	-	-	5.333	0.048	0.256	-	-	-	5.177	0.062	0.321
Round5 Ring 5	5.176	0.017	0.088	-	-	-	5.419	0.031	0.168	-	-	-	5.088	0.057	0.290
Round5 Non-Ring	5.400	0.070	0.378	-	-	-	5.202	0.099	0.515	-	-	-	4.972	0.396	1.969
Round5 Ring LongKey	3.389	0.018	0.061	-	-	-	-	-	-	-	-	-	-	-	-
SABER	5.412	0.017	0.092	-	-	-	5.379	0.029	0.156	-	-	-	5.533	0.045	0.249
SIKE	5.152	3.071	15.823	5.292	4.303	22.772	5.195	7.824	40.642	-	-	-	5.309	12.266	65.121
SIKE Compressed	5.219	5.230	27.293	5.220	7.122	37.180	5.250	12.684	66.586	-	-	-	5.225	20.062	104.814
Three Bears	-	-	-	4.800	0.040	0.192	-	-	-	5.097	0.062	0.316	5.122	0.090	0.461
Three Bears Eph.	-	-	-	5.100	0.010	0.051	-	-	-	5.385	0.013	0.070	4.750	0.016	0.076

5.2 Public-key Encryption

Table 17: Energy consumption of `crypto_encrypt_keypair` function for **Keypair Generation** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	5.014	283.240	1420.300	-	-	-	5.018	946.930	4751.320	-	-	-	5.035	2834.329	14271.160
LEDAcrypt DFRSL	4.903	399.954	1961.020	-	-	-	4.981	1470.086	7322.090	-	-	-	4.993	4230.839	21125.400
LAC	6.533	0.015	0.098	-	-	-	11.519	0.027	0.311	-	-	-	7.882	0.034	0.268

Table 18: Energy consumption of `crypto_encrypt` function for **Encryption** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	4.747	0.297	1.410	-	-	-	4.951	0.509	2.520	-	-	-	4.924	0.922	4.540
LEDAcrypt DFRSL	4.861	0.397	1.930	-	-	-	4.747	0.969	4.600	-	-	-	4.956	1.469	7.280
LAC	8.381	0.021	0.176	-	-	-	8.457	0.035	0.296	-	-	-	9.451	0.051	0.482

Table 19: Energy consumption of `crypto_encrypt_open` function for **Decryption** of Round 2 **Public-key Encryption** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
LEDAcrypt DFR64	5.108	0.695	3.550	-	-	-	5.072	1.313	6.660	-	-	-	5.052	2.324	11.740
LEDAcrypt DFRSL	5.052	0.968	4.890	-	-	-	5.117	1.962	10.040	-	-	-	5.069	3.551	18.000
LAC	13.333	0.006	0.080	-	-	-	10.167	0.018	0.183	-	-	-	11.556	0.018	0.208

5.3 Digital Signature

Table 20: Energy consumption of `crypto_sign_keypair` function for **Keypair Generation** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	4.370	0.027	0.118	4.395	0.038	0.167	4.811	0.053	0.255	5.055	0.073	0.369	-	-	-
CRYSTALS-Dilithium AES	5.000	0.020	0.100	4.107	0.028	0.115	4.421	0.038	0.168	4.755	0.049	0.233	-	-	-
GeMSS	4.991	13.027	65.017	-	-	-	5.051	65.317	329.908	-	-	-	5.058	201.090	1017.190
BlueGeMSS	4.966	13.228	65.691	-	-	-	4.997	65.497	327.297	-	-	-	4.950	201.510	997.568
RedGeMSS	5.816	13.216	76.858	-	-	-	5.229	65.017	339.969	-	-	-	5.015	201.378	1009.862
Luov Small Sig Chaacha	-	-	-	4.819	0.579	2.790	-	-	-	5.053	2.561	12.941	5.127	4.215	21.611
Luov Small Sig Keccak	-	-	-	5.213	0.992	5.171	-	-	-	5.489	3.216	17.652	5.559	5.386	29.943
MQDSS	-	-	-	5.121	0.214	1.096	-	-	-	5.123	0.504	2.582	-	-	-
Picnic UR	4.900	0.01	0.049	-	-	-	4.357	0.014	0.061	-	-	-	4	0.017	0.068
Picnic FS	2.600	0.010	0.026	-	-	-	4.143	0.014	0.058	-	-	-	4.647	0.017	0.079
Picnic2 FS	5.200	0.010	0.052	-	-	-	4.714	0.014	0.066	-	-	-	4.353	0.017	0.074
qTESLA	4.963	0.295	1.464	-	-	-	5.053	0.866	4.376	-	-	-	5.033	4.345	21.869
qTESLA-s	4.997	0.305	1.524	-	-	-	5.029	0.820	4.124	-	-	-	5.000	4.306	21.532
Rainbow Classic	4.744	2.758	13.084	-	-	-	4.669	26.103	121.867	-	-	-	4.804	38.481	184.880
Rainbow Compressed/Cyclic	4.749	2.966	14.087	-	-	-	4.634	29.421	136.343	-	-	-	4.874	41.078	200.194
Rainbow Cyclic	4.738	2.992	14.176	-	-	-	4.671	29.634	138.416	-	-	-	4.878	40.777	198.908
Rainbow SSE Classic	4.589	3.065	14.066	-	-	-	4.413	27.900	123.136	-	-	-	4.601	44.024	202.550
Rainbow SSE Compressed/Cyclic	4.625	3.174	14.680	-	-	-	4.459	31.716	141.406	-	-	-	4.693	47.663	223.691
Rainbow SSE Cyclic	4.675	3.188	14.904	-	-	-	4.490	31.929	143.365	-	-	-	4.654	47.686	221.944
SPHINCS+ SHA256 s simple	5.301	116.180	615.820	-	-	-	5.309	16.380	86.960	-	-	-	5.345	20.840	111.380
SPHINCS+ SHA256 s robust	5.039	223.330	1125.340	-	-	-	5.019	328.760	1650.120	-	-	-	5.206	85.540	445.310
SPHINCS+ SHA256 f simple	5.370	3.680	19.760	-	-	-	5.603	5.190	29.080	-	-	-	5.454	1.300	7.090
SPHINCS+ SHA256 f robust	5.305	7.380	39.150	-	-	-	5.323	10.180	54.190	-	-	-	5.400	5.320	28.730
SPHINCS+ SHAKE256 s simple	5.507	34.170	188.190	-	-	-	5.235	51.620	270.230	-	-	-	5.514	67.720	373.390
SPHINCS+ SHAKE256 s robust	5.473	66.180	362.230	-	-	-	5.519	94.640	522.330	-	-	-	5.545	124.650	691.150
SPHINCS+ SHAKE256 f simple	5.673	1.070	6.070	-	-	-	5.580	1.620	9.040	-	-	-	5.496	4.210	23.140
SPHINCS+ SHAKE256 f robust	5.609	2.070	11.610	-	-	-	5.515	2.930	16.160	-	-	-	5.564	7.770	43.230
SPHINCS+ HARAKA s simple	5.041	31.160	157.070	4.893	53.900	263.720	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA s robust	5.040	50.080	252.400	5.210	69.930	364.31	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f simple	7.010	1	7.010	6.179	1.620	10.010	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f robust	6.683	1.610	10.760	5.969	2.230	13.310	-	-	-	-	-	-	-	-	-

Table 21: Energy consumption of `crypto_sign` function for **Signing** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	5.026	0.078	0.392	4.849	0.126	0.611	4.55	0.151	0.687	4.500	0.162	0.729	-	-	-
CRYSTALS-Dilithium AES	4.692	0.052	0.244	4.567	0.097	0.443	4.112	0.143	0.588	4.172	0.145	0.605	-	-	-
GeMSS	4.890	254.548	1244.551	-	-	-	4.666	726.732	3391.091	-	-	-	4.699	1223.860	5751.300
BlueGeMSS	4.818	40.729	196.217	-	-	-	4.674	107.610	502.963	-	-	-	4.717	167.729	791.175
RedGeMSS	5.896	1.228	7.240	-	-	-	6.033	2.986	18.016	-	-	-	5.954	4.646	27.660
Luov Small Sig Chacha	-	-	-	4.368	0.288	1.258	-	-	-	4.283	0.756	3.238	4.387	1.224	5.370
Luov Small Sig Keccak	-	-	-	5.163	0.701	3.619	-	-	-	5.558	1.472	8.182	5.629	2.424	13.645
MQDSS	-	-	-	5.480	0.829	4.543	-	-	-	5.620	2.0949	11.769	-	-	-
Picnic UR	5.058	3.889	19.670	-	-	-	5.045	9.580	48.329	-	-	-	5.051	15.803	79.814
Picnic FS	4.848	3.053	14.801	-	-	-	5.010	7.167	35.907	-	-	-	4.857	12.518	60.796
Picnic2 FS	5.126	94.990	486.873	-	-	-	5.178	268.170	1388.546	-	-	-	5.162	535.599	2764.521
qTESLA	4.217	0.092	0.388	-	-	-	3.950	0.121	0.478	-	-	-	3.359	0.309	1.038
qTESLA-s	4.358	0.095	0.414	-	-	-	3.951	0.122	0.482	-	-	-	3.320	0.331	1.099
Rainbow Classic	3.480	0.025	0.087	-	-	-	4.821	0.162	0.781	-	-	-	5.132	0.205	1.052
Rainbow Compressed/Cyclic	4.388	2.143	9.404	-	-	-	4.503	18.979	85.453	-	-	-	4.558	29.571	134.780
Rainbow Cyclic	4.115	0.026	0.107	-	-	-	4.908	0.163	0.800	-	-	-	5.059	0.204	1.032
Rainbow SSE Classic	4.622	0.045	0.208	-	-	-	4.682	0.176	0.824	-	-	-	5.004	0.238	1.191
Rainbow SSE Compressed/Cyclic	4.335	2.215	9.602	-	-	-	4.369	20.098	87.802	-	-	-	4.464	32.432	144.781
Rainbow SSE Cyclic	4.533	0.045	0.204	-	-	-	4.644	0.177	0.822	-	-	-	4.884	0.241	1.177
SPHINCS+ SHA256 s simple	5.328	2042.830	10883.230	-	-	-	5.320	434.370	2310.780	-	-	-	5.314	274.920	1460.860
SPHINCS+ SHA256 s robust	5.336	3989.580	21286.640	-	-	-	5.395	7574.320	40865.710	-	-	-	5.393	1012.070	5457.770
SPHINCS+ SHA256 f simple	5.271	123.160	649.140	-	-	-	5.275	165.380	872.450	-	-	-	5.285	33.300	175.980
SPHINCS+ SHA256 f robust	5.283	240.240	1269.120	-	-	-	5.313	322.770	1714.890	-	-	-	5.412	124.390	673.160
SPHINCS+ SHAKE256 s simple	5.407	562.960	3043.750	-	-	-	5.501	1179.740	6489.690	-	-	-	5.535	830.420	4596.470
SPHINCS+ SHAKE256 s robust	5.511	993.210	5473.480	-	-	-	5.553	2037.460	11314.230	-	-	-	5.506	1489.370	8200.250
SPHINCS+ SHAKE256 f simple	5.523	35.120	193.970	-	-	-	5.498	46.890	257.790	-	-	-	5.434	101.480	551.490
SPHINCS+ SHAKE256 f robust	5.407	64.970	351.300	-	-	-	5.484	86.750	475.700	-	-	-	5.476	183.360	1004.150
SPHINCS+ HARAKA s simple	5.195	668.820	3474.440	5.188	1451.240	7529.410	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA s robust	5.203	898.180	4673.460	5.264	2258.900	11891.060	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f simple	3.761	40.620	152.790	4.720	48.980	231.180	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f robust	4.861	53.680	260.930	4.691	70.140	329.050	-	-	-	-	-	-	-	-	-






Table 22: Energy consumption of `crypto_sign_open` function for **Verification** of Round 2 **Digital Signature** schemes. Time is reported in milliseconds, energy in milliJoules, and power in Watts.

Scheme	Level 1			Level 2			Level 3			Level 4			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
CRYSTALS-Dilithium SHAKE	4.829	0.035	0.169	4.933	0.045	0.222	4.900	0.060	0.294	4.650	0.080	0.372	-	-	-
CRYSTALS-Dilithium AES	4.724	0.029	0.137	4.514	0.037	0.167	3.915	0.047	0.184	4.233	0.060	0.254	-	-	-
GeMSS	5.172	0.064	0.331	-	-	-	4.925	0.146	0.719	-	-	-	6.933	0.300	2.080
BlueGeMSS	4.938	0.065	0.321	-	-	-	5.156	0.147	0.758	-	-	-	5.164	0.305	1.575
RedGeMSS	8.478	0.069	0.585	-	-	-	7.805	0.159	1.241	-	-	-	7.749	0.311	2.410
Luov Small Sig Chacha	-	-	-	2.685	0.124	0.333	-	-	-	2.995	0.386	1.156	2.606	0.576	1.501
Luov Small Sig Keccak	-	-	-	5.022	0.537	2.697	-	-	-	5.518	1.100	6.070	5.523	1.762	9.731
MQDSS	-	-	-	5.578	0.555	3.096	-	-	-	5.657	1.429	8.084	-	-	-
Picnic UR	5.030	3.134	15.763	-	-	-	5.014	7.804	39.132	-	-	-	5.025	13.094	65.796
Picnic FS	4.821	2.481	11.962	-	-	-	5.012	5.957	29.856	-	-	-	4.877	10.520	51.307
Picnic2 FS	5.070	51.602	261.616	-	-	-	5.108	118.133	603.427	-	-	-	5.069	209.204	1060.452
qTESLA	5.800	0.025	0.145	-	-	-	5.660	0.047	0.266	-	-	-	4.663	0.086	0.401
qTESLA-s	5.200	0.025	0.130	-	-	-	5.182	0.044	0.228	-	-	-	4.872	0.086	0.419
Rainbow Classic	2.583	0.012	0.031	-	-	-	4.351	0.037	0.161	-	-	-	4.567	0.060	0.274
Rainbow Compressed/Cyclic	3.996	1.386	5.539	-	-	-	3.931	7.6152	29.937	-	-	-	3.922	18.521	72.646
Rainbow Cyclic	3.906	1.395	5.449	-	-	-	3.913	7.709	30.165	-	-	-	3.966	18.501	73.366
Rainbow SSE Classic	3.808	0.026	0.099	-	-	-	4.972	0.179	0.890	-	-	-	5.237	0.152	0.796
Rainbow SSE Compressed/Cyclic	3.881	1.413	5.484	-	-	-	3.977	7.788	30.973	-	-	-	3.951	18.490	73.050
Rainbow SSE Cyclic	3.971	1.412	5.607	-	-	-	3.999	7.814	31.252	-	-	-	3.937	18.558	73.067
SPHINCS+ SHA256 s simple	4.927	6.040	29.760	-	-	-	4.926	0.940	4.630	-	-	-	4.902	1.230	6.030
SPHINCS+ SHA256 s robust	4.938	12.260	60.540	-	-	-	4.898	19.500	95.520	-	-	-	4.932	3.680	18.150
SPHINCS+ SHA256 f simple	4.853	14.400	69.880	-	-	-	4.905	23.320	114.380	-	-	-	4.992	2.460	12.280
SPHINCS+ SHA256 f robust	4.943	29.600	146.300	-	-	-	4.896	48.620	238.020	-	-	-	4.942	7.210	35.630
SPHINCS+ SHAKE256 s simple	5.039	1.030	5.190	-	-	-	4.928	1.520	7.490	-	-	-	5.000	2.000	10.000
SPHINCS+ SHAKE256 s robust	4.801	2.060	9.890	-	-	-	4.721	2.980	14.070	-	-	-	5.005	3.800	19.020
SPHINCS+ SHAKE256 f simple	4.829	2.450	11.830	-	-	-	4.877	3.890	18.970	-	-	-	4.965	4.030	20.010
SPHINCS+ SHAKE256 f robust	4.541	4.950	22.480	-	-	-	4.699	7.750	36.420	-	-	-	4.945	7.870	38.920
SPHINCS+ HARAKA s simple	3.971	1.040	4.130	3.546	1.520	5.390	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA s robust	4.337	1.660	7.200	4.363	2.730	11.910	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f simple	3.691	2.361	8.710	3.642	3.690	13.440	-	-	-	-	-	-	-	-	-
SPHINCS+ HARAKA f robust	4.346	3.760	16.340	4.411	6.570	28.980	-	-	-	-	-	-	-	-	-

6 Discussion of Results

Using the energy consumption results shown in the previous section, we now rank the candidate submissions. In this ranking, we do not consider the different variants of each algorithm, but rather the best energy consuming variant in each submission package. In this way, we can best compare each proposed algorithm against each other. Further, we have visually distinguished each ranked position by its underlying mathematical problem through the use of different cell fill patterns, shown in Table 23. The ranks are ordered by increasing energy consumption and are categorized by function and security level. We have chosen not to rank the submissions for public-key encryption as there are too few to consider; this information can be easily gathered from the raw data in the previous section. Additionally, we have only included the categorizations based on security levels 1, 3, and 5 since very few submissions have targeted levels 2 and 4. To provide a comprehensive comparison, we have consolidated the results from levels 2 and 4 into the results from 1 and 3, respectively. Moreover, we have separated our results based on the implementation it uses. For those pertaining to the Assembly Optimized Implementation, we also provide a number in brackets within each entry; this number represents the scheme’s rank in the Optimized C Implementation for ease of comparison. Lastly, it should be noted that it was not required for submitters to provide an Assembly Optimized Implementation. As a result, our comparison here is limited to those schemes for which the submitters have provided Assembly Optimized Implementations. Likewise, those which have provided such submissions may not have used the same types of instructions in their implementation. We refer the reader to Table 3 and 4 to see the types of assembly optimizations each candidate has made.

Table 23: Legend used to indicate the underlying mathematics on which the cryptosystem is based.

Pattern	Underlying Problem
	Lattice-Based
	Code-Based
	Rank-based
	Multivariate-based
	Hash-Based
	Other

6.1 Key Encapsulation Mechanisms

We list the five most energy-efficient submissions for KEMs using the Optimized C Implementation in Section 6.1.1 and those using the Assembly Optimized Implementation in Section 6.1.2. In each section, we provide a ranking for the three functions performing keypair generation, encapsulation and decapsulation. We also consider it to be meaningful to compare the total energy required to complete all three operations. This would be indicative of the total energy needed to establish a shared secret. This can be seen in Table 27 and Table 31.

When analyzing the Optimized C results, the average power across all security levels taken with regard to each of the three functions ranges from 24.89-25.73W. It is clear that the majority of the most efficient algorithms are lattice-based; this is expected as lattice-based cryptosystems are known to be among the most efficient [2]. When it comes to key encapsulation, however, lattice-based schemes occupy only 9 of the 15 positions with the other ranks held by code-based algorithms. It is also interesting to note that Level 2- and Level 4-secure submission Three Bears, which is also lattice-based, is more energy-efficient than most Level 1 and Level 3 submissions while providing a stronger security guarantee. When considering the total energy

expended, once again, lattice-based algorithms seem to rank best. Even though code-based algorithms such as NTS-KEM and Classic McEliece offer competitive results for encapsulation, the total energy needed is dominated by the energy required for key generation, where Classic McEliece performs the worst in terms of energy required of all submitted algorithms.

Considering the Assembly Optimized Implementations, the average power across all security levels taken with regard to each of the three functions ranges from 4.58-4.73W. The energy-efficiency of the lattice-based schemes is even more prominent when we analyze the rankings of the Assembly Optimized Implementations, where nearly all ranking positions for keypair generation, encapsulation, decapsulation, and total energy consumed is occupied by one of the many lattice-based schemes under consideration. This is a testament to lattice-based cryptosystems’ high parallelizability. Here, CRYSTALS-Kyber tops almost all categories being evaluated. This is due in part to their use of AVX2 instructions and use of hardware-accelerated symmetric primitives. It is clear that the performance of candidate submissions can be drastically improved when hardware support for symmetric primitives based on the Keccak permutation become available. Another implementation which has taken advantage of both of these avenues for assembly optimization is SABER. This improvement is showcased by quite a drastic leap in energy-efficiency ranking from the Optimized C to the Assembly Optimized Implementation. The only other implementation that provides AES-NI optimization in addition to AVX2 is BIKE; unfortunately, we were obliged to build the Assembly Optimized Implementation of this scheme without AES-NI support due to a compiler bug/incompatibility. It would be interesting to see whether the use of these instructions within this scheme would improve its energy footprint in a comparable way to SABER and CRYSTALS-Kyber.

6.1.1 Optimized C Implementation

Table 24: The top five most energy-efficient submissions for **Keypair Generation** of **Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Keypair Generation		
	Level 1	Level 3	Level 5
1	Round5	Round5	Round5
2	Three Bears	Three Bears	Three Bears
3	LAC	SABER	LAC
4	SABER	LAC	SABER
5	BIKE	ROLLO	ROLLO

Table 25: The top five most energy-efficient submissions for **Encapsulation of Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Encapsulation		
	Level 1	Level 3	Level 5
1	NTS-KEM	Three Bears	Three Bears
2	Three Bears	Round5	Round5
3	Round5	Classic McEliece	LAC
4	LAC	LAC	NTS-KEM
5	Classic McEliece	NTS-KEM	Classic McEliece

Table 26: The top five most energy-efficient submissions for **Decapsulation of Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Decapsulation		
	Level 1	Level 3	Level 5
1	Round5	Three Bears	Three Bears
2	Three Bears	Round5	Round5
3	NewHope	LAC	NewHope
4	LAC	CRYSTALS-Kyber	LAC
5	NTS-KEM	NTS-KEM	CRYSTALS-Kyber

Table 27: The top five most energy-efficient submissions for **total** energy consumed of **Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Total		
	Level 1	Level 3	Level 5
1	Round5	Three Bears	Three Bears
2	Three Bears	Round5	Round5
3	LAC	LAC	LAC
4	NewHope	CRYSTALS-Kyber	★★★★ROLLO★★★★
5	CRYSTALS-Kyber	★★★★ROLLO★★★★	NewHope

6.1.2 Assembly Optimized Implementation

Table 28: The top five most energy-efficient submissions for **Keypair Generation of Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Entry colour distinguishes the problem on which algorithm is based. The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Keypair Generation		
	Level 1	Level 3	Level 5
1	CRYSTALS-Kyber (7)	CRYSTALS-Kyber (6)	CRYSTALS-Kyber (7)
2	LAC (3)	LAC (4)	LAC (3)
3	SABER (4)	SABER (3)	NewHope (6)
4	NewHope (8)	Three Bears (2)	SABER (4)
5	Round5 (1)	Round5 (1)	ThreeBears (2)

Table 29: The top five most energy-efficient submissions for **Encapsulation of Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Encapsulation		
	Level 1	Level 3	Level 5
1	CRYSTALS-Kyber (8)	CRYSTALS-Kyber (8)	CRYSTALS-Kyber (9)
2	Classic McEliece (5)	Classic McEliece (3)	Classic McEliece (5)
3	LAC (4)	NTRU (12)	NewHope (8)
4	NewHope (9)	LAC (4)	SABER (11)
5	SABER (14)	SABER (10)	LAC (3)

Table 30: The top five most energy-efficient submissions for **Decapsulation of Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Decapsulation		
	Level 1	Level 3	Level 5
1	CRYSTALS-Kyber (6)	CRYSTALS-Kyber (4)	NewHope (3)
2	NewHope (3)	NTRU (11)	CRYSTALS-Kyber (5)
3	Three Bears (2)	Three Bears (1)	Three Bears (1)
4	Round5 (1)	SABER (8)	Classic McEliece (14)
5	LAC (4)	Round5 (2)	SABER (9)

Table 31: The top five most energy-efficient submissions for **total** energy consumed of **Key Encapsulation Mechanisms**. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Total		
	Level 1	Level 3	Level 5
1	CRYSTALS-Kyber (5)	CRYSTALS-Kyber (4)	CRYSTALS-Kyber (5)
2	NewHope (4)	SABER (6)	NewHope (4)
3	LAC (3)	Three Bears (1)	SABER (7)
4	SABER (8)	NTRU (14)	LAC (3)
5	Three Bears (2)	LAC (3)	Three Bears (1)

6.2 Digital Signatures

Tables 32-37 demonstrate the same ranking process as described previously but here, we consider the results of the candidate digital signature algorithms. Unlike in the KEM ranking, we do not provide a total energy consumption metric here. This is due to the nature of digital signatures; the focus on which underlying functions should be optimized from an energy-efficiency standpoint will shift based on the application for which the digital signature scheme is deployed. Algorithms with efficient signing would generally be preferred, for example, in wireless sensor networks where resource-constrained devices must have a means to transmit authentic data measurements to a base station [36]. In contrast, applications such as public-key certification where a single message is to be signed once and verified by the masses are ideally designed to have an efficient verification procedure [37].

We first study the Optimized C Implementations from Tables 32-34. The average power across all security levels taken with regard to each of the three functions ranges from 25.60-26.65W. Although Picnic, an algorithm which is based on a novel hard problem, dominates the energy consumption metrics for keypair generation, it is unable to compare to the other submissions when it comes to signing and verification. In fact, its Picnic2 FS variant consumes the most energy of all when verifying a signature. When comparing the raw energy measurements from Tables 12 and 13, we can see that all of the submitted schemes, except for Rainbow, possess algorithms which are more energy-efficient when performing signature verification as opposed to signature generation. This may lead one to believe that it would be more desirable to use an algorithm like Rainbow in applications where signing is required to be most efficient. Despite this, there are several lattice-based techniques which consume even less energy to perform a signing operation while boasting even better verification metrics. Based on the rankings, algorithms such as CRYSTALS-Dilithium, qTESLA, and Falcon are among the most energy-efficient for both signing and verification procedures. When it comes to verification, multivariate-based scheme GeMSS and its BlueGeMSS variant perform comparatively to their closest ranked lattice-based scheme, having only between 12-64% difference in energy consumption across the different security levels studied. This is in contrast to the next best multivariate algorithm, Rainbow, whose percentage difference is 51-173% between its closest ranked lattice-based counterpart.

We now shift our attention to the Assembly Optimized Implementations. The average power across all security levels taken with regard to each of the three functions ranges from 4.75-5.03W. The ranking of

schemes while performing keypair generation has not changed significantly; we can see in Table 35 that the submissions’ rankings have merely shifted a position or two at each security level. CRYSTALS-Dilithium, following a similar procedure as in its KEM scheme, used both AVX2 and AES-NI optimizations in its signature scheme; however, it is not top ranked in keypair generation, signing, or verification as it was for nearly all security levels for its KEM submission. Further, as we mentioned in the last paragraph, Rainbow is the only scheme whose signing step is more energy-efficient than its verification step. This trend persists in the Assembly Optimized Implementation; however, unlike in the Optimized C rankings, here it is also the most energy-efficient when performing signing compared to the other implementations studied.

6.2.1 Optimized C Implementation

Table 32: The top five most energy-efficient submissions for **Keypair Generation** of **Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Keypair Generation		
	Level 1	Level 3	Level 5
1	Picnic	Picnic	Picnic
2	CRYSTALS-Dilithium	CRYSTALS-Dilithium	qTESLA
3	qTESLA	qTESLA	SPHINCS+
4	MQDSS	MQDSS	Luov
5	SPHINCS+	SPHINCS+	Falcon

Table 33: The top five most energy-efficient submissions for **Signing** of **Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Signing		
	Level 1	Level 3	Level 5
1	Rainbow	CRYSTALS-Dilithium	Falcon
2	qTESLA	qTESLA	qTESLA
3	Falcon	Rainbow	Rainbow
4	CRYSTALS-Dilithium	Luov	Luov
5	Luov	GeMSS	Picnic

Table 34: The top five most energy-efficient submissions for **Verification** of **Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23).

Rank	Verification		
	Level 1	Level 3	Level 5
1	qTESLA	qTESLA	Falcon
2	Falcon	CRYSTALS-Dilithium	qTESLA
3	GeMSS	GeMSS	GeMSS
4	Rainbow	Rainbow	Rainbow
5	CRYSTALS-Dilithium	SPHINCS+	SPHINCS+

6.2.2 Assembly Optimized Implementation

Table 35: The top five most energy-efficient submissions for **Keypair Generation** of **Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Keypair Generation		
	Level 1	Level 3	Level 5
1	Picnic (1)	Picnic (1)	Picnic (1)
2	CRYSTALS-Dilithium (2)	CRYSTALS-Dilithium (2)	SPHINCS+ (3)
3	MQDSS (4)	MQDSS (4)	qTESLA (2)
4	qTESLA (3)	qTESLA (3)	Luov (4)
5	Luov (6)	SPHINCS+ (5)	Rainbow (6)

Table 36: The top five most energy-efficient submissions for **Signing** of **Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Signing		
	Level 1	Level 3	Level 5
1	Rainbow (1)	qTESLA (2)	Rainbow (3)
2	CRYSTALS-Dilithium (4)	CRYSTALS-Dilithium (1)	qTESLA (2)
3	qTESLA (2)	Rainbow (4)	Luov (4)
4	Luov (6)	Luov (4)	GeMSS (6)
5	MQDSS (9)	MQDSS (8)	Picnic (5)

Table 37: The top five most energy-efficient submissions for **Verification of Digital Signature** submissions. Schemes targeting Level 2 and 4 are included in Level 1 and 3. Fill patterns correspond to the underlying problem on which the algorithm is based (see Table 23). The number in brackets represents the scheme’s rank in the Optimized C Implementation.

Rank	Verification		
	Level 1	Level 3	Level 5
1	Rainbow (4)	Rainbow (4)	Rainbow (4)
2	qTESLA (1)	CRYSTALS-Dilithium (2)	qTESLA (2)
3	CRYSTALS-Dilithium (5)	qTESLA (1)	Luov (6)
4	GeMSS (3)	GeMSS (3)	GeMSS (3)
5	Luov (7)	Luov (6)	SPHINCS+ (5)

7 Conclusion

In this work, we have reported our measurements of the energy consumption by the PQC Round 2 candidates, including Optimized C Implementations as well as Assembly Optimized Implementations. Results have been categorized by cryptographic function and proposed security level. Candidates have been ranked based on their energy consumption to demonstrate which schemes are most energy-efficient. Our results show that lattice-based schemes tend to be very efficient in practice and are highly parallelizable, a trait which can bring about further energy savings when vectorized instructions are available on the target platform. When considering signing operations, we see that multivariate-based schemes are very competitive with their lattice-based competitors, a trend that is even more evident when assembly instructions are used. It is important to note that our ranking only provides one metric of evaluation; a holistic approach should be used when determining which algorithm best suites an application. We hope to use our findings in the future to pinpoint which subroutines of the candidate submissions expend the most energy to provide direction for further optimizations.

Acknowledgments

This research was supported in part through a grant provided by the Natural Sciences and Engineering Research Council of Canada. The authors would also like to thank Tanushree Banerjee for her guidance and support in the preliminary stages of this project.

Notes

Updated June 7, 2019.

References

- [1] “Post-Quantum Cryptography Standardization Call for Proposals Announcement,” 2017. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>.

- [2] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone, “Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process,” tech. rep., National Institute of Standards and Technology, 2019.
- [3] “Post-Quantum Cryptography Round 2 Submissions,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [4] “Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process,” 2016. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf>.
- [5] T. Banerjee and M. A. Hasan, “Energy Consumption of Candidate Algorithms for NIST PQC Standards,” tech. rep., University of Waterloo Centre for Applied Cryptographic Research, 2018.
- [6] “IgProf, The Ignominous Profiler,” 2018. Accessed: May-2019. [Online]. Available: <https://igprof.org/>.
- [7] “PQC - API notes,” 2017. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/example-files/api-notes.pdf>.
- [8] “Let’s Get Ready to Rumble: The NIST PQC “Competition”,” 2018. Accessed: May-2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Presentations/Let-s-Get-Ready-to-Rumble-The-NIST-PQC-Competiti/images-media/PQCrypto-April2018_Moody.pdf.
- [9] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Güneysu, C. A. Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, “BIKE: Bit Flipping Key Encapsulation,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [10] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, and W. Wang, “Classic McEliece: conservative code-based cryptography,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [11] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Kyber,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [12] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, “FrodoKEM: Learning With Errors Key Encapsulation,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [13] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, “Hamming Quasi-Cyclic (HQC),” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [14] X. Lu et al., “LAC: Lattice-Based Cryptosystems,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.

- [15] M. Baldi, A. Barengi, F. Chiaraluce, G. Pelosi, and P. Santini, “LEDACrypt: Low-density parity-check code-based cryptographic systems,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [16] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, and D. Stebila, “NewHope,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [17] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang, “NTRU,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [18] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, “NTRU Prime: Round 2,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [19] M. Albrecht, C. Cid, K. G. Paterson, C. J. Tjhai, and M. Tomlinson, “NTS-KEM,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [20] C. A. Melchor, N. Aragon, M. Bardet, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, A. Hauteville, A. Otmani, O. Ruatta, J.-P. Tillich, and G. Zémor, “ROLLO-Rank-Ouroboros, LAKE & LOCKER,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [21] H. Baan, S. Bhattacharya, S. Fluhrer, O. Garcia-Morchon¹, T. Laarhoven, R. Player, R. Rietman¹, M.-J. O. Saarinen, L. Tolhuizen¹, J. L. Torre-Arce, and Z. Zhang, “Round5: KEM and PKE based on (Ring) Learning with Rounding,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [22] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, A. Couvreur, J.-C. Deneuville, P. Gaborit, A. Hauteville, and G. Zémor, “Rank quasi-cyclic (RQC),” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [23] J.-P. D’Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, “SABER: Mod-LWR based KEM (Round 2 Submission),” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [24] D. Jao et al., “Supersingular isogeny key encapsulation,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [25] M. Hamburg, “Post-quantum cryptography proposal: THREEBEARS,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [26] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Dilithium,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.

- [27] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, “FALCON: Fast-Fourier Lattice-based Compact Signatures over NTRU,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [28] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem, “GeMSS: A Great Multivariate Short Signature,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [29] W. Beullens et al., “LUOV,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [30] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe, “MQDSS specifications,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [31] G. Zaverucha, “The Picnic Signature Algorithm,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [32] N. Bindel et al., “Submission to NIST’s post-quantum project (2nd Round): lattice-based digital signature scheme qTESLA,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [33] J. Ding et al., “Rainbow - Algorithm Specification and Documentation,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [34] J.-P. Aumasson, D. J. Bernstein, M. E. C. Dobraunig, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, and P. Schwabe, “SPHINCS+,” 2019. Accessed: May-2019. [Online]. Available: <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions>.
- [35] K. N. Khan, “Energy Profiling using IgProf,” in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.
- [36] J. Liu, J. Baek, J. Zhou, Y. Yang, and J. Wong, “Efficient online/offline identity-based signature for wireless sensor network,” *International Journal of Information Security*, 2010.
- [37] A. Menezes, S. Vanstone, and P. V. Oorschot, *Handbook of Applied Cryptography*, ch. 11. Digital Signatures. CRC Press, 1996.