IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 4, 2021, accepted April 19, 2021, date of publication May 5, 2021, date of current version May 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077843

Energy Efficiency Analysis of Post-Quantum Cryptographic Algorithms

CRYSTAL ANDREA ROMA[1], CHI-EN AMY TAI[2], AND M. ANWAR HASAN[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[2]Department of Management Sciences, University of Waterloo, Waterloo, ON N2L 3G1, Canada

Corresponding author: M. Anwar Hasan (ahasan@uwaterloo.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

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 post-quantum cryptography (PQC) standardization process initiated by the National Institute of Standards and Technology (NIST) is well underway. 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 use of battery-operated devices, high-performance computing environments where energy costs are critical, as well as in the interest of green computing. In this paper, the energy consumption of PQC candidates is evaluated on an Intel Core i7-6700 CPU using PAPI, the Performance API. 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. Lastly, IgProf is used to identify the most energy-consuming subroutines within a select number of submissions to highlight potential areas for optimization.

INDEX TERMS Post-quantum cryptography, energy consumption, digital signature, key encapsulation mechanism, public-key cryptography.

I. 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. Given the expansion in quantum computing research in recent years, it is possible that a large-scale quantum computer may be realized in the foreseeable future. Under the quantum paradigm, many mathematical problems which were once deemed intractable may be easily solved. With this, much of today's public-key infrastructure will become obsolete. In order to avoid such a catastrophic breach of security, the National Institute of Standards and Technology (NIST) in 2017 launched its post-quantum cryptography (PQC) standardization project [1]. The project had sixty-four candidate algorithms for Round 1, narrowed the number to twenty-six for Round 2, and is now at an advanced stage with fifteen algorithms for Round 3 - seven as finalists and eight alternatives [2]. NIST's aim is to develop new quantum-resistant standards similar to the classical digital signature and key establishment schemes published in the Federal Information Processing Standards Publication (FIPS) 186 and NIST Special Publications (SP) 800-56 A and B [3].

Candidate PQC algorithms are evaluated based on correctness, speed, and size of keys, ciphertexts, and signatures [4]. Although not an official criterion for evaluation by NIST, 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 as well as high-performance computing environments

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

where the energy consumption of software translates directly into maintenance and cooling costs. Energy efficiency of software has also gained more attention due to the idea of green computing, a movement striving to achieve more environmentally-friendly IT by emphasizing the importance of energy efficiency from both a software and hardware standpoint.

A. RELATED WORK

Energy consumption of cryptographic algorithms is not as widely reported as execution time, a trend that is not unique to cryptography per say, but of software profiling in general. As suggested by the authors in [5], there is an increasing ''battery gap'' motivated by a mismatch between the energy needed by security processing and the available battery and processing capabilities. Their work provides the first comprehensive energy measurement of SSL/TLS using external sensors to calculate the energy of the device under test. More recent works have used software-based energy profiling techniques similar to those used in this work. For instance, the Running Average Power Limit (RAPL) interface is used to measure the energy consumed by encryption and decryption operations for common cryptographic techniques such as the Caesar Cipher, RSA, AES, and Triple DES in [6]. The authors in [7] also use RAPL to measure the energy consumption of lightweight stream ciphers using chaos-based cryptosystems. In [8], the Intel Power Gadget is used to compare the energy consumption of elliptic curve point addition and doubling using different coordinate systems. The work in [9] also uses a software-based energy measurement tool known as powerstat to analyze two implementations of AES CBC to compare the energy consumed by a basic software implementation of AES versus one that takes advantage of special x86 instructions on an Intel platform.

In addition to the PQC standardization, NIST is currently holding a lightweight cryptography competition in which candidates are to be evaluated based on metrics including area, memory, energy consumption, and performance [10]. Despite the importance of energy consumption in settings in which lightweight ciphers would be used, few of the candidate algorithms have reported this metric in their submission packages opting to instead focus on latency and area metrics. For those Round 2 lightweight candidates that have reported energy consumption including [11]–[17], these results have been reported for hardware implementations of their respective algorithms rather than software-based designs as has been done in this work.

Unlike the lightweight cryptography standardization process, energy consumption is not an evaluation criterion for the post-quantum cryptography process. As a result, most PQC algorithms have published detailed information pertaining to memory and timing requirements; energy metrics have not been provided. Although limited, there have been independent studies on the energy consumed by some PQC algorithms. The work in [8] compares the energy consumption of Supersingular Isogeny Based Diffie Hellman (SIDH) post-quantum secure key exchange algorithm against Elliptic Curve Diffie Hellman where it was shown that SIDH consumes 37 to 47 times more energy compared to ECDH targeting the same security levels. The author in [18] has provided energy analyses on a Cortex M4 platform for a number of PQC algorithms with focus directed to mobile energy consumption. Detailed results targeting the full breadth of submissions on such a platform is not possible as many algorithms under consideration have only implemented software for the required Intel x86-64 target platform.

B. SCOPE AND CONTRIBUTIONS

NIST Round 3 candidate algorithms represent about 61.5% of the Round 2 algorithms (one of the Round 3 algorithms is a merger of two from Round 2). In this paper we do not restrict our analysis to Round 3 only, rather we consider all twenty-six Round 2 candidates. This is rationalized by the fact that cryptanalysis has already impacted some of the Round 3 algorithms and NIST has expressed concerns with a lack of diversity of the algorithms chosen, especially in the case of digital signature schemes [19]. Secondly, Round 2 algorithms that did not move forward to Round 3 include those with similar designs and performance metrics to their counterparts in Round 3, and those which simply did not receive enough community attention, prompting NIST to encourage additional research into many of those Round 2 schemes, specifically rank-based cryptosystems such as ROLLO and RQC [2].

This paper extends the work completed in [20] in which the energy of NIST PQC Round 1 candidates were studied. Many Round 2 algorithms have a variety of changes in their parameters and optimized implementations compared to their Round 1 versions such as the LEDAcrypt algorithm which added an additional optimized implementation exploiting Intel AVX2 extensions [21] and the GeMSS digital signature algorithm which added two new parameters shown to be significantly faster than its Round 1 parameter set [22]. In this paper, we report energy consumption results not only for the algorithms' optimized implementations written in portable ANSI C, but also for their other implementations, if any, that are more efficient but platform-specific. While providing detailed power, execution time, and energy metrics for most Round 2 algorithms' parameter sets, our work in this paper considers various security levels and different security notions, such as IND-CCA and IND-CPA, which have been achieved by the algorithms. Energy consumption analysis of algorithms for different security can be crucial for some protocols. For example, algorithms achieving IND-CCA security notions may be able to perform a single key generation step and continue using that same keypair across multiple encapsulation and decapsulation operations, which is however not the case for those only achieving IND-CPA security. Additionally, our work provides information related to subroutine energy consumption to provide greater insight into possible avenues of optimization. We also provide instructions on how

to perform these experiments so that they can be repeated by others in the community.

C. ORGANIZATION

The rest of this paper is organized as follows. Section 2 gives some preliminaries of the PQC algorithms under study. Section 3 describes the method by which the energy consumption of each algorithm is captured. The energy profiling results for each operation are given in Section 4. All respective analyses, discussions, and internal subroutine energy consumption data are provided in Section 5. Lastly, concluding remarks are provided in Section 6.

II. PRELIMINARIES

A. PQC ALGORITHM FUNCTIONALITY AND SECURITY

For the energy analysis performed, all twenty-six algorithms from Round 2 of NIST's PQC Standardization Process are considered. Details of these algorithms along with their specific parameter sets are available online on NIST's website [23] and are not reviewed here for brevity. The algorithms under study target key encapsulation or digital signature operations. Each of these cryptographic functions requires a triple of algorithms as stated below.

Key Encapsulation Mechanisms (KEM) provide a means by which two parties can establish a shared secret. There are three main operations in each proposed KEM:

1) `crypto_kem_keypair` produces a public key, `pk`, and a corresponding secret key, `sk`.
2) `crypto_kem_enc` takes the public key, `pk`, as input, produces a shared secret, `ss`, and a ciphertext of that shared secret, `ct`.
3) `crypto_kem_dec` takes the ciphertext, `ct`, and secret key, `sk`, as input to reproduce the shared secret, `ss`, as output.

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

1) `crypto_sign_keypair` produces a public key, `pk`, and a private key, `sk`.
2) `crypto_sign` 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`.
3) `crypto_sign_open` is a routine which verifies a signed message `sm`, its length, `smlen`, using the public key, `pk`, and the original message, `m`, of length `mlen`.

Based on the parameters specified, the algorithms adhere to five NIST-defined levels of security, listed below.

Level 1: Algorithm is at least as hard to break as AES128.
Level 2: Algorithm is at least as hard to break as SHA256.
Level 3: Algorithm is at least as hard to break as AES192.
Level 4: Algorithm is at least as hard to break as SHA384.
Level 5: Algorithm is at least as hard to break as AES256.

B. ALGORITHM CATEGORIZATION AND IMPLEMENTATIONS

Of the twenty-six Round 2 algorithms, there are seventeen KEM schemes and nine digital signature algorithms. In Table 1 and Table 2, the algorithms are categorized based on the underlying PQC family, namely lattice, code, rank, and isogeny for key encapsulation and lattice, multivariate, hash, and other for digital signature schemes. Such a classification is quite broad as there are a number of different problems within each family. For simplicity, this classification is used as well as the aforementioned functionality and security level to report the energy measurement data in subsequent sections of this work. Additional information for each respective algorithm can be found within the appropriate documentation cited in Table 1 and Table 2.

TABLE 1. Categorization of KEM schemes based on the mathematics of the cryptosystem.

Scheme	Lattice	Code	Rank	Isogeny
BIKE [24]		✓		
Classic McEliece [25]		✓		
FrodoKEM [26]	✓			
HQC [27]		✓		
Kyber [28]	✓			
LAC [29]	✓			
LEDAcrypt [21]		✓		
NewHope [30]	✓			
NTRU [31]	✓			
NTRU Prime [32]	✓			
NTS-KEM [33]		✓		
ROLLO [34]			✓	
Round5 [35]	✓			
RQC [36]			✓	
SABER [37]	✓			
SIKE [38]				✓
Three Bears [39]	✓			

TABLE 2. Categorization of digital signature schemes based on the mathematics of the cryptosystem.

Scheme	Lattice	Multivariate	Hash	Other
Dilithium [40]	✓			
Falcon [41]	✓			
GeMSS [22]		✓		
LUOV [42]		✓		
MQDSS [43]		✓		
Picnic [44]				✓
qTESLA [45]	✓			
Rainbow [46]		✓		
SPHINCS+ [47]			✓	

For the purpose of the NIST standardization process, most algorithms have a number of different parameter sets and implementations. Each algorithm has at least two implementations: a *reference implementation* for algorithm comprehension and an *optimized implementation* to demonstrate performance [4]. Both are written in portable ANSI C. Some algorithms have a third realization that makes use of platform-specific instructions, including Single Instruction Multiple Data (SIMD) extensions such as any of the Streaming SIMD Extensions (SSE) or Advanced Vector Extensions (AVX) as well as Advanced Encryption Standard New

TABLE 3. Energy consumed by key encapsulation mechanisms targeting security level 1 in optimized C implementations.

Scheme	Security Notion	Key Generation			Encapsulation			Decapsulation		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CCA	16.368	0.242	3.965	16.440	0.302	4.957	16.343	1.879	30.709
BIKE1	IND-CPA	16.224	0.203	3.289	16.187	0.247	4.003	16.091	0.960	15.447
BIKE2	IND-CCA	15.996	0.443	7.084	16.344	0.118	1.923	16.090	1.596	25.687
BIKE2	IND-CPA	15.929	0.377	5.999	16.283	0.103	1.680	15.921	0.882	14.046
BIKE3	IND-CCA	15.985	0.161	2.577	16.350	0.226	3.689	16.210	1.892	30.673
BIKE3	IND-CPA	16.053	0.124	1.992	16.270	0.209	3.404	15.639	1.186	18.549
Classic McEliece	IND-CCA2	16.798	159.096	2672.428	16.372	0.063	1.026	14.351	19.038	273.216
FRODO AES	IND-CCA	15.317	13.754	210.663	15.226	14.038	213.739	15.229	14.019	213.495
FRODO SHAKE	IND-CCA	15.823	3.026	47.882	15.929	3.278	52.212	15.938	3.253	51.839
hqc-1	IND-CCA2	16.315	0.350	5.710	16.305	0.669	10.911	16.251	1.055	17.137
Kyber	IND-CCA2	16.207	0.039	0.639	16.223	0.051	0.826	16.216	0.060	0.966
Kyber-90s	IND-CCA2	16.098	0.063	1.015	16.116	0.076	1.220	16.137	0.083	1.341
LAC	IND-CCA	16.717	0.036	0.596	16.652	0.059	0.977	16.790	0.078	1.314
LEDAcrypt DFR64	IND-CCA2	15.529	397.097	6166.531	16.145	2.543	41.050	15.271	3.597	54.931
LEDAcrypt DFRSL	IND-CCA2	15.456	616.768	9532.513	16.288	3.920	63.846	15.243	4.416	67.315
LEDAcrypt N02	IND-CPA	16.093	12.125	195.132	16.233	0.679	11.028	15.200	3.262	49.581
LEDAcrypt N03	IND-CPA	16.014	4.069	65.160	16.232	0.550	8.925	15.360	3.813	58.567
LEDAcrypt N04	IND-CPA	16.008	3.864	61.858	16.154	0.699	11.288	15.342	5.481	84.082
NewHope	IND-CCA	16.077	0.039	0.630	15.999	0.058	0.936	16.023	0.065	1.037
NewHope	IND-CPA	16.184	0.034	0.548	16.236	0.049	0.793	16.030	0.012	0.186
NTRU LPrime	IND-CCA2	14.709	6.855*	100.821*	14.529	12.276*	178.364*	14.511	18.388*	266.832*
NTRU sPrime	IND-CCA2	14.845	60.916*	904.314*	14.654	6.863*	100.571*	14.458	18.538*	268.019*
NTRU-HPS	IND-CCA2	15.774	3.531	55.700	16.199	0.227	3.681	16.398	0.489	8.026
NTS-KEM	IND-CCA2	16.286	16.965	276.292	16.978	0.029	0.489	16.045	0.217	3.480
ROLLO-I	IND-CPA	16.308	0.776	12.657	16.217	0.172	2.788	16.025	0.552	8.849
ROLLO-III	IND-CPA	16.203	0.159	2.579	16.185	0.346	5.606	16.034	0.531	8.517
Round5 N1 0d AES	IND-CPA	16.530	1.584	26.190	15.726	1.605	25.239	15.593	0.090	1.410
Round5 N1 0d SHAKE	IND-CPA	16.597	1.598	26.522	15.819	1.619	25.609	15.648	0.096	1.504
Round5 ND 0d AES	IND-CPA	16.097	0.058	0.932	16.076	0.106	1.701	15.966	0.055	0.878
Round5 ND 0d SHAKE	IND-CPA	15.818	0.059	0.928	15.987	0.106	1.688	15.786	0.053	0.840
Round5 ND 5d AES	IND-CPA	16.139	0.045	0.734	16.148	0.084	1.355	16.141	0.043	0.694
Round5 ND 5d SHAKE	IND-CPA	16.094	0.045	0.719	16.071	0.083	1.331	15.922	0.041	0.660
RQC	IND-CCA2	16.010	0.284	4.548	16.060	0.609	9.786	16.085	2.894	46.554
SABER	IND-CCA	16.151	0.031	0.504	15.919	0.040	0.635	16.056	0.044	0.704
SIKE	IND-CCA	14.979	17.620	263.930	14.960	28.787	430.651	14.965	30.734	459.935
SIKE Compressed	IND-CCA	14.989	42.890	642.895	14.986	52.704	789.801	14.992	48.995	734.548
Three Bears	IND-CCA	16.924	0.021*	0.349*	16.966	0.026*	0.440*	16.903	0.039*	0.665*
Three Bears	IND-CPA	16.805	0.020*	0.341*	16.913	0.026*	0.439*	16.819	0.009*	0.157*

Instructions (AES-NI). In many cases, this *additional optimized implementation* may only apply to a subset of the parameter sets belonging to a particular algorithm. The reader is referred to each algorithm's respective documentation for more information on the specific optimizations applied to each variant.

III. EXPERIMENT METHODOLOGY

A. ENERGY MEASUREMENT OF THREE MAIN OPERATIONS

Modern Intel CPUs are equipped with the RAPL interface, a feature which provides access to energy and performance counters. Depending on the platform, energy measurements from the system's sockets (Package), CPU cores and caches (Power Plane 0), GPU (Power Plane 1), or the energy consumed by memory (DRAM) is available for sampling [48]. To obtain the total timing and energy measurements for each operation, PAPI, the Performance API, is used and configured to support RAPL. PAPI is a platform-independent library which provides access to performance measurements across the hardware and software stack [49].

In this work, a simple C main file is written which calls the three operations required by the cryptographic schemes being studied. Once the PAPI event set is properly initialized, the energy and time required for an operation to complete is obtained. A minimum of 1000 iterations of each operation are executed. Both the Package and the DRAM energy values are measured with the results reporting the sum of the two values. Based on the expected performance of each of the submissions under consideration, it is anticipated that many of the candidate algorithms will execute much faster than the update rate of the tool (~1ms). In these cases, a loop is used to increase the number of iterations of the algorithm performed. When profiling for digital signature schemes, the results will change based on the message that is being signed. To provide consistent data between tests, a text file containing 1000 randomly generated 3300 byte messages is created to be used by all digital signature schemes.[1]

As mentioned in Section 2, each algorithm must include an optimized implementation, which is a basic portable C implementation. In order to make a reasonable comparison between each algorithm, these implementations are first profiled to give a fair, baseline comparison of all submitted schemes. This will be referred to as the *optimized C implementation*. In some cases, the optimized C implementation is a copy of the reference implementation.

[1]The files used can be found in the supplemental material of this work.

TABLE 4. Energy consumed by digital signature algorithms targeting security level 1 in optimized C implementations.

Scheme	Key Generation			Signing			Verification		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.392	0.171	2.632	15.868	0.591	9.384	15.575	0.176	2.74
Dilithium SHAKE	15.738	0.080	1.259	16.117	0.419	6.745	15.815	0.100	1.585
Falcon	14.804	6.388	94.570	16.111	0.237	3.812	15.407	0.045	0.687
BlueGeMSS	16.218	492.553	7988.135	16.078	614.812	9885.020	16.026	8.657	138.747
GeMSS	16.175	653.995	10578.378	16.057	4521.190	72597.049	15.971	8.473	135.319
RedGeMSS	16.110	359.028	5783.957	15.907	13.137	208.971	16.022	8.843	141.683
Luov Large Chacha	14.952	2.384*	35.645*	15.649	7.706*	120.590*	15.210	5.644*	85.838*
Luov Large Keccak	15.333	2.626*	40.256*	15.817	7.974*	126.124*	15.556	5.722*	89.015*
Luov Small Chacha	14.851	4.143*	61.534*	15.603	1.195*	18.648*	15.234	0.893*	13.606*
Luov Small Keccak	15.242	4.739*	72.228*	16.263	1.745*	28.386*	16.204	1.415*	22.924*
MQDSS	15.220	0.325*	4.943*	15.225	7.940*	120.886*	15.674	5.812*	91.097*
Picnic FS	15.635	0.011	0.175	15.727	3.497	55.006	15.709	2.843	44.668
Picnic UR	16.000	0.011	0.176	15.919	4.423	70.417	15.924	3.571	56.871
Picnic2 FS	16.455	0.011	0.181	16.414	122.254	2006.729	16.581	57.325	950.495
qTESLA	15.286	0.346	5.289	15.648	0.180	2.824	15.522	0.040	0.623
qTESLA-p	15.517	1.624	25.199	15.620	1.034	16.148	15.787	0.258	4.072
qTESLA-s	15.721	0.348	5.471	16.021	0.187	3.000	15.786	0.040	0.633
Rainbow Classic	15.857	7.380	117.024	15.651	0.115	1.795	12.586	0.060	0.755
Rainbow Compressed Cyclic	15.817	8.202	129.734	15.861	4.527	71.803	15.922	1.467	23.362
Rainbow Cyclic	15.833	8.222	130.183	16.667	0.112	1.871	15.959	1.472	23.485
SPHINCS+ Haraka f robust	15.154	7.470	113.199	15.175	275.340	4178.272	15.158	11.619	176.115
SPHINCS+ Haraka f simple	15.737	5.014	78.904	15.497	181.492	2812.645	15.720	7.560	118.849
SPHINCS+ Haraka s robust	15.112	237.476	3588.636	15.131	4410.841	66739.039	15.133	5.165	78.169
SPHINCS+ Haraka s simple	15.692	158.852	2492.696	15.385	2890.033	44463.508	15.673	3.368	52.781
SPHINCS+ SHA-2 f robust	15.192	3.113	47.294	15.167	93.494	1417.981	15.158	4.219	63.950
SPHINCS+ SHA-2 f simple	15.787	1.612	25.448	15.587	50.952	794.194	15.769	2.102	33.146
SPHINCS+ SHA-2 s robust	15.193	98.942	1503.235	15.192	1376.377	20909.488	15.187	1.743	26.479
SPHINCS+ SHA-2 s simple	15.767	51.468	811.515	15.515	764.511	11861.73	15.754	0.882	13.902
SPHINCS+ SHAKE f robust	15.566	5.262	81.910	15.586	158.587	2471.673	15.623	6.991	109.214
SPHINCS+ SHAKE f simple	15.940	2.752	43.867	15.969	87.223	1392.845	15.930	3.560	56.714
SPHINCS+ SHAKE s robust	15.454	168.425	2602.785	15.782	2351.608	37114.162	15.996	2.903	46.435
SPHINCS+ SHAKE s simple	15.969	88.445	1412.365	16.018	1319.814	21141.276	16.010	1.494	23.916

Certain algorithm implementations use assembly instructions within their optimized C implementations (such as GeMSS, qTESLA, LEDAcrypt, MQDSS, and HQC). In these cases, the reference implementation is used as part of this experiment set. Further, some algorithms provide additional implementations which have been designed to better showcase their algorithms' achievable performance. These additional implementations may have compiler options which allow the application to be built using platform-specific optimizations and instruction-set extensions to target more modern processors, while others have hand optimized portions of their applications using x86 assembly, and some have improved performance with customization inherently related to their algorithm. These implementations will be collectively referred to as the *additional optimized implementations*.The experiments are grouped based on a corresponding security level. When reporting results, each entry name specifies the specific implementation tested following the algorithm's naming conventions. For details about the differences between these variants, the reader is referred to the supporting documentation provided in each algorithm's submission package. In cases where the proposed algorithm includes multiple implementations targeting the same security level, results are reported for the lowest energy-consuming variant.

All experiments have been performed on a 64-bit processor Intel Core i7-6700 CPU running at 3.40GHz with 8GB of RAM running Ubuntu 16.04 LTS. To be consistent with methodologies many of the candidates have used when profiling their own algorithms, all experiments were performed with only one active CPU core while Hyperthreading and Turbo Boost were disabled. All implementations have been compiled using gcc version 9.2.1. In the case of the optimized C experiments, the goal is to provide a fair, baseline comparison between all implementations. As a result, any `-m` type options which may have been included in an algorithm's Makefile have been omitted. These options direct the compiler to make use of special platform-specific instructions and extensions [50]. The choice of optimization flag such as `-Ofast` and `-O3` have been shown to improve the execution time of software and by consequence, improve the overall energy consumed [51]. In this work, we choose to build the optimized C experiment set using the `-O3` flag as most software packages have used this option in their own builds. In the case of the additional optimized implementations, candidates have used specialized instructions. Consequently, these implementations have been compiled with the same flags provided in each submission package's Makefile.

B. ENERGY MEASUREMENT OF SUBROUTINES

To gain better insight into an algorithm's implementation which contribute most to its energy profile, another set of experiments is performed on a subset of the optimized C implementation candidates. Although PAPI could be used to

obtain function-level energy usage of all candidates, it would require instrumenting all individual functions comprising each algorithm operation. To demonstrate which subroutines are contributing most to an operation's energy consumption, IgProf is used [52]. IgProf works on the basis of statistical sampling, leveraging PAPI to obtain measurements from the RAPL interface at a fixed interval and attributing the accumulated energy to the current location of code execution [53]. This experiment is performed on the three most energy-consuming and three least energy-consuming algorithms for both the KEM and digital signature schemes under investigation targeting security level 1.

IV. ENERGY CONSUMPTION RESULTS BY ALGORITHM

The results of the energy consumption of each of the three operations which comprise the cryptographic algorithms previously described are given. For brevity, only the level 1 results are shown here. Very few submissions have targeted level 2; these have been consolidated into the level 1 results, as well. These algorithms have been emphasized by an asterisk (∗). For those schemes which have included parameter sets targeting both level 1 and 2, only the lower level result is considered. The results targeting all security levels can be found in Appendix. At each level, the most energy-consuming algorithm implementation is distinguished in red text, while the least energy-consuming scheme is marked in green. All timing results are reported in milliseconds (ms) and all energy measurements in milliJoules (mJ). The average power is recorded in Watts (W). All results have been rounded to the third decimal place.

V. ANALYSIS AND DISCUSSION OF RESULTS

To demonstrate the range of energy consumption, the energy measurement results of the optimized C implementation experiment set have been plotted against time and distinguished by the underlying cryptographic family by color in Fig. 1 and Fig. 2. Due to the large variance in energy consumption and execution time of the algorithms studied, a logarithmic scale is used on both axes. For this same reason, the median is used as opposed to the average to quantitatively compare the algorithm families. Most algorithms under study have a number of variants; the lowest energy-consuming variant of each candidate is used to calculate the median as opposed to each individual measurement result. In this way, the median is not skewed towards the algorithm that has the most parameter sets. It is observed that lattice-based algorithms have the lowest median energy consumption across all security levels for the tuple of functions required for key encapsulation. Taking the level 1 results as an example, lattice schemes show 0.639mJ, 0.977mJ, and 0.966mJ level 1 median energy consumption for key generation, encapsulation, and decapsulation, respectively. Compared to code-based schemes at the same level, a 61.858mJ, 1.681mJ, and 17.137mJ median energy consumption is observed. Between ROLLO and RQC, the two rank-based algorithms under consideration, a level 1 median energy consumption

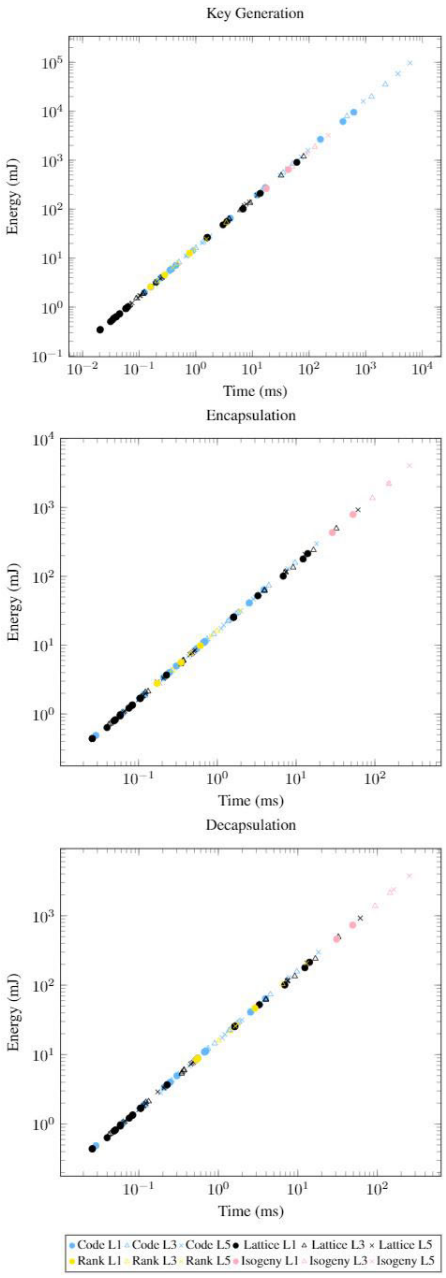

FIGURE 1. Energy consumed by key encapsulation mechanisms of the optimized C implementation set.

of 3.565mJ, 6.287mJ, and 27.535mJ is seen for key generation, encapsulation, and decapsulation. On the other hand, the single isogeny-based KEM scheme studied, SIKE, consumes 263.930mJ, 430.651mJ, and 459.935mJ for the three functions when considering its level 1 uncompressed variant.

When considering the digital signature submissions, it is observed once again that the lattice-based submissions are most energy-efficient. They have the lowest median energy consumption across all security levels; lattice schemes show 1.259mJ, 3.812mJ, and 0.687mJ level 1 median energy consumption for key generation, signing, and verification, respectively. When considering the multivariate algorithms

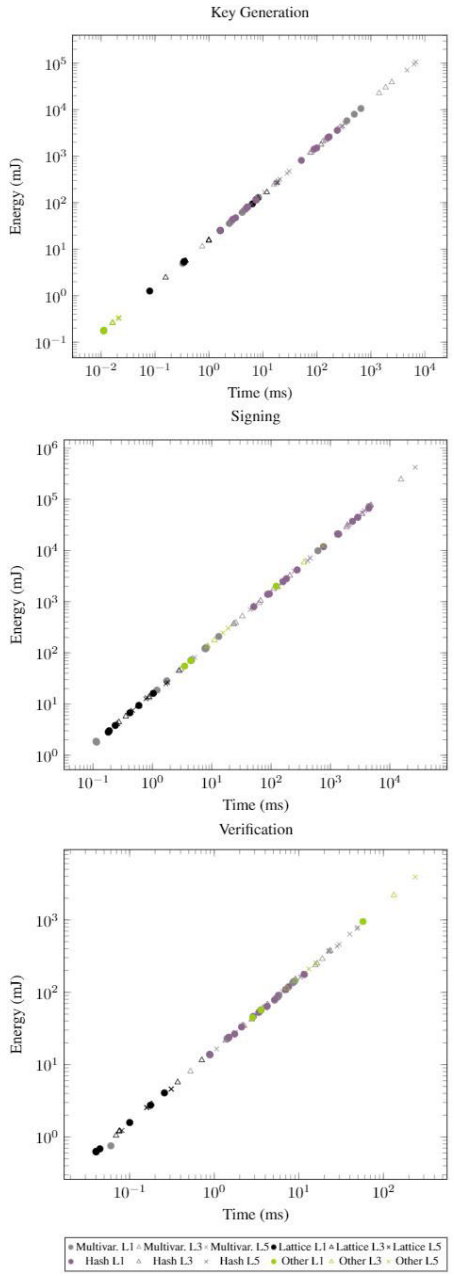

FIGURE 2. Energy consumed by digital signature schemes of the optimized C implementation set.

studied, the median energy consumed for these three functions is 76.335mJ, 69.767mJ, and 52.353mJ. Although the median energy consumed by multivariate schemes is significantly more than the lattice schemes studied for digital signature computation, there are several schemes which are very competitive with the lattice-based algorithms, as will be studied in proceeding sections. There are two digital signature schemes studied which are based on symmetric primitives including SPHINCS+, a hash-based scheme, and Picnic, a digital signature algorithm based on symmetric key primitives and zero-knowledge proofs. The lowest energy-consuming parameter set of the SPHINCS+ algorithm consumes 25.448mJ for key generation, 794.194mJ for signing, and 13.902mJ for verification at the level 1 security target. On the other hand, Picnic can achieve 0.175mJ, 55.006mJ, and 44.668mJ for these three functions. In general, based on the median values and the plots in Fig. 2, the energy required to perform verification is less than that to perform signing.

A. EFFECT OF ALGORITHM ON POWER CONSUMPTION

Power usage on today's CMOS-based CPUs can be represented as the sum of dynamic power and static power. The primary energy consumption is the result of switching in transistors which is accounted for in the dynamic power and can be represented as:

$$P_{dyn} = \frac{1}{2}\alpha C V^2 f \tag{1}$$

where α is a constant related to the activity, C is the capacitive load, V represents the voltage, and f the operating frequency. Dynamic power consumption is directly proportional to the operating frequency so that for a fixed task, lowering the frequency will lower the power usage [54]. By fixing the frequency of the test platform, we try to minimize the fluctuations in the measurements obtained. In addition to frequency, voltage can have a significant impact on the dynamic power consumption of a CPU. Dynamic voltage and frequency scaling (DVFS) is a technique used on modern processors to dynamically scale the operating voltage and frequency in response to the current workload and temperature [55]. It was shown in [56] that the power consumption of a Skylake processor, the target platform used in this study, remains relatively stable despite fluctuations in temperature.

In this work, we measure the average energy, represented in mJ, and the execution time, measured in ms. The average power dissipation is represented as the ratio of average energy consumed over this period of time.

$$P_{avg} = Energy/Time \tag{2}$$

As a result, the energy consumed by an application can be reduced if the execution time is reduced without significantly increasing the power dissipation or if the power dissipation is reduced without a proportional increase in runtime [55]. It is observed in Fig. 1 and Fig. 2 that there is a strong correlation between the energy consumption and execution time of each algorithm. When analyzing the optimized C results, the average power across all security levels is 15.957W, 15.973W, and 15.743W for key generation, encapsulation, and decapsulation, respectively (see Tables 11-13). In the case of the digital signature schemes, the optimized C implementations show an average power of 15.469W, 15.456W, and 15.422W required to perform key generation, signing, and verification across all levels (see Tables 14-16).

It is noted that there is not a significant change in the power consumed by each algorithm. As an example, a relative standard deviation of about 4% is observed in the power consumption metrics for the additional optimized key generation

TABLE 5. Energy consumed by key encapsulation mechanisms targeting security level 1 in additional optimized implementations.

Scheme	Security Notion	Key Generation			Encapsulation			Decapsulation		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CPA	15.580	0.033	0.508	15.866	0.042	0.668	15.493	0.173	2.684
BIKE2	IND-CPA	16.611	0.372	6.183	16.350	0.092	1.510	15.514	0.167	2.587
BIKE3	IND-CPA	16.013	0.098	1.576	16.116	0.192	3.101	15.636	0.256	4.000
Classic McEliece AVX	IND-CCA2	16.354	50.749	829.949	16.983	0.022	0.380	16.812	0.039	0.655
Classic McEliece SSE	IND-CCA2	16.281	111.460	1814.652	16.335	0.027	0.437	16.377	0.051	0.834
FRODO AES	IND-CCA	16.984	0.404	6.858	16.435	0.562	9.232	16.467	0.538	8.853
FRODO SHAKE	IND-CCA	18.073	1.190	21.505	17.955	1.287	23.105	17.999	1.264	22.743
hqc-1	IND-CCA2	16.044	0.096	1.547	16.091	0.166	2.664	15.886	0.317	5.042
Kyber	IND-CCA2	16.918	0.011	0.184	16.762	0.014	0.236	16.863	0.011	0.182
Kyber-90s	IND-CCA2	16.177	0.007	0.117	16.086	0.009	0.148	16.148	0.006	0.105
LAC	IND-CCA	16.704	0.019	0.324	16.580	0.030	0.490	16.726	0.036	0.599
LEDAcrypt DFR64	IND-CCA2	15.670	325.242	5096.631	15.950	0.137	2.178	15.645	0.333	5.210
LEDAcrypt DFRSL	IND-CCA2	15.847	464.128	7355.061	16.268	0.163	2.648	15.597	0.433	6.750
LEDAcrypt N02	IND-CPA	15.524	1.274	19.779	16.091	0.043	0.687	15.409	0.259	3.997
LEDAcrypt N03	IND-CPA	15.486	0.548	8.488	15.929	0.033	0.518	15.664	0.343	5.369
LEDAcrypt N04	IND-CPA	15.699	0.862	13.528	16.058	0.040	0.644	15.669	0.759	11.886
NewHope	IND-CCA	16.717	0.021	0.355	16.713	0.032	0.528	16.738	0.032	0.541
NewHope	IND-CPA	16.841	0.017	0.284	16.870	0.025	0.420	16.592	0.005	0.089
NTS-KEM AVX	IND-CCA2	16.116	16.196	261.013	16.757	0.027	0.456	16.333	0.118	1.924
NTS-KEM SSE	IND-CCA2	15.940	17.090	272.424	16.643	0.027	0.456	16.030	0.202	3.245
Round5 N1 0d AES	IND-CPA	16.514	0.140	2.317	16.952	0.169	2.867	15.883	0.072	1.142
Round5 N1 0d SHAKE	IND-CPA	16.467	0.153	2.520	16.917	0.182	3.080	15.722	0.077	1.205
Round5 ND 0d AES	IND-CPA	16.387	0.014	0.231	16.450	0.023	0.372	16.242	0.011	0.174
Round5 ND 0d SHAKE	IND-CPA	16.377	0.016	0.255	16.449	0.024	0.395	16.408	0.011	0.173
Round5 ND 5d AES	IND-CPA	16.175	0.019	0.314	16.223	0.031	0.501	16.043	0.015	0.236
Round5 ND 5d SHAKE	IND-CPA	16.015	0.021	0.329	16.224	0.032	0.517	16.096	0.015	0.238
SABER	IND-CCA	17.016	0.018	0.304	17.008	0.020	0.338	16.977	0.019	0.325
SIKE	IND-CCA	16.936	1.932	32.719	16.942	3.135	53.112	16.935	3.357	56.845
SIKE Compressed	IND-CCA	16.740	4.957	82.971	16.794	6.048	101.565	16.820	5.576	93.789
Three Bears	IND-CCA	16.489	0.022*	0.366*	16.459	0.029*	0.475*	16.376	0.045*	0.744*
Three Bears	IND-CPA	16.250	0.022*	0.363*	16.594	0.029*	0.488*	16.191	0.011*	0.180*

TABLE 6. Energy consumed by digital signature algorithms targeting security level 1 in additional optimized implementations.

Scheme	Key Generation			Signing			Verification		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.799	0.030	0.477	16.077	0.103	1.653	15.858	0.039	0.619
Dilithium SHAKE	16.712	0.042	0.695	16.867	0.126	2.129	16.583	0.048	0.788
BlueGeMSS	16.473	16.318	268.803	15.394	52.282	804.838	16.735	0.053	0.889
GeMSS	16.080	16.192	260.368	15.098	319.657	4826.297	16.361	0.052	0.855
RedGeMSS	15.947	16.168	257.821	15.174	1.444	21.908	16.436	0.055	0.908
Luov Small Chacha	15.697	0.706*	11.077*	17.677	0.295*	5.214*	17.427	0.125*	2.181*
Luov Small Keccak	15.767	1.183*	18.648*	16.484	0.777*	12.810*	16.087	0.607*	9.761*
MQDSS	16.223	0.235*	3.817*	17.147	0.984*	16.869*	17.579	0.658*	11.570*
Picnic FS AVX	16.010	0.011	0.179	17.070	1.783	30.431	17.166	1.433	24.602
Picnic FS SSE	16.013	0.011	0.175	15.948	3.422	54.568	15.958	2.770	44.197
Picnic UR AVX	15.814	0.011	0.176	16.993	2.200	37.391	17.019	1.780	30.286
Picnic UR SSE	15.797	0.012	0.182	16.033	4.388	70.346	16.195	3.524	57.068
Picnic2 FS AVX	16.453	0.011	0.186	17.389	71.053	1235.536	17.202	33.453	575.467
Picnic2 FS SSE	16.410	0.011	0.181	17.052	104.405	1780.298	16.894	56.541	955.210
qTESLA	16.141	0.330	5.320	16.825	0.101	1.696	16.595	0.026	0.426
qTESLA-s	16.133	0.329	5.301	17.002	0.104	1.766	16.604	0.026	0.427
Rainbow Classic AVX	17.021	3.033	51.619	16.861	0.028	0.475	17.051	0.014	0.235
Rainbow Classic SSE	16.567	3.146	52.126	16.403	0.050	0.820	16.832	0.028	0.475
Rainbow Compressed Cyclic AVX	17.091	3.256	55.654	16.901	2.286	38.635	16.548	1.414	23.402
Rainbow Compressed Cyclic SSE	16.567	3.416	56.591	16.553	2.326	38.501	16.418	1.439	23.622
Rainbow Cyclic AVX	17.046	3.255	55.482	16.884	0.028	0.478	16.469	1.417	23.336
Rainbow Cyclic SSE	16.591	3.363	55.793	16.348	0.050	0.823	16.433	1.432	23.532
SPHINCS+ Haraka f robust	17.217	0.116	1.994	16.470	4.759	78.388	14.737	0.343	5.047
SPHINCS+ Haraka f simple	17.160	0.097	1.661	16.282	3.587	58.410	15.049	0.221	3.331
SPHINCS+ Haraka s robust	16.642	3.594	59.806	16.529	81.522	1347.437	15.386	0.153	2.353
SPHINCS+ Haraka s simple	16.383	2.972	48.686	16.045	60.205	965.996	15.507	0.100	1.547
SPHINCS+ SHA-2 f robust	17.613	0.666	11.727	17.489	22.098	386.464	16.195	3.078	49.853
SPHINCS+ SHA-2 f simple	17.286	0.325	5.617	17.204	10.966	188.664	15.853	1.510	23.934
SPHINCS+ SHA-2 s robust	17.291	21.219	366.907	17.397	362.855	6312.659	15.936	1.283	20.445
SPHINCS+ SHA-2 s simple	17.310	10.350	179.155	17.393	179.498	3121.931	15.964	0.634	10.122
SPHINCS+ SHAKE f robust	17.557	2.169	38.075	17.422	67.830	1181.753	15.834	5.051	79.976
SPHINCS+ SHAKE f simple	17.511	1.138	19.932	17.423	37.148	647.238	15.998	2.538	40.597
SPHINCS+ SHAKE s robust	17.537	69.344	1216.122	17.552	1029.428	18068.813	15.893	2.093	33.266
SPHINCS+ SHAKE s simple	17.525	36.385	637.642	17.539	577.403	10127.173	15.981	1.060	16.947

TABLE 7. The five most energy-efficient algorithms for key encapsulation mechanisms and digital signature operations considering the optimized C implementation set. White cell color indicates a lattice scheme, code-based schemes are in blue, multivariate schemes are in gray, hash-based schemes in purple, and other symmetric schemes are in green.

Operation	Rank	Key Encapsulation Mechanisms			Digital Signature		
		Level 1	Level 3	Level 5	Level 1	Level 3	Level 5
Key Generation	1	Three Bears	Three Bears	Three Bears	Picnic	Picnic	Picnic
	2	SABER	SABER	NewHope	Dilithium	Dilithium	qTESLA
	3	NewHope	Kyber	SABER	MQDSS	MQDSS	SPHINCS+
	4	LAC	LAC	Kyber	qTESLA	qTESLA	LUOV
	5	Kyber	Round5	LAC	SPHINCS+	SPHINCS+	Falcon
Encapsulation/ Signing	1	Three Bears	Three Bears	Three Bears	Rainbow	qTESLA	Falcon
	2	NTS-KEM	SABER	NewHope	qTESLA	Falcon	qTESLA
	3	SABER	Kyber	SABER	Falcon	Rainbow	Rainbow
	4	NewHope	NTS-KEM	Kyber	Dilithium	Dilithium	LUOV
	5	Kyber	Classic McEliece	NTS-KEM	LUOV	LUOV	Picnic
Decapsulation/ Verification	1	Three Bears	Three Bears	Three Bears	qTESLA	Falcon	Falcon
	2	NewHope	SABER	NewHope	Falcon	qTESLA	qTESLA
	3	Round5	Kyber	SABER	Rainbow	Dilithium	Rainbow
	4	SABER	Round5	Kyber	Dilithium	Rainbow	SPHINCS+
	5	Kyber	LAC	Round5	LUOV	SPHINCS+	LUOV

algorithms studied in this work and approximately a 14% difference between the maximum and minimum power measurements. This result is quite different from the lightweight experiments using the ARM Cortex M4 described in [18] where a relative standard deviation of 22% is observed and about a 50% difference in the power consumed by the maximum and minimum key generation algorithms. Generally speaking, x86 processors are known to draw more power than ARM cores. ARM processors are based on a RISC instruction set architecture (ISA) and are considered to be power-optimized while x86 processors are based on a CISC architecture and most are considered to be performance-optimized. It is shown in [57] that the choice of power or performance-optimized core design has a greater impact on the core power consumption as opposed to the ISA itself. In fact, the authors also show that despite differences in power, performance-optimized x86-based platforms use only slightly more energy than power-optimized ARM processors. In addition to differences in ISA, embedded processors typically operate at lower frequency, may be fabricated using different semiconductor technologies, and may use different data path lengths; these are only a few of the differences which make it difficult to make a correlation between the energy measurements obtained in this work to the energy consumption expected on a low-end embedded processor. There have been works which have studied the differences in energy consumed per instruction on x86 and ARM processors, other studies demonstrating differences in total energy, power, and performance across various benchmarks on the two different processors, as well as projects which show how different workloads and resource imbalances lead to unexpected performance and energy profiles on the power-efficient processors [57]–[59]; however, this goes beyond the scope of this work.

B. ENERGY CONSUMPTION AT ALGORITHMIC LEVEL

For each function, the algorithms are ranked by energy consumption so that a rank of 1 signifies the most energy-efficient scheme. In this ranking, the lowest energy-consuming variant of each algorithm is used to best compare each proposed algorithm against each other. Further, each ranking is visually distinguished by the underlying cryptographic family to which the scheme belongs. The results have been separated based on the implementation type.

1) KEY ENCAPSULATION MECHANISMS

It is clear in Table 7 that the majority of the most efficient algorithms are lattice-based. In fact, level 2- and level 4-secure lattice-based submission Three Bears is more energy-efficient than most level 1 and level 3 submissions while targeting a higher level of security. Lattice-based cryptography is generally regarded to be very efficient and even better performance is attainable using different variants of the underlying hard problem [3]. When it comes to the encapsulation operation, however, the energy consumed by code-based schemes such as NTS-KEM and Classic McEliece are very competitive with their lattice-based counterparts. Despite having energy-efficient encapsulation operations, the energy required for key generation within these schemes is very high. It should be noted, however, that both of these schemes achieve indistinguishably under adaptive chosen ciphertext attack (IND-CCA2) security properties, unlike many of the lattice-schemes studied which have variants achieving only indistinguishably under chosen plaintext attack (IND-CPA), as well. As a result, the keypairs derived by these candidates can be used for long periods of time which may justify the large amount of energy required. Although the energy consumption of these schemes is not as low, code-based cryptographic algorithms are competitive alternatives to lattice-based ones as they have been well-studied and have few security vulnerabilities [3].

2) DIGITAL SIGNATURE

Table 7 also displays the same ranking process as it pertains to the digital signature schemes under study. Although Picnic, an algorithm which is based on a novel hard problem based in symmetric cryptographic primitives, is ranked

TABLE 8. Top energy-consuming subroutines needed for KEM operations. Results are shown for the three *least* and *most* energy-consuming candidates. White cell color indicates a lattice scheme, code-based schemes are in blue, rank-based algorithms in yellow, and isogeny-based schemes in pink.

Operation	Three Least Energy-Consuming Algorithms			Three Most Energy-Consuming Algorithms		
	Scheme	Energy	Subroutine	Scheme	Energy	Subroutine
Key Generation	Three Bears (IND-CPA)	44% 26% 13%	KeccakP1600_Permute_24rounds mac_3120 noise	LEDAcrypt LT DFRSL	77% 23% 0%	DFR_test gf2x_mod_inverse left_bit_shift_n
	SABER	32% 30% 11%	karatsuba_simple KeccakF1600_StatePermute toom_cook_4way	Classic McEliece	50% 8% 3%	pk_gen gf_mul merge
	NewHope (IND-CPA)	36% 21% 12%	KeccakF1600_StatePermute ntt montgomery_reduce	NTRU sPrime	64% 22% 8%	uint32_divmod_uint14 int32_mod_uint14 ZKeyGen
Encapsulation	Three Bears (IND-CPA)	39% 30% 10%	KeccakP1600_Permute_24rounds mac_3120 noise	SIKE Compressed	58% 32% 5%	mp_mul rdc_mont fp2mul434_mont
	NTS-KEM	51% 16% 6%	nts_kem_encapsulate keccakf random_uint16_bounded	FRODO AES	94% 2% 1%	Cipher frodo_mul_add_sa_plus_e KeccakF1600_StatePermute
	SABER	39% 33% 14%	karatsuba_simple KeccakF1600_StatePermute toom_cook_4way	NTRU Lprime	67% 24% 8%	uint32_divmod_uint14 int32_mod_uint14 Rq_mult_small
Decapsulation	Three Bears (IND-CPA)	31% 26% 15%	KeccakP1600_Permute_24rounds mac_3120 noise	SIKE Compressed	59% 31% 5%	mp_mul rdc_mont fp2mul434_mont
	NewHope (IND-CPA)	34% 25% 9%	ntt montgomery_reduce poly_tomsg	Classic McEliece	83% 6% 4%	gf_mul synd gf_inv
	Round5 Ring SHAKE	86% 5% 3%	ringmul_p probe_cm KeccakF1600_FastLoop_Absorb	NTRU sPrime	69% 23% 5%	uint32_divmod_uint14 int32_mod_uint14 Rq_mult_small

TABLE 9. Top energy-consuming subroutines needed for digital signature operations. Results are shown for the three *least* and *most* energy-consuming candidates. White cell color indicates a lattice scheme, multivariate schemes are in gray, hash-based schemes in purple, and other symmetric schemes in green.

Operation	Three Least Energy-Consuming Algorithms			Three Most Energy-Consuming Algorithms		
	Scheme	Energy	Subroutine	Scheme	Energy	Subroutine
Key Generation	Picnic FS	29% 16% 12%	read mzd_shuffle_128_30 open64	GeMSS	29% 11% 8%	NTL::mul NTL::add NTL::WordVector::operator=
	Dilithium SHAKE	47% 24% 20%	KeccakF1600_StatePermute poly_uniform_eta poly_pointwise_invmontgomery	SPHINCS+ HARAKA s robust	90% 6% 2%	aesenc haraka512_perm haraka256
	MQDSS	69% 15% 8%	KeccakF1600_StatePermute gf31_nrand_schar SHAKE256_squeezeblocks	Rainbow Cyclic	19% 14% 12%	gf16mat_prod batch_2trimat_madd_gf16 batch_trimat_madd_gf16
Signing	Rainbow Classic	56% 17% 11%	gf16mat_gauss_elim_ref gf16mat_prod batch_quad_trimat_eval_gf16	GeMSS	7% 3% 2%	NTL::mul libntl.so.5.0.0+216120 NTL::WV_BlockConstructAlloc
	qTESLA	41% 18% 14%	KeccakF1600_StatePermute sparse_mul16 poly_mul	SPHINCS+ HARAKA s robust	90% 6% 1%	aesenc haraka512_perm haraka256
	Falcon	20% 12% 8%	BerExp falcon_sign_free sampler	Picnic2 FS	32% 16% 11%	KeccakP1600_Permute_24rounds mpc_matrix_mul_nl_part_uint64_128 mpc_matrix_addmul_r_uint64_128
Verification	qTESLA	54% 11% 9%	KeccakF1600_StatePermute sparse_mul32 poly_mul	Picnic2 FS	59% 6% 5%	KeccakP1600_Permute_24rounds transpose_64_64 KeccakP1600_AddBytes
	Falcon	33% 29% 13%	process_block mq_NTT_binary mq_iNTT_binary	SPHINCS+ HARAKA f robust	90% 6% 2%	aesenc haraka512_perm haraka256
	Rainbow Classic	84% 0% 0%	batch_quad_trimat_eval_gf16 gf256v_set_zero libcrypto.so.1.0.0+491721	RedGeMSS	17% 17% 9%	NTL::GF2XFromBytes NTL::mul NTL::add

best for energy consumption metrics for key generation, it is quite energy-inefficient compared to the other algorithms when it comes to signing and verification. In fact, its Picnic2 FS variant consumes the most energy of all algorithms studied when verifying a signature. Based on the rankings, lattice-based algorithms such as Dilithium, qTESLA, and Falcon are among the most energy-efficient for both signing and verification procedures. When comparing the

TABLE 10. The five most energy-efficient algorithms for key encapsulation mechanisms and digital signature operations considering the additional optimized implementation set. White cell color indicates a lattice scheme, code-based schemes are in blue, multivariate schemes are in gray, hash-based schemes in purple, and other symmetric schemes are in green. The number in brackets shows the algorithm's rank in the optimized C implementation set for ease of comparison.

Operation	Rank	Optimized C Implementation			Additional Optimized Implementation		
		Level 1	Level 3	Level 5	Level 1	Level 3	Level 5
Key Generation	1	Kyber (5)	Kyber (3)	Kyber (4)	Picnic (1)	Picnic (1)	Picnic (1)
	2	Round5 (6)	Round5 (5)	NewHope (2)	Dilithium (2)	Dilithium (2)	SPHINCS+ (3)
	3	NewHope (3)	SABER (2)	SABER (3)	SPHINCS+ (5)	SPHINCS+ (5)	qTESLA (2)
	4	SABER (2)	Three Bears (1)	Round5 (6)	MQDSS (3)	MQDSS (3)	LUOV (4)
	5	LAC (4)	LAC (4)	LAC (5)	qTESLA (4)	qTESLA (4)	Rainbow (6)
Encapsulation/ Signing	1	Kyber (5)	Kyber (3)	Kyber (4)	Rainbow (1)	Dilithium (4)	Rainbow (3)
	2	SABER (3)	SABER (2)	NewHope (2)	Dilithium (4)	qTESLA (1)	qTESLA (2)
	3	Round5 (8)	NTRU-HRSS (10)	SABER (3)	qTESLA (2)	Rainbow (3)	LUOV (4)
	4	Classic McEliece (7)	Classic McEliece (5)	Three Bears (1)	LUOV (5)	LUOV (5)	GeMSS (6)
	5	NewHope (4)	Round5 (7)	Classic McEliece (7)	MQDSS (7)	MQDSS (7)	Picnic (5)
Decapsulation/ Verification	1	NewHope (2)	Kyber (3)	NewHope (2)	Rainbow (3)	qTESLA (2)	Rainbow (3)
	2	Kyber (5)	Three Bears (1)	Kyber (4)	qTESLA (1)	Rainbow (4)	qTESLA (2)
	3	Round5 (3)	NTRU-HRSS (7)	Three Bears (1)	Dilithium (4)	Dilithium (3)	GeMSS (7)
	4	Three Bears (1)	Round5 (4)	Round5 (4)	GeMSS (9)	GeMSS (9)	LUOV (5)
	5	SABER (4)	SABER (2)	SABER (3)	SPHINCS+ (6)	LUOV (6)	SPHINCS+ (4)

raw energy measurements, Rainbow is the only scheme that is more energy-efficient when performing verification as opposed to signing across all security levels; however, the lattice-based schemes previously mentioned are still more efficient at verifying signatures than Rainbow when focusing on the optimized C experiment set.

C. ENERGY CONSUMPTION OF SUBROUTINES

A single function may contribute to the majority of the energy consumption. In other cases, the energy profile is more uniformly distributed across the different functions which constitute an operation. In either case, identifying subroutines which consume the greatest energy are important in future works aimed at optimizing algorithm implementations for energy efficiency. In Table 8 and Table 9, the three most energy-consuming subroutines of the tuple of functions comprising key encapsulation mechanisms and digital signature schemes are reported. These are limited to the three least and most energy-consuming algorithms contained within the optimized C implementation set and level 1 security.

1) KEY ENCAPSULATION MECHANISMS

The three most and least energy-efficient schemes from the optimized C KEM experiment set are lattice-based, code-based, and isogeny-based. In general, the most computationally demanding components comprising lattice-based algorithms are those responsible for modular arithmetic, more specifically, matrix multiplication or polynomial multiplication depending on the lattice variant, as well as a discrete sampling component [60]. This is consistent with the subroutine energy consumption results in Table 8 where many functions are concerned with multiplication such as `mac` within the Three Bears implementation, `karatsuba_simple` within SABER, and `ntt` within NewHope. Code-based cryptography is comprised of binary matrix-vector multiplication operations and like many number-theory based public-key schemes, its efficiency is coupled with the efficiency of the underlying field arithmetic operations. For instance, `gf_mul` within Classic McEliece is used to perform field multiplication and makes up a sizeable portion of the decapsulation energy consumption [25]. As isogeny-based cryptography has its roots in elliptic curve cryptography, the most energy-intensive subroutines within SIKE are those associated with modular multiplication, particularly `mp_mul` and `rdc_mont` which are used to perform multi-precision Comba multiplication and efficient Montgomery reduction using Comba [38]. Other functions which consume large portions of energy for a particular operation are more algorithm-dependent. For instance, the majority of the energy required to perform key generation of LEDAcrypt's DFRSL parameter set is due to the `DFR_test` function, a routine responsible for testing the decryption failure rate of a keypair [21]. Furthermore, the energy consumed by the encapsulation operation in FRODO's implementation is almost entirely due to the `Cipher` function, a standalone AES implementation [26]. Likewise, it was found that over 60% of the energy consumed by NTRU Lprime's encapsulation operation is attributed to a function responsible for computing division in time independent of the input operand [32].

Many of the implementations studied use an external Keccak library which provides access to a family of sponge functions used for hash functions and extendable output functions including SHA-3, SHAKE, and cSHAKE [61], [62]. Table 8 shows that the energy consumed by the subroutines related to this family of functions can account for nearly half of the total energy required to perform a cryptographic operation. Although the results in Table 8 are related to the optimized C implementation set, the use of processor-specific compilation flags is not expected to significantly improve the energy profiles. Most PC-class CPUs do not have instruction set extensions for these functions in the same way they support AES and other SHA extensions. Hardware support for the Keccak family of functions would provide additional energy savings for the algorithms under consideration.

2) DIGITAL SIGNATURE SCHEME

By the nature of hash-and-sign digital signature schemes, it is expected that a significant portion of energy consumption will be linked to the hashing algorithm used. It is observed once again that large portions of energy are attributed to the Keccak family of functions (see Table 9). Not all implementations have used a SHA-3 hash function within their signature scheme. For instance, SPHINCS+, a hash-based signature scheme, has submitted variants of their algorithm which use SHAKE, SHA-256, as well as the Haraka family of hash functions. The Haraka variant of the scheme uses the Haraka short-input hash function which is based on AES and is not a NIST-approved hash function [47]. The majority of the energy spent in these operations is due to the function `aesenc` which is required for the AES encryption steps [47]. Due to hardware-accelerated components available on most modern x86 platforms, this metric can be significantly improved, as will be discussed in the next section.

When observing the results related to the GeMSS submission, the three top energy-consuming functions are all supplied by the NTL library. NTL is an external C++ library which supplies data structures and algorithms used by GeMSS for polynomial arithmetic over finite fields [22]. The external library is only used throughout their reference implementation while this arithmetic has been written by the algorithm's authors using assembly optimizations in their additional optimized implementation. Additionally, it is seen that the `batch_quad_trimat_eval_gf16` function, required by Rainbow Classic, constitutes over 80% of the energy required for verification. These functions are concerned with batched matrix operations required for the scheme [46]. Lastly, in submissions such as GeMSS, Picnic, and Rainbow, it is observed that the energy consumed by the top three functions does not make up a very large proportion of the total energy consumption. This suggests that future works targeting energy optimization will need to be distributed across multiple subroutines within the implementation.

D. ON ADDITIONAL OPTIMIZED IMPLEMENTATION IMPROVEMENTS

Unlike in the optimized C implementations, the additional optimized implementations have been compiled to make use of platform-specific instructions and optimizations. One such improvement which has been used by many of the algorithms studied is the use of SIMD instructions such as Advanced Vector Extensions (AVX) and Streaming SIMD Extensions (SSE). These instructions use dedicated large registers ranging in width from 128 to 512 bits (depending on the platform) so that a single operation can be applied to multiple operands simultaneously [63]. These vectorized instructions can achieve software parallelization without the use of multiple physical or logical cores (thread parallelism). Despite the incremental power which may be observed when using these instructions, using AVX can be more energy-efficient due to the increased performance achievable [64], [65]. In some cases, the use of SIMD instructions has been shown to be more energy-efficient than thread parallelization techniques on Intel platforms [66].

In addition to the use of vectorization, modern Intel platforms also provide instruction set extensions to specifically improve the performance of cryptographic-related applications. The Advanced Encryption Standard New Instructions (AES-NI) offer hardware support of AES encryption, decryption and key expansion [67]. AES-NI enables significant acceleration of AES compared to a pure software implementation and can offer increased protection against certain side-channel attacks [67], [68]. This improvement in performance also translates to a reduced energy footprint; in fact, it was shown in [9] that AES-NI can achieve up to 13.5x better speed over a software-based implementation of AES at a 90% reduction in energy consumption. In addition to AES hardware acceleration, Intel platforms also support the carry-less multiplication instruction set (CLMUL) [69]. These instructions allow for efficient polynomial multiplication over binary finite fields, a fundamental operation in both the symmetric and asymmetric cryptography space. The Secure Hash New Instructions (SHA-NI) are another example of a hardware-accelerated x86 extension available from Intel to improve cryptographic arithmetic. These SSE-based instructions can improve both the performance and power consumption of SHA-1 and SHA-256 [70]; however, this extension is not supported on the target platform used in this work.

In Table 10, the algorithms are ranked by energy consumption pertaining to the additional optimized implementation set. Again, a rank of 1 indicates the lowest energy-consuming algorithm and each ranking is visually distinguished by the underlying cryptographic family by color. A number in brackets is provided within each entry; this number represents the scheme's rank in the optimized C implementation for ease of comparison. It should be noted that it was not required for algorithms to have an additional optimized implementation. As a result, the comparison here is limited to those schemes for which one is provided.

Significant improvements in rankings are observed for a number of KEM algorithms between their optimized C and additional optimized implementations. For instance, Round5 [35] through its use of AVX instructions to improve matrix multiplications and AES hardware acceleration has shown between 73% and 80% reduction in energy consumption for level 1 key generation, encapsulation, and decapsulation for its ring variant with no forward error correction (ND 0d). Another candidate whose energy efficiency has improved due to the use of vectorized instructions is NTRU-HRSS [31] where a 88% and 97% savings in energy consumption is observed for encapsulation and decapsulation operations, respectively. Classic McEliece [25] has also made use of vector instructions within its additional optimized implementations, offering both SSE- and AVX-optimized implementations. Through the experiments performed, it is

observed that the AVX-optimized implementation is more energy-efficient than the SSE variant; a 63% improvement in energy consumption is seen for the AVX-optimized encapsulation operation whereas only a 57% improvement is seen for the same algorithm using SSE-based optimizations. This is also the case for NTS-KEM [33], another code-based scheme which has offered both AVX- and SSE-optimized implementations.

Kyber tops almost all categories being evaluated. Their algorithm has been designed with an additional "90s" variant which has been designed specifically to make use of symmetric primitives such as AES and SHA-2 which can make use of hardware-accelerated instructions [28]. While their "90s" variant was more energy intensive than the version of Kyber reliant on the SHAKE family of functions in the optimized implementation experiment set, the "90s" variant is between 36% and 42% more energy-efficient than the SHAKE variant across level 1 key generation, encapsulation, and decapsulation when considering the additional optimized implementations.

When considering the additional optimized digital signature algorithms in Table 10, most of the ranked positions are now occupied by multivariate-based signature schemes as compared to the optimized C implementation set which was mostly occupied by lattice-based algorithms (see Table 7). Notably, the Rainbow algorithm, which was mentioned to be the only scheme that was consistently more energy-efficient at performing verification as opposed to signing, tops the rankings of both signing and verification operations at security levels 1 and 5. The hash-based SPHINCS+ implementation has shown one of the largest improvements in energy consumption which can be partially attributed to its use of symmetric primitives other than the Keccak family of functions [47]; it has made use of the Haraka hash function in its level 1 target which has been specifically designed to be very efficient on platforms supporting AES-NI instructions [71]. SPHINCS+'s f simple parameter set using the Haraka hash function has achieved over 97% energy consumption reduction across all three functions comprising the digital signature algorithm. GeMSS' schemes have also shown improvements in energy efficiency resulting from their use of x86 extensions including `PCLMULQDQ`, part of the carry-less multiplication instruction set, to improve multiplication of binary polynomials which is a critical operation in their algorithm [22]. The energy consumption of all level 1 parameter sets has decreased by over 90% across key generation, signing, and verification operations between the additional optimized implementations and the optimized C counterparts.

VI. CONCLUDING REMARKS

A. ADDITIONAL REMARKS

Designing energy-efficient software has gained a lot of interest in recent years; however, designing software from an energy efficiency standpoint is often complex due to the number of independent factors involved. As discussed, all experiments have been performed at a frequency of 3.4GHz with Turbo Boost and Hyperthreading disabled with only a single core active. Modern PCs have frequency scaling governors which can dynamically change the operating frequency of the CPU depending on the load experienced [55]. Power dissipation is directly proportional to a system's operating frequency. As the computer setup changes, it is expected that the power will vary. This would naturally have an effect on the energy consumption of a particular algorithm. Furthermore, energy is the power dissipation over a period of time. As a result, designing for reduced execution time *can* have a direct effect on the energy consumed; however, this is not always the case and is highly dependent on the means by which the speedup is achieved [72]. This is demonstrated in the results obtained in this work where the power consumption of the additional optimized implementation is almost always larger than that of the optimized C implementations, despite using the same computer setup. Although there is not much documentation regarding the energy consumption of specific operations, there have been studies which show that the energy consumption between different types of instructions varies and may influence the total energy of software [73].

It is also emphasized that the results in this paper have been categorized by energy efficiency for each security level only and separated by the operation performed. This is due to the fact that applications will have different requirements of the cryptographic scheme they wish to deploy. In the context of digital signatures, 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 [74]. In contrast, applications such as public-key certification where a single message is signed once and verified by the masses are ideally designed to have an efficient verification procedure [75]. In the case of KEM schemes, algorithms achieving IND-CCA security can make use of static keys. In contrast, those only achieving IND-CPA security need to create a new keypair for each iteration. As a result, the additional energy required by IND-CCA and IND-CCA2 algorithms for keypair generation may be justified. Thus, the focus on which underlying functions should be optimized from an energy efficiency standpoint will shift based on the application for which the scheme is deployed.

In this work, only the the package and DRAM energy are considered. Our methodology can be expanded to study the energy of other software-based ciphers including lightweight cryptographic schemes such as [11], [15], [16], [76]. Additional energy metrics which may be important include communication energy costs. This energy consumption would largely depend on the bits of transmitted/received data. As a result, the large key and ciphertext lengths typical of lattice and code-based schemes may be of greater importance. For instance, the isogeny-based scheme consumes more energy for computation than most algorithms under consideration.

Nonetheless, it may be more desirable based on transmission energy alone, as noted by the author in [18], as it boasts very short ciphertext and public-key sizes.

B. CONCLUSION

In this work, the energy consumed by PQC algorithms is studied by measuring the energy required of NIST's PQC Round 2 candidates. Results have been categorized by cryptographic function and proposed security level. Algorithms have been ranked based on their energy consumption to demonstrate which schemes are most energy-efficient. Further insights are shown into the energy consumption profile of a select number of candidates to demonstrate which subroutines contribute most to the overall energy consumed. The results show that lattice-based schemes tend to be very energy-efficient in practice. When considering signing operations, it is observed that multivariate-based schemes are very competitive with their lattice-based counterparts, a trend that is even more evident when platform-specific optimized instructions are used. It is important to note that this ranking only provides one metric of evaluation; a holistic approach should be used when determining which algorithm best suits an application. It is hoped that the findings displayed here can identify potential avenues for future optimizations.

APPENDIX

The full set of results are reported here and categorized based on security levels 1, 3, and 5. Schemes targeting levels 2 and 4 (marked $*$) have been consolidated into levels 1 and 3, respectively. For each level, the most energy-consuming algorithm is distinguished in red text, while the least is marked in green. Timing results are reported in milliseconds (ms), energy in milliJoules (mJ), and average power in Watts (W). Table 11 to Table 13 reports energy of the KEM algorithms and Table 14 through Table 16 displays the data for digital signature operations of the optimized C implementations. Table 17 to 19 reports energy of the KEM algorithms and Table 20 to Table 22 shows the data for digital signature operations for the additional optimized implementations.

TABLE 11. Energy consumed by key encapsulation mechanisms for key generation operations in optimized C implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CCA	16.368	0.242	3.965	16.193	0.705	11.420	16.159	1.501	24.253
BIKE1	IND-CPA	16.224	0.203	3.289	16.370	0.504	8.248	15.831	0.885	14.010
BIKE2	IND-CCA	15.996	0.443	7.084	15.974	1.352	21.598	15.908	3.028	48.170
BIKE2	IND-CPA	15.929	0.377	5.999	16.065	1.004	16.132	15.522	1.751	27.182
BIKE3	IND-CCA	15.985	0.161	2.577	16.114	0.473	7.618	16.186	0.932	15.087
BIKE3	IND-CPA	16.053	0.124	1.992	16.129	0.323	5.209	16.240	0.678	11.010
Classic McEliece	IND-CCA2	16.798	159.096	2672.428	17.006	465.209	7911.274	17.508	907.989	15896.689
FRODO AES	IND-CCA	15.317	13.754	210.663	15.221	32.089	488.414	15.151	60.200	912.111
FRODO SHAKE	IND-CCA	15.823	3.026	47.882	15.817	6.732	106.484	15.814	12.123	191.712
hqc-1	IND-CCA2	16.315	0.350	5.710	15.981	0.847	13.540	15.513	1.316	20.414
hqc-2	IND-CCA2	-	-	-	15.915	0.902	14.359	16.175	1.606	25.977
hqc-3	IND-CCA2	-	-	-	-	-	-	16.232	1.767	28.689
Kyber	IND-CCA2	16.207	0.039	0.639	16.099	0.069	1.114	16.153	0.106	1.709
Kyber-90s	IND-CCA2	16.098	0.063	1.015	16.037	0.117	1.883	15.930	0.189	3.007
LAC	IND-CCA	16.717	0.036	0.596	16.938	0.090	1.517	16.872	0.102	1.716
LEDAcrypt DFR64	IND-CCA2	15.529	397.097	6166.531	15.661	1268.899	19871.635	15.763	3702.345	58358.835
LEDAcrypt DFRSL	IND-CCA2	15.456	616.768	9532.513	15.784	2220.525	35048.529	15.906	6083.541	96766.918
LEDAcrypt N02	IND-CPA	16.093	12.125	195.132	16.011	34.913	558.986	15.978	72.932	1165.273
LEDAcrypt N03	IND-CPA	16.014	4.069	65.160	15.998	14.455	231.245	15.969	41.435	661.686
LEDAcrypt N04	IND-CPA	16.008	3.864	61.858	15.980	13.216	211.182	15.968	30.780	491.489
NewHope	IND-CCA	16.077	0.039	0.630	-	-	-	16.119	0.075	1.214
NewHope	IND-CPA	16.184	0.034	0.548	-	-	-	16.013	0.068	1.094
NTRU LPrime	IND-CCA2	14.709	6.855*	100.821*	14.690	9.150	134.408	-	-	-
NTRU sPrime	IND-CCA2	14.845	60.916*	904.314*	14.859	80.195	1191.616	-	-	-
NTRU-HPS	IND-CCA2	15.774	3.531	55.700	15.686	6.116	95.925	15.709	8.896	139.747
NTRU-HRSS	IND-CCA2	-	-	-	15.672	6.543	102.545	-	-	-
NTS-KEM	IND-CCA2	16.286	16.965	276.292	16.555	51.044	845.052	16.425	95.353	1566.209
ROLLO-I	IND-CPA	16.308	0.776	12.657	15.330	3.463	53.081	15.804	1.543	24.392
ROLLO-III	IND-CPA	16.203	0.159	2.579	16.235	0.209	3.392	16.165	0.393	6.361
Round5 N1 0d AES	IND-CPA	16.530	1.584	26.190	16.428	3.861	63.434	16.470	7.194	118.482
Round5 N1 0d SHAKE	IND-CPA	16.597	1.598	26.522	16.513	3.880	64.072	16.301	7.765	126.571
Round5 ND 0d AES	IND-CPA	16.097	0.058	0.932	16.145	0.115	1.862	16.132	0.238	3.833
Round5 ND 0d SHAKE	IND-CPA	15.818	0.059	0.928	16.070	0.124	1.985	16.014	0.252	4.036
Round5 ND 5d AES	IND-CPA	16.139	0.045	0.734	16.064	0.198	3.174	16.003	0.259	4.151
Round5 ND 5d SHAKE	IND-CPA	16.094	0.045	0.719	15.916	0.191	3.033	16.033	0.277	4.435
RQC	IND-CCA2	16.010	0.284	4.548	16.223	0.476	7.719	16.241	0.941	15.279
SABER	IND-CCA	16.151	0.031	0.504	16.069	0.059	0.942	16.029	0.095	1.522
SIKE	IND-CCA	14.979	17.620	263.930	14.712	50.606	744.496	14.794	92.303	1365.496
SIKE Compressed	IND-CCA	14.989	42.890	642.895	14.750	126.725	1869.225	14.782	217.025	3208.110
Three Bears	IND-CCA	16.924	0.021*	0.349*	16.802	0.036*	0.599*	16.262	0.057	0.926
Three Bears	IND-CPA	16.805	0.020*	0.341*	16.521	0.037*	0.610*	16.441	0.059	0.964

TABLE 12. Energy consumed by key encapsulation mechanisms for encapsulation operations in optimized C implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CCA	16.440	0.302	4.957	16.069	0.902	14.496	15.974	1.980	31.621
BIKE1	IND-CPA	16.187	0.247	4.003	16.162	0.679	10.982	15.619	1.119	17.480
BIKE2	IND-CCA	16.344	0.118	1.923	16.351	0.315	5.155	16.396	0.716	11.746
BIKE2	IND-CPA	16.283	0.103	1.680	16.360	0.258	4.228	15.978	0.430	6.877
BIKE3	IND-CCA	16.350	0.226	3.689	16.388	0.745	12.203	16.176	1.570	25.403
BIKE3	IND-CPA	16.270	0.209	3.404	16.360	0.540	8.836	16.205	1.207	19.557
Classic McEliece	IND-CCA2	16.372	0.063	1.026	16.293	0.120	1.963	15.977	0.212	3.385
FRODO AES	IND-CCA	15.226	14.038	213.739	15.231	32.405	493.542	15.112	61.117	923.617
FRODO SHAKE	IND-CCA	15.929	3.278	52.212	15.943	7.293	116.269	15.902	13.030	207.194
hqc-1	IND-CCA2	16.305	0.669	10.911	16.393	1.630	26.714	15.870	2.507	39.785
hqc-2	IND-CCA2	-	-	-	16.424	1.738	28.553	16.220	3.040	49.313
hqc-3	IND-CCA2	-	-	-	-	-	-	16.223	3.456	56.070
Kyber	IND-CCA2	16.223	0.051	0.826	16.164	0.083	1.334	16.145	0.123	1.981
Kyber-90s	IND-CCA2	16.116	0.076	1.220	16.049	0.133	2.128	15.965	0.207	3.306
LAC	IND-CCA	16.652	0.059	0.977	16.895	0.123	2.071	16.881	0.173	2.913
LEDAcrypt DFR64	IND-CCA2	16.145	2.543	41.050	16.324	4.517	73.736	16.485	7.835	129.153
LEDAcrypt DFRSL	IND-CCA2	16.288	3.920	63.846	16.284	9.702	157.993	16.382	18.196	298.089
LEDAcrypt N02	IND-CPA	16.233	0.679	11.028	16.280	1.366	22.236	16.171	2.674	43.234
LEDAcrypt N03	IND-CPA	16.232	0.550	8.925	16.270	1.405	22.855	16.292	2.861	46.607
LEDAcrypt N04	IND-CPA	16.154	0.699	11.288	16.207	1.827	29.611	16.240	3.432	55.738
NewHope CCA	IND-CCA	15.999	0.058	0.936	-	-	-	16.052	0.114	1.826
NewHope CPA	IND-CPA	16.236	0.049	0.793	-	-	-	16.014	0.100	1.604
NTRU LPrime	IND-CCA2	14.529	12.276*	178.364*	14.485	16.626	240.820	-	-	-
NTRU sPrime	IND-CCA2	14.654	6.863*	100.571*	14.653	9.167	134.326	-	-	-
NTRU-HPS	IND-CCA2	16.199	0.227	3.681	16.205	0.372	6.025	16.259	0.521	8.473
NTRU-HRSS	IND-CCA2	-	-	-	15.384	0.345	5.315	-	-	-
NTS-KEM	IND-CCA2	16.978	0.029	0.489	15.167	0.122	1.851	15.002	0.188	2.817
ROLLO-I	IND-CPA	16.217	0.172	2.788	15.875	0.272	4.312	15.941	0.329	5.247
ROLLO-III	IND-CPA	16.185	0.346	5.606	16.188	0.459	7.434	16.151	0.835	13.488
Round5 N1 0d AES	IND-CPA	15.726	1.605	25.239	15.625	3.962	61.906	15.647	7.381	115.489
Round5 N1 0d SHAKE	IND-CPA	15.819	1.619	25.609	15.657	3.986	62.405	15.571	7.469	116.296
Round5 ND 0d AES	IND-CPA	16.076	0.106	1.701	16.170	0.216	3.492	16.168	0.451	7.294
Round5 ND 0d SHAKE	IND-CPA	15.987	0.106	1.688	16.062	0.228	3.660	16.036	0.469	7.515
Round5 ND 5d AES	IND-CPA	16.148	0.084	1.355	16.129	0.367	5.916	16.033	0.486	7.790
Round5 ND 5d SHAKE	IND-CPA	16.071	0.083	1.331	15.952	0.349	5.573	16.080	0.507	8.158
RQC	IND-CCA2	16.060	0.609	9.786	16.255	0.995	16.169	16.158	1.945	31.424
SABER	IND-CCA	15.919	0.040	0.635	16.071	0.073	1.167	15.954	0.116	1.843
SIKE	IND-CCA	14.960	28.787	430.651	14.721	93.035	1369.553	14.796	149.615	2213.731
SIKE Compressed	IND-CCA	14.986	52.704	789.801	14.756	150.387	2219.076	14.785	273.976	4050.767
Three Bears	IND-CCA	16.966	0.026*	0.440*	16.767	0.042*	0.706*	16.527	0.065	1.071
Three Bears	IND-CPA	16.913	0.026*	0.439*	16.549	0.044*	0.728*	16.425	0.067	1.102

TABLE 13. Energy consumed by key encapsulation mechanisms for decapsulation operations in optimized C implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CCA	16.343	1.879	30.709	16.179	4.624	74.808	16.310	10.156	165.647
BIKE1	IND-CPA	16.091	0.960	15.447	16.012	2.640	42.277	15.652	6.157	96.363
BIKE2	IND-CCA	16.090	1.596	25.687	16.082	3.784	60.850	16.253	8.211	133.451
BIKE2	IND-CPA	15.921	0.882	14.046	15.925	2.412	38.417	15.519	5.772	89.574
BIKE3	IND-CCA	16.210	1.892	30.673	16.211	4.689	76.006	16.180	9.870	159.694
BIKE3	IND-CPA	15.639	1.186	18.549	15.868	2.642	41.929	15.885	5.979	94.980
Classic McEliece	IND-CCA2	14.351	19.038	273.216	14.169	45.418	643.523	14.038	87.007	1221.374
FRODO AES	IND-CCA	15.229	14.019	213.495	15.219	32.575	495.762	15.120	61.044	922.976
FRODO SHAKE	IND-CCA	15.938	3.253	51.839	15.924	7.232	115.168	15.913	12.949	206.064
hqc-1	IND-CCA2	16.251	1.055	17.137	16.217	2.486	40.314	15.723	3.769	59.266
hqc-2	IND-CCA2	-	-	-	16.246	2.628	42.689	16.188	4.565	73.900
hqc-3	IND-CCA2	-	-	-	-	-	-	16.198	5.194	84.125
Kyber	IND-CCA2	16.216	0.060	0.966	16.195	0.094	1.517	16.166	0.136	2.204
Kyber-90s	IND-CCA2	16.137	0.083	1.341	16.116	0.142	2.289	15.995	0.220	3.514
LAC	IND-CCA	16.790	0.078	1.314	16.961	0.200	3.389	16.956	0.252	4.273
LEDAcrypt DFR64	IND-CCA2	15.271	3.597	54.931	15.248	8.169	124.558	15.266	15.810	241.357
LEDAcrypt DFRSL	IND-CCA2	15.243	4.416	67.315	15.316	10.448	160.025	15.283	20.908	319.551
LEDAcrypt N02	IND-CPA	15.200	3.262	49.581	15.180	9.314	141.387	15.122	15.089	228.181
LEDAcrypt N03	IND-CPA	15.360	3.813	58.567	15.231	9.209	140.264	15.105	20.033	302.598
LEDAcrypt N04	IND-CPA	15.342	5.481	84.082	15.220	14.559	221.598	15.315	21.120	323.446
NewHope CCA	IND-CCA	16.023	0.065	1.037	-	-	-	16.080	0.128	2.058
NewHope CPA	IND-CPA	16.030	0.012	0.186	-	-	-	16.186	0.024	0.392

TABLE 13. *(Continued.)* Energy consumed by key encapsulation mechanisms for decapsulation operations in optimized C implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
NTRU LPrime	IND-CCA2	14.511	18.388*	266.832*	14.501	24.901	361.092	-	-	-
NTRU sPrime	IND-CCA2	14.458	18.538*	268.019*	14.456	25.139	363.406	-	-	-
NTRU-HPS	IND-CCA2	16.398	0.489	8.026	16.420	0.852	13.984	16.393	1.242	20.359
NTRU-HRSS	IND-CCA2	-	-	-	16.096	0.904	14.546	-	-	-
NTS-KEM	IND-CCA2	16.045	0.217	3.480	16.104	0.412	6.633	15.867	0.880	13.967
ROLLO-I	IND-CPA	16.025	0.552	8.849	15.582	1.410	21.977	15.721	1.638	25.751
ROLLO-III	IND-CPA	16.034	0.531	8.517	15.781	1.017	16.053	15.688	1.774	27.837
Round5 N1 0d AES	IND-CPA	15.593	0.090	1.410	15.814	0.140	2.220	15.626	0.460	7.182
Round5 N1 0d SHAKE	IND-CPA	15.648	0.096	1.504	15.810	0.147	2.330	15.586	0.482	7.507
Round5 ND 0d AES	IND-CPA	15.966	0.055	0.878	16.050	0.113	1.814	16.047	0.234	3.751
Round5 ND 0d SHAKE	IND-CPA	15.786	0.053	0.840	15.934	0.115	1.833	15.924	0.238	3.795
Round5 ND 5d AES	IND-CPA	16.141	0.043	0.694	16.046	0.189	3.026	15.986	0.247	3.954
Round5 ND 5d SHAKE	IND-CPA	15.922	0.041	0.660	15.856	0.178	2.819	16.015	0.259	4.154
RQC	IND-CCA2	16.085	2.894	46.554	16.227	6.405	103.943	16.090	12.880	207.245
SABER	IND-CCA	16.056	0.044	0.704	16.081	0.079	1.271	16.031	0.124	1.987
SIKE	IND-CCA	14.965	30.734	459.935	14.723	93.665	1379.025	14.804	160.872	2381.510
SIKE Compressed	IND-CCA	14.992	48.995	734.548	14.752	144.762	2135.582	14.785	253.631	3749.923
Three Bears	IND-CCA	16.903	0.039*	0.665*	16.642	0.060*	1.004*	16.411	0.089	1.459
Three Bears	IND-CPA	16.819	0.009*	0.157*	16.406	0.013*	0.205*	16.201	0.016	0.253

TABLE 14. Energy consumed by digital signature algorithms for key generation operations in optimized C implementations.

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.392	0.171	2.632	15.349	0.370	5.679	-	-	-
Dilithium SHAKE	15.738	0.080	1.259	15.682	0.157	2.462	-	-	-
Falcon	14.804	6.388	94.570	14.194	11.829	167.897	14.470	18.613	269.325
BlueGeMSS	16.218	492.553	7988.135	16.060	1879.161	30178.812	15.205	6287.946	95606.548
GeMSS	16.175	653.995	10578.378	16.022	2434.249	39001.154	15.522	6811.709	105732.212
RedGeMSS	16.110	359.028	5783.957	15.982	1418.704	22673.791	15.190	4670.075	70940.717
Luov Large Chacha	14.952	2.384*	35.645*	14.874	7.300*	108.587*	15.338	17.557	269.293
Luov Large Keccak	15.333	2.626*	40.256*	15.240	8.011*	122.084*	15.633	19.206	300.249
Luov Small Chacha	14.851	4.143*	61.534*	15.288	16.033*	245.110*	15.340	27.894	427.898
Luov Small Keccak	15.242	4.739*	72.228*	15.507	17.638*	273.516*	15.454	30.717	474.701
MQDSS	15.220	0.325*	4.943*	15.177	0.754*	11.437*	-	-	-
Picnic FS	15.635	0.011	0.175	16.250	0.016	0.260	15.585	0.021	0.334
Picnic UR	16.000	0.011	0.176	15.757	0.016	0.255	15.566	0.021	0.332
Picnic2 FS	16.455	0.011	0.181	16.375	0.016	0.262	15.546	0.021	0.326
qTESLA	15.286	0.346	5.289	15.385	0.993	15.277	15.569	5.231	81.449
qTESLA-p	15.517	1.624	25.199	15.538	8.122	126.203	-	-	-
qTESLA-s	15.721	0.348	5.471	15.775	0.993	15.665	15.584	5.259	81.949
qTESLA-size	-	-	-	-	-	-	15.508	7.604	117.920
qTESLA-size-s	-	-	-	-	-	-	15.489	7.591	117.579
Rainbow Classic	15.857	7.380	117.024	15.109	79.606	1202.750	15.381	168.168	2586.659
Rainbow Compressed Cyclic	15.817	8.202	129.734	14.846	119.708	1777.172	14.902	298.117	4442.438
Rainbow Cyclic	15.833	8.222	130.183	14.903	119.703	1783.984	14.950	297.763	4451.661
SPHINCS+ Haraka f robust	15.154	7.470	113.199	-	-	-	-	-	-
SPHINCS+ Haraka f simple	15.737	5.014	78.904	-	-	-	-	-	-
SPHINCS+ Haraka s robust	15.112	237.476	3588.636	-	-	-	-	-	-
SPHINCS+ Haraka s simple	15.692	158.852	2492.696	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	15.192	3.113	47.294	15.200	4.601	69.933	15.220	17.596	267.808
SPHINCS+ SHA-2 f simple	15.787	1.612	25.448	15.770	2.373	37.422	15.229	6.154	93.715
SPHINCS+ SHA-2 s robust	15.193	98.942	1503.235	15.189	146.819	2230.028	15.207	282.332	4293.372
SPHINCS+ SHA-2 s simple	15.767	51.468	811.515	15.735	75.065	1181.156	15.202	99.080	1506.233
SPHINCS+ SHAKE f robust	15.566	5.262	81.910	15.554	7.746	120.483	15.567	20.612	320.860
SPHINCS+ SHAKE f simple	15.940	2.752	43.867	16.055	4.063	65.230	15.605	10.735	167.516
SPHINCS+ SHAKE s robust	15.454	168.425	2602.785	15.511	247.740	3842.576	15.593	327.278	5103.147
SPHINCS+ SHAKE s simple	15.969	88.445	1412.365	16.052	129.667	2081.411	15.629	170.671	2667.452

TABLE 15. Energy consumed by digital signature algorithms for signing operations in optimized C implementations.

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.868	0.591	9.384	15.684	0.882	13.830	-	-	-
Dilithium SHAKE	16.117	0.419	6.745	16.003	0.571	9.141	-	-	-
Falcon	16.111	0.237	3.812	16.061	0.357	5.726	16.197	0.464	7.521
BlueGeMSS	16.078	614.812	9885.020	15.879	1928.981	30630.647	15.734	2397.283	37718.637
GeMSS	16.057	4521.190	72597.049	15.835	15477.687	245088.622	15.721	26767.546	420814.712
RedGeMSS	15.907	13.137	208.971	15.885	32.750	520.253	15.807	44.508	703.542

TABLE 15. *(Continued.)* **Energy consumed by digital signature algorithms for signing operations in optimized C implementations.**

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Luov Large Chacha	15.649	7.706*	120.590*	15.599	23.298*	363.442*	16.047	52.035	834.984
Luov Large Keccak	15.817	7.974*	126.124*	15.673	23.832*	373.516*	15.943	53.988	860.738
Luov Small Chacha	15.603	1.195*	18.648*	15.619	3.051*	47.659*	15.726	5.169	81.280
Luov Small Keccak	16.263	1.745*	28.386*	16.291	4.575*	74.526*	16.277	7.601	123.715
MQDSS	15.225	7.940*	120.886*	15.197	25.467*	387.033*	-	-	-
Picnic FS	15.727	3.497	55.006	15.812	8.564	135.410	15.953	15.339	244.694
Picnic UR	15.919	4.423	70.417	15.892	11.116	176.653	15.881	18.926	300.573
Picnic2 FS	16.414	122.254	2006.729	16.314	358.787	5853.265	16.259	750.020	12194.222
qTESLA	15.648	0.180	2.824	15.954	0.254	4.053	16.271	0.775	12.613
qTESLA-p	15.620	1.034	16.148	15.893	2.821	44.830	-	-	-
qTESLA-s	16.021	0.187	3.000	16.238	0.272	4.420	16.312	0.816	13.303
qTESLA-size	-	-	-	-	-	-	14.772	1.686	24.912
qTESLA-size-s	-	-	-	-	-	-	14.708	1.788	26.300
Rainbow Classic	15.651	0.115	1.795	14.948	0.599	8.957	15.552	1.066	16.579
Rainbow Compressed Cyclic	15.861	4.527	71.803	14.954	62.889	940.409	15.012	153.739	2307.892
Rainbow Cyclic	16.667	0.112	1.871	14.908	0.877	13.067	15.141	1.789	27.088
SPHINCS+ Haraka f robust	15.175	275.340	4178.272	-	-	-	-	-	-
SPHINCS+ Haraka f simple	15.497	181.492	2812.645	-	-	-	-	-	-
SPHINCS+ Haraka s robust	15.131	4410.841	66739.039	-	-	-	-	-	-
SPHINCS+ Haraka s simple	15.385	2890.033	44463.508	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	15.167	93.494	1417.981	15.179	127.912	1941.518	15.304	412.116	6306.917
SPHINCS+ SHA-2 f simple	15.587	50.952	794.194	15.685	66.509	1043.158	15.580	150.468	2344.314
SPHINCS+ SHA-2 s robust	15.192	1376.377	20909.488	15.186	3410.745	51794.388	15.584	3487.577	54351.194
SPHINCS+ SHA-2 s simple	15.515	764.511	11861.73	15.361	1872.013	28756.168	15.478	1292.802	20009.673
SPHINCS+ SHAKE f robust	15.586	158.587	2471.673	15.583	207.276	3229.908	15.672	456.479	7153.942
SPHINCS+ SHAKE f simple	15.969	87.223	1392.845	16.048	111.011	1781.552	16.013	246.206	3942.396
SPHINCS+ SHAKE s robust	15.782	2351.608	37114.162	15.947	4824.091	76929.577	15.990	3743.929	59863.583
SPHINCS+ SHAKE s simple	16.018	1319.814	21141.276	16.082	2756.102	44323.183	16.027	2047.039	32808.307

TABLE 16. **Energy consumed by digital signature algorithms for verification operations in optimized C implementations.**

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.575	0.176	2.745	15.408	0.370	5.705	-	-	-
Dilithium SHAKE	15.815	0.100	1.585	15.709	0.182	2.856	-	-	-
Falcon	15.407	0.045	0.687	15.144	0.069	1.047	15.196	0.081	1.234
BlueGeMSS	16.026	8.657	138.747	15.988	23.567	376.777	15.691	48.533	761.557
GeMSS	15.971	8.473	135.319	15.839	23.894	378.448	15.964	39.979	638.228
RedGeMSS	16.022	8.843	141.683	16.066	22.845	367.029	15.700	49.880	783.089
Luov Large Chacha	15.210	5.644*	85.838*	15.066	15.721*	236.859*	15.247	28.333	431.999
Luov Large Keccak	15.556	5.722*	89.015*	15.216	16.449*	250.288*	15.266	30.096	459.432
Luov Small Chacha	15.234	0.893*	13.606*	15.207	2.263*	34.409*	15.155	3.983	60.358
Luov Small Keccak	16.204	1.415*	22.924*	16.247	3.784*	61.484*	16.034	6.410	102.780
MQDSS	15.674	5.812*	91.097*	15.279	18.951*	289.563*	-	-	-
Picnic FS	15.709	2.843	44.668	15.817	7.200	113.883	15.978	13.148	210.077
Picnic UR	15.924	3.571	56.871	15.883	9.141	145.191	15.868	15.947	253.045
Picnic2 FS	16.581	57.325	950.495	16.537	132.861	2197.184	16.487	236.996	3907.255
qTESLA	15.522	0.040	0.623	15.855	0.075	1.194	16.096	0.159	2.554
qTESLA-p	15.787	0.258	4.072	16.173	0.713	11.532	-	-	-
qTESLA-s	15.786	0.040	0.633	15.957	0.076	1.212	15.971	0.160	2.548
qTESLA-size	-	-	-	-	-	-	14.815	0.310	4.597
qTESLA-size-s	-	-	-	-	-	-	14.796	0.311	4.599
Rainbow Classic	12.586	0.060	0.755	15.357	0.524	8.051	15.486	1.067	16.529
Rainbow Compressed Cyclic	15.922	1.467	23.362	16.424	9.034	148.371	16.409	22.538	369.816
Rainbow Cyclic	15.959	1.472	23.485	16.461	9.205	151.521	16.441	22.871	376.026
SPHINCS+ Haraka f robust	15.158	11.619	176.115	-	-	-	-	-	-
SPHINCS+ Haraka f simple	15.720	7.560	118.849	-	-	-	-	-	-
SPHINCS+ Haraka s robust	15.133	5.165	78.169	-	-	-	-	-	-
SPHINCS+ Haraka s simple	15.673	3.368	52.781	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	15.158	4.219	63.950	15.183	6.964	105.739	15.469	10.468	161.929
SPHINCS+ SHA-2 f simple	15.769	2.102	33.146	15.763	3.418	53.883	15.736	3.469	54.587
SPHINCS+ SHA-2 s robust	15.187	1.743	26.479	15.192	2.775	42.153	15.677	5.360	84.028
SPHINCS+ SHA-2 s simple	15.754	0.882	13.902	15.725	1.368	21.518	15.641	1.775	27.768
SPHINCS+ SHAKE f robust	15.623	6.991	109.214	15.925	11.140	177.414	15.951	11.300	180.239
SPHINCS+ SHAKE f simple	15.930	3.560	56.714	16.036	5.662	90.798	16.126	5.742	92.605
SPHINCS+ SHAKE s robust	15.996	2.903	46.435	15.967	4.268	68.142	16.080	5.585	89.806
SPHINCS+ SHAKE s simple	16.010	1.494	23.916	16.046	2.171	34.835	16.118	2.834	45.685

TABLE 17. Energy consumed by key encapsulation mechanisms for key generation operations in additional optimized implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CPA	15.580	0.033	0.508	15.614	0.090	1.407	15.704	0.090	1.412
BIKE2	IND-CPA	16.611	0.372	6.183	16.430	1.009	16.581	15.715	1.761	27.669
BIKE3	IND-CPA	16.013	0.098	1.576	16.017	0.278	4.450	16.118	0.607	9.789
Classic McEliece AVX	IND-CCA2	16.354	50.749	829.949	17.863	121.938	2178.133	18.013	286.316	5157.370
Classic McEliece SSE	IND-CCA2	16.281	111.460	1814.652	16.913	285.050	4821.042	17.248	455.494	7856.279
FRODO AES	IND-CCA	16.984	0.404	6.858	16.945	0.870	14.736	16.906	1.476	24.945
FRODO SHAKE	IND-CCA	18.073	1.190	21.505	18.131	2.554	46.305	18.051	4.545	82.039
hqc-1	IND-CCA2	16.044	0.096	1.547	16.063	0.177	2.844	16.276	0.267	4.343
hqc-2	IND-CCA2	-	-	-	16.263	0.185	3.010	16.395	0.285	4.668
hqc-3	IND-CCA2	-	-	-	-	-	-	16.276	0.291	4.737
Kyber	IND-CCA2	16.918	0.011	0.184	16.931	0.019	0.317	17.292	0.026	0.446
Kyber-90s	IND-CCA2	16.177	0.007	0.117	16.036	0.010	0.159	16.174	0.013	0.214
LAC	IND-CCA	16.704	0.019	0.324	16.795	0.043	0.728	16.660	0.054	0.898
LEDAcrypt DFR64	IND-CCA2	15.670	325.242	5096.631	15.714	1176.149	18482.537	16.087	3333.106	53619.488
LEDAcrypt DFRSL	IND-CCA2	15.847	464.128	7355.061	15.674	1845.987	28934.210	15.951	5247.737	83706.301
LEDAcrypt N02	IND-CPA	15.524	1.274	19.779	15.420	4.204	64.831	15.424	9.215	142.136
LEDAcrypt N03	IND-CPA	15.486	0.548	8.488	15.732	1.857	29.211	15.425	5.669	87.435
LEDAcrypt N04	IND-CPA	15.699	0.862	13.528	15.871	2.577	40.900	15.712	5.744	90.240
NewHope	IND-CCA	16.717	0.021	0.355	-	-	-	16.787	0.038	0.641
NewHope	IND-CPA	16.841	0.017	0.284	-	-	-	16.866	0.031	0.525
NTRU-HRSS	IND-CCA2	-	-	-	17.011	0.121	2.060	-	-	-
NTS-KEM AVX	IND-CCA2	16.116	16.196	261.013	16.321	48.193	786.550	16.172	88.548	1431.984
NTS-KEM SSE	IND-CCA2	15.940	17.090	272.424	16.267	50.951	828.806	16.181	95.087	1538.590
Round5 N1 0d AES	IND-CPA	16.514	0.140	2.317	16.747	0.252	4.215	16.538	0.652	10.776
Round5 N1 0d SHAKE	IND-CPA	16.467	0.153	2.520	16.822	0.267	4.492	16.619	0.704	11.700
Round5 ND 0d AES	IND-CPA	16.387	0.014	0.231	16.312	0.050	0.811	16.539	0.058	0.964
Round5 ND 0d SHAKE	IND-CPA	16.377	0.016	0.255	16.355	0.053	0.870	16.644	0.064	1.059
Round5 ND 5d AES	IND-CPA	16.175	0.019	0.314	16.429	0.030	0.486	16.362	0.054	0.886
Round5 ND 5d SHAKE	IND-CPA	16.015	0.021	0.329	16.109	0.033	0.550	16.022	0.058	0.959
SABER	IND-CCA	17.016	0.018	0.304	16.930	0.029	0.496	17.009	0.044	0.754
SIKE	IND-CCA	16.936	1.932	32.719	16.997	4.576	77.782	16.907	7.733	130.739
SIKE Compressed	IND-CCA	16.740	4.957	82.971	16.909	12.010	203.071	17.017	18.765	319.324
Three Bears	IND-CCA	16.489	0.022*	0.366*	16.703	0.040*	0.673*	16.666	0.065	1.082
Three Bears	IND-CPA	16.250	0.022*	0.363*	16.396	0.041*	0.674*	16.881	0.067	1.125

TABLE 18. Energy consumed by key encapsulation mechanisms for encapsulation operations in additional optimized implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CPA	15.866	0.042	0.668	15.831	0.103	1.635	15.928	0.102	1.623
BIKE2	IND-CPA	16.350	0.092	1.510	16.247	0.243	3.941	15.767	0.401	6.330
BIKE3	IND-CPA	16.116	0.192	3.101	16.127	0.507	8.173	16.064	1.161	18.644
Classic McEliece AVX	IND-CCA2	16.983	0.022	0.380	16.713	0.044	0.743	16.749	0.074	1.247
Classic McEliece SSE	IND-CCA2	16.335	0.027	0.437	16.307	0.057	0.924	16.151	0.103	1.658
FRODO AES	IND-CCA	16.435	0.562	9.232	16.760	1.107	18.553	16.919	1.829	30.949
FRODO SHAKE	IND-CCA	17.955	1.287	23.105	17.903	2.744	49.121	17.749	4.803	85.254
hqc-1	IND-CCA2	16.091	0.166	2.664	16.104	0.300	4.838	16.294	0.457	7.441
hqc-2	IND-CCA2	-	-	-	16.241	0.313	5.075	16.352	0.486	7.940
hqc-3	IND-CCA2	-	-	-	-	-	-	16.272	0.498	8.097
Kyber	IND-CCA2	16.762	0.014	0.236	16.761	0.023	0.384	16.908	0.032	0.544
Kyber-90s	IND-CCA2	16.086	0.009	0.148	15.925	0.013	0.206	15.761	0.018	0.291
LAC	IND-CCA	16.580	0.030	0.490	16.663	0.060	1.004	16.852	0.084	1.410
LEDAcrypt DFR64	IND-CCA2	15.950	0.137	2.178	16.219	0.262	4.253	16.274	0.591	9.613
LEDAcrypt DFRSL	IND-CCA2	16.268	0.163	2.648	16.169	0.579	9.360	16.407	0.864	14.184
LEDAcrypt N02	IND-CPA	16.091	0.043	0.687	16.009	0.087	1.393	16.251	0.170	2.757
LEDAcrypt N03	IND-CPA	15.929	0.033	0.518	16.278	0.077	1.259	16.136	0.174	2.815
LEDAcrypt N04	IND-CPA	16.058	0.040	0.644	16.173	0.103	1.660	16.195	0.205	3.320
NewHope	IND-CCA	16.713	0.032	0.528	-	-	-	16.697	0.059	0.986
NewHope	IND-CPA	16.870	0.025	0.420	-	-	-	16.829	0.046	0.777
NTRU-HRSS	IND-CCA2	-	-	-	16.571	0.038	0.634	-	-	-
NTS-KEM AVX	IND-CCA2	16.757	0.027	0.456	15.544	0.104	1.619	15.180	0.170	2.586
NTS-KEM SSE	IND-CCA2	16.643	0.027	0.456	15.222	0.112	1.710	14.920	0.186	2.769
Round5 N1 0d AES	IND-CPA	16.952	0.169	2.867	17.452	0.307	5.361	16.970	0.745	12.634
Round5 N1 0d SHAKE	IND-CPA	16.917	0.182	3.080	17.497	0.326	5.704	17.123	0.784	13.421
Round5 ND 0d AES	IND-CPA	16.450	0.023	0.372	16.599	0.080	1.329	16.827	0.096	1.619
Round5 ND 0d SHAKE	IND-CPA	16.449	0.024	0.395	16.628	0.083	1.386	16.896	0.100	1.695
Round5 ND 5d AES	IND-CPA	16.223	0.031	0.501	16.493	0.048	0.792	16.511	0.085	1.400
Round5 ND 5d SHAKE	IND-CPA	16.224	0.032	0.517	16.591	0.051	0.851	16.601	0.089	1.477
SABER	IND-CCA	17.008	0.020	0.338	17.041	0.033	0.565	16.983	0.050	0.845
SIKE	IND-CCA	16.942	3.135	53.112	17.000	8.387	142.576	16.920	12.415	210.070
SIKE Compressed	IND-CCA	16.794	6.048	101.565	16.930	14.090	238.542	17.025	23.486	399.847
Three Bears	IND-CCA	16.459	0.029*	0.475*	16.616	0.049*	0.806*	16.603	0.075	1.246
Three Bears	IND-CPA	16.594	0.029*	0.488*	16.560	0.050*	0.826*	16.819	0.077	1.298

TABLE 19. Energy consumed by key encapsulation mechanisms for decapsulation operations in additional optimized implementations.

Scheme	Security Notion	Level 1			Level 3			Level 5		
		Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
BIKE1	IND-CPA	15.493	0.173	2.684	15.559	0.452	7.037	15.655	0.878	13.740
BIKE2	IND-CPA	15.514	0.167	2.587	15.453	0.402	6.216	15.370	0.858	13.192
BIKE3	IND-CPA	15.636	0.256	4.000	15.607	0.697	10.884	15.723	1.587	24.951
Classic McEliece AVX	IND-CCA2	16.812	0.039	0.655	16.988	0.077	1.312	16.962	0.092	1.563
Classic McEliece SSE	IND-CCA2	16.377	0.051	0.834	16.326	0.115	1.875	16.210	0.133	2.148
FRODO AES	IND-CCA	16.467	0.538	8.853	16.789	1.060	17.802	16.977	1.767	29.998
FRODO SHAKE	IND-CCA	17.999	1.264	22.743	17.917	2.699	48.360	17.781	4.737	84.226
hqc-1	IND-CCA2	15.886	0.317	5.042	15.897	0.509	8.088	16.183	0.707	11.445
hqc-2	IND-CCA2	-	-	-	16.094	0.501	8.057	16.248	0.754	12.259
hqc-3	IND-CCA2	-	-	-	-	-	-	16.173	0.770	12.446
Kyber	IND-CCA2	16.863	0.011	0.182	16.899	0.019	0.317	16.977	0.027	0.466
Kyber-90s	IND-CCA2	16.148	0.006	0.105	15.894	0.010	0.155	15.923	0.015	0.234
LAC	IND-CCA	16.726	0.036	0.599	16.801	0.091	1.536	16.953	0.113	1.923
LEDAcrypt DFR64	IND-CCA2	15.645	0.333	5.210	15.754	0.681	10.721	15.947	1.315	20.966
LEDAcrypt DFRSL	IND-CCA2	15.597	0.433	6.750	15.939	0.998	15.913	16.394	1.867	30.605
LEDAcrypt N02	IND-CPA	15.409	0.259	3.997	15.376	0.631	9.704	15.664	1.019	15.963
LEDAcrypt N03	IND-CPA	15.664	0.343	5.369	15.581	0.771	12.016	15.716	1.483	23.300
LEDAcrypt N04	IND-CPA	15.669	0.759	11.886	15.823	1.993	31.533	16.067	3.101	49.820
NewHope	IND-CCA	16.738	0.032	0.541	-	-	-	16.702	0.062	1.042
NewHope	IND-CPA	16.592	0.005	0.089	-	-	-	16.549	0.010	0.163
NTRU-HRSS	IND-CCA2	-	-	-	16.725	0.022	0.375	-	-	-
NTS-KEM AVX	IND-CCA2	16.333	0.118	1.924	16.377	0.215	3.526	16.627	0.423	7.027
NTS-KEM SSE	IND-CCA2	16.030	0.202	3.245	16.082	0.400	6.440	16.212	0.871	14.126
Round5 N1 0d AES	IND-CPA	15.883	0.072	1.142	15.963	0.101	1.612	15.575	0.400	6.232
Round5 N1 0d SHAKE	IND-CPA	15.722	0.077	1.205	15.993	0.108	1.719	15.836	0.427	6.763
Round5 ND 0d AES	IND-CPA	16.242	0.011	0.174	16.497	0.043	0.717	16.634	0.053	0.888
Round5 ND 0d SHAKE	IND-CPA	16.408	0.011	0.173	16.440	0.043	0.715	16.668	0.054	0.907
Round5 ND 5d AES	IND-CPA	16.043	0.015	0.236	16.332	0.024	0.400	16.169	0.047	0.752
Round5 ND 5d SHAKE	IND-CPA	16.096	0.015	0.238	16.272	0.025	0.406	16.169	0.048	0.780
SABER	IND-CCA	16.977	0.019	0.325	17.057	0.033	0.566	17.046	0.050	0.860
SIKE	IND-CCA	16.935	3.357	56.845	17.008	8.459	143.868	16.923	13.408	226.900
SIKE Compressed	IND-CCA	16.820	5.576	93.789	16.933	13.526	229.040	17.036	21.773	370.918
Three Bears	IND-CCA	16.376	0.045*	0.744*	16.551	0.071*	1.182*	16.538	0.105	1.732
Three Bears	IND-CPA	16.191	0.011*	0.180*	16.375	0.015*	0.238*	16.643	0.018	0.304

TABLE 20. Energy consumed by digital signature for key generation operations in additional optimized implementations.

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.799	0.030	0.477	15.561	0.053	0.819	-	-	-
Dilithium SHAKE	16.712	0.042	0.695	16.634	0.080	1.338	-	-	-
BlueGeMSS	16.473	16.318	268.803	16.829	81.654	1374.129	16.770	250.917	4207.971
GeMSS	16.080	16.192	260.368	16.732	80.976	1354.920	16.707	249.262	4164.353
RedGeMSS	15.947	16.168	257.821	16.529	81.000	1338.864	16.568	251.026	4159.108
Luov Small Chacha	15.697	0.706*	11.077*	17.880	2.153*	38.502*	18.177	4.274	77.691
Luov Small Keccak	15.767	1.183*	18.648*	18.003	3.004*	54.089*	18.236	5.539	101.013
MQDSS	16.223	0.235*	3.817*	16.121	0.548*	8.830*	-	-	-
Picnic FS AVX	16.010	0.011	0.179	15.976	0.013	0.213	15.871	0.017	0.273
Picnic FS SSE	16.013	0.011	0.175	15.697	0.018	0.289	15.646	0.021	0.333
Picnic UR AVX	15.814	0.011	0.176	15.793	0.013	0.203	16.007	0.016	0.255
Picnic UR SSE	15.797	0.012	0.182	15.479	0.016	0.243	15.413	0.024	0.373
Picnic2 FS AVX	16.453	0.011	0.186	16.369	0.013	0.214	15.954	0.016	0.261
Picnic2 FS SSE	16.410	0.011	0.181	15.755	0.016	0.254	15.671	0.021	0.324
qTESLA	16.141	0.330	5.320	15.985	0.932	14.897	16.041	4.840	77.641
qTESLA-s	16.133	0.329	5.301	16.006	0.922	14.753	16.032	4.866	78.020
qTESLA-size	-	-	-	-	-	-	16.001	7.872	125.961
qTESLA-size-s	-	-	-	-	-	-	16.099	9.362	150.720
Rainbow Classic AVX	17.021	3.033	51.619	16.106	29.843	480.635	17.203	45.751	787.062
Rainbow Classic SSE	16.567	3.146	52.126	15.501	30.797	477.392	16.563	49.380	817.902
Rainbow Compressed Cyclic AVX	17.091	3.256	55.654	15.990	33.680	538.548	17.266	48.352	834.856
Rainbow Compressed Cyclic SSE	16.567	3.416	56.591	15.448	35.179	543.443	16.588	53.846	893.214
Rainbow Cyclic AVX	17.046	3.255	55.482	15.917	33.639	535.419	17.245	48.503	836.436
Rainbow Cyclic SSE	16.591	3.363	55.793	15.412	35.208	542.632	16.458	53.828	885.897
SPHINCS+ Haraka f robust	17.217	0.116	1.994	-	-	-	-	-	-
SPHINCS+ Haraka f simple	17.160	0.097	1.661	-	-	-	-	-	-
SPHINCS+ Haraka s robust	16.642	3.594	59.806	-	-	-	-	-	-
SPHINCS+ Haraka s simple	16.383	2.972	48.686	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	17.613	0.666	11.727	17.243	1.006	17.35	17.321	5.607	97.119
SPHINCS+ SHA-2 f simple	17.286	0.325	5.617	17.237	0.483	8.318	17.197	1.252	21.527
SPHINCS+ SHA-2 s robust	17.291	21.219	366.907	17.232	32.169	554.354	17.248	89.682	1546.814
SPHINCS+ SHA-2 s simple	17.310	10.350	179.155	17.287	15.318	264.801	17.267	20.180	348.451
SPHINCS+ SHAKE f robust	17.557	2.169	38.075	17.432	3.144	54.805	17.509	8.266	144.733
SPHINCS+ SHAKE f simple	17.511	1.138	19.932	17.486	1.664	29.089	17.554	4.377	76.829
SPHINCS+ SHAKE s robust	17.537	69.344	1216.122	17.516	100.503	1760.398	17.435	132.203	2304.999
SPHINCS+ SHAKE s simple	17.525	36.385	637.642	17.503	53.131	929.978	17.467	70.018	1223.005

TABLE 21. Energy consumed by digital signature for signing operations in additional optimized implementations.

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	16.077	0.103	1.653	15.892	0.133	2.116	-	-	-
Dilithium SHAKE	16.867	0.126	2.129	16.680	0.182	3.032	-	-	-
BlueGeMSS	15.394	52.282	804.838	16.052	116.799	1874.894	16.074	181.760	2921.569
GeMSS	15.098	319.657	4826.297	15.788	771.641	12182.907	15.820	1320.126	20884.447
RedGeMSS	15.174	1.444	21.908	15.531	3.149	48.909	15.572	5.088	79.235
Luov Small Chacha	17.677	0.295*	5.214*	17.509	0.806*	14.116*	17.729	1.290	22.870
Luov Small Keccak	16.484	0.777*	12.810*	17.802	1.659*	29.537*	18.120	2.544	46.094
MQDSS	17.147	0.984*	16.869*	17.497	2.632*	46.059*	-	-	-
Picnic FS AVX	17.070	1.783	30.431	16.919	4.233	71.623	17.082	7.452	127.294
Picnic FS SSE	15.948	3.422	54.568	16.175	9.454	152.917	15.976	15.262	243.819
Picnic UR AVX	16.993	2.200	37.391	17.019	5.189	88.304	17.048	8.832	150.578
Picnic UR SSE	16.033	4.388	70.346	15.886	10.745	170.696	15.793	20.896	330.023
Picnic2 FS AVX	17.389	71.053	1235.536	17.210	198.987	3424.477	17.252	407.244	7025.980
Picnic2 FS SSE	17.052	104.405	1780.298	16.854	291.056	4905.421	16.798	606.173	10182.704
qTESLA	16.825	0.101	1.696	16.771	0.127	2.137	16.951	0.346	5.860
qTESLA-s	17.002	0.104	1.766	16.865	0.137	2.306	17.093	0.359	6.129
qTESLA-size	-	-	-	-	-	-	15.492	1.142	17.685
qTESLA-size-s	-	-	-	-	-	-	15.651	1.407	22.015
Rainbow Classic AVX	16.861	0.028	0.475	15.948	0.184	2.932	16.711	0.227	3.799
Rainbow Classic SSE	16.403	0.050	0.820	15.874	0.200	3.172	16.687	0.273	4.553
Rainbow Compressed Cyclic AVX	16.901	2.286	38.635	16.047	21.636	347.185	16.996	34.561	587.420
Rainbow Compressed Cyclic SSE	16.553	2.326	38.501	15.621	22.245	347.499	16.583	36.903	611.978
Rainbow Cyclic AVX	16.884	0.028	0.478	15.828	0.184	2.913	16.600	0.232	3.849
Rainbow Cyclic SSE	16.348	0.050	0.823	15.885	0.200	3.178	16.578	0.273	4.529
SPHINCS+ Haraka f robust	16.470	4.759	78.388	-	-	-	-	-	-
SPHINCS+ Haraka f simple	16.282	3.587	58.410	-	-	-	-	-	-
SPHINCS+ Haraka s robust	16.529	81.522	1347.437	-	-	-	-	-	-
SPHINCS+ Haraka s simple	16.045	60.205	965.996	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	17.489	22.098	386.464	17.072	31.050	530.083	17.305	127.625	2208.538
SPHINCS+ SHA-2 f simple	17.204	10.966	188.664	17.097	15.365	262.701	17.212	31.491	542.003
SPHINCS+ SHA-2 s robust	17.397	362.855	6312.659	17.432	710.459	12384.616	17.326	1031.741	17876.214
SPHINCS+ SHA-2 s simple	17.393	179.498	3121.931	17.394	372.608	6481.016	17.377	256.334	4454.214
SPHINCS+ SHAKE f robust	17.422	67.830	1181.753	17.298	88.052	1523.103	17.458	187.933	3280.994
SPHINCS+ SHAKE f simple	17.423	37.148	647.238	17.366	47.408	823.288	17.499	102.449	1792.770
SPHINCS+ SHAKE s robust	17.552	1029.428	18068.813	17.520	2084.009	36511.746	17.452	1535.987	26806.802
SPHINCS+ SHAKE s simple	17.539	577.403	10127.173	17.514	1195.416	20936.474	17.493	848.509	14843.168

TABLE 22. Energy consumed by digital signature for verification operations in additional optimized implementations.

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
Dilithium AES	15.858	0.039	0.619	15.549	0.064	0.988	-	-	-
Dilithium SHAKE	16.583	0.048	0.788	16.556	0.086	1.423	-	-	-
BlueGeMSS	16.735	0.053	0.889	16.999	0.143	2.434	16.859	0.316	5.331
GeMSS	16.361	0.052	0.855	16.903	0.140	2.372	16.844	0.312	5.249
RedGeMSS	16.436	0.055	0.908	16.708	0.143	2.385	16.751	0.319	5.345
Luov Small Chacha	17.427	0.125*	2.181*	16.974	0.397*	6.744*	17.242	0.576	9.927
Luov Small Keccak	16.087	0.607*	9.761*	17.726	1.245*	22.072*	18.084	1.836	33.209
MQDSS	17.579	0.658*	11.570*	17.893	1.842*	32.951*	-	-	-
Picnic FS AVX	17.166	1.433	24.602	16.905	3.545	59.921	17.051	6.333	107.979
Picnic FS SSE	15.958	2.770	44.197	16.101	7.909	127.337	16.037	12.874	206.455
Picnic UR AVX	17.019	1.780	30.286	16.978	4.344	73.750	16.999	7.445	126.552
Picnic UR SSE	16.195	3.524	57.068	15.792	8.789	138.787	15.732	17.272	271.727
Picnic2 FS AVX	17.202	33.453	575.467	16.968	74.751	1268.382	16.982	131.665	2235.965
Picnic2 FS SSE	16.894	56.541	955.210	16.653	129.100	2149.849	16.610	236.941	3935.563
qTESLA	16.595	0.026	0.426	16.744	0.045	0.748	16.659	0.091	1.524
qTESLA-s	16.604	0.026	0.427	16.705	0.046	0.767	16.551	0.090	1.496
qTESLA-size	-	-	-	-	-	-	15.545	0.220	3.423
qTESLA-size-s	-	-	-	-	-	-	15.730	0.252	3.962
Rainbow Classic AVX	17.051	0.014	0.235	17.700	0.047	0.829	18.164	0.079	1.429
Rainbow Classic SSE	16.832	0.028	0.475	16.185	0.201	3.245	17.712	0.178	3.158
Rainbow Compressed Cyclic AVX	16.548	1.414	23.402	16.587	8.332	138.199	16.407	20.796	341.196
Rainbow Compressed Cyclic SSE	16.418	1.439	23.622	16.565	8.484	140.534	16.531	21.072	348.354
Rainbow Cyclic AVX	16.469	1.417	23.336	16.501	8.340	137.624	16.449	20.886	343.562
Rainbow Cyclic SSE	16.433	1.432	23.532	16.531	8.499	140.492	16.409	20.890	342.792
SPHINCS+ Haraka f robust	14.737	0.343	5.047	-	-	-	-	-	-
SPHINCS+ Haraka f simple	15.049	0.221	3.331	-	-	-	-	-	-

TABLE 22. *(Continued.)* **Energy consumed by digital signature for verification operations in additional optimized implementations.**

Scheme	Level 1			Level 3			Level 5		
	Power	Time	Energy	Power	Time	Energy	Power	Time	Energy
SPHINCS+ Haraka s robust	15.386	0.153	2.353	-	-	-	-	-	-
SPHINCS+ Haraka s simple	15.507	0.100	1.547	-	-	-	-	-	-
SPHINCS+ SHA-2 f robust	16.195	3.078	49.853	15.883	5.137	81.588	15.863	7.606	120.648
SPHINCS+ SHA-2 f simple	15.853	1.510	23.934	15.810	2.469	39.040	15.841	2.513	39.801
SPHINCS+ SHA-2 s robust	15.936	1.283	20.445	15.923	2.047	32.595	15.808	3.885	61.420
SPHINCS+ SHA-2 s simple	15.964	0.634	10.122	15.832	0.989	15.665	15.952	1.287	20.537
SPHINCS+ SHAKE f robust	15.834	5.051	79.976	15.740	8.050	126.699	15.853	8.063	127.831
SPHINCS+ SHAKE f simple	15.998	2.538	40.597	15.842	4.046	64.105	15.796	4.091	64.628
SPHINCS+ SHAKE s robust	15.893	2.093	33.266	15.867	3.094	49.088	15.886	3.986	63.319
SPHINCS+ SHAKE s simple	15.981	1.060	16.947	15.926	1.546	24.627	15.863	2.015	31.966

ACKNOWLEDGMENT

An early version of this work is available as a technical report on the University of Waterloo's Centre for Applied Cryptographic Research website and was presented at NIST's Second PQC Standardization Conference in 2019.

REFERENCES

[1] *Post-Quantum Cryptography Standardization Call for Proposals Announcement*. (2017). [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

[2] National Institute of Standards and Technology. (2020). *PQC Standardization Process: Third Round Candidate Announcement*. [Online]. Available: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

[3] 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," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR-8240, Jan. 2019, doi: 10.6028/nist.ir.8240.

[4] National Institute of Standards and Technology. (2016). *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[5] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Trans. Mobile Comput.*, vol. 5, no. 2, pp. 128–143, Feb. 2006.

[6] C. G. Thorat and V. S. Inamdar, "Energy measurement of encryption techniques using RAPL," in *Proc. Int. Conf. Comput., Commun., Control Autom. (ICCUBEA)*, Aug. 2017, pp. 1–4.

[7] B. Atawneh, L. AL-Hammoury, and M. Abutaha, "Power consumption of a chaos-based stream cipher algorithm," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Mar. 2020, pp. 1–4.

[8] T. Banerjee and M. A. Hasan, "Energy efficiency analysis of elliptic curve based cryptosystems," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1579–1583.

[9] E. G. Abdallah, Y. R. Kuang, and C. Huang, "Advanced encryption standard new instructions (AES-NI) analysis: Security, performance, and power consumption," in *Proc. 12th Int. Conf. Comput. Autom. Eng. (ICCAE)*. New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 167–172. [Online]. Available: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3384613.3384648

[10] National Institute of Standards and Technology. (2018). *Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process*. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf

[11] R. AlTawy, G. Gong, M. He, K. Mandal, and R. Rohit. (2019). *SPIX: An Authenticated Cipher Submission to the NIST LWC Competition*. [Online]. Available: https://uwaterloo.ca/communications-security-lab/lwc/spix

[12] T. Iwata, M. Khairallah, K. Minematsu, and T. Peyrin. (2019). *Romulus V1.2*. [Online]. Available: https://romulusae.github.io/romulus/

[13] M. Hell, T. Johansson, W. Meier, J. Sönnerup, and H. Yoshida. (2019). *Grain-128AEAD—A Lightweight AEAD Stream Cipher*. [Online]. Available: https://grain-128aead.github.io/

[14] S. Banik, A. Chakraborti, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo. (2019). *GIFT-COFB*. [Online]. Available: https://www.isical.ac.in/~lightweight/COFB/

[15] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. (2019). *Ascon V1.2*. [Online]. Available: https://ascon.iaik.tugraz.at/

[16] M. Aagaard, R. AlTawy, G. Gong, K. Mandal, and R. Rohit. (2019). *ACE: An Authenticated Encryption and Hash Algorithm*. [Online]. Available: https://uwaterloo.ca/communications-security-lab/lwc/ace

[17] M. Aagaard, R. AlTawy, G. Gong, K. Mandal, R. Rohit, and N. Zidaric. (2019). *WAGE: An Authenticated Cipher*. [Online]. Available: https://uwaterloo.ca/communications-security-lab/lwc/wage

[18] M.-J. O. Saarinen, "Mobile energy requirements of the upcoming NIST post-quantum cryptography standards," in *Proc. 8th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Aug. 2020, pp. 23–30.

[19] D. Moody. (Jan. 2021). *Diversity of Signature Schemes*. [Online]. Available: https://groups.google.com/a/list.nistgov/g/pqc-forum/c/2LEoSpskELs/m/VB1jng0aCAAJ

[20] T. Banerjee and M. A. Hasan, "Energy consumption of candidate algorithms for NIST PQC standards," Univ. Waterloo, Centre Appl. Cryptograph. Res., Waterloo, ON, Canada, Tech. Rep. CACR 2018-06, Jun. 2018. [Online]. Available: http://cacr.uwaterloo.ca/techreports/2018/cacr2018-06.pdf

[21] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini. (2020). *LEDAcrypt: Low-Density Parity-Check Code-Based Cryptographic Systems (Second Round)*. [Online]. Available: https://www.ledacrypt.org/

[22] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. (2020). *GeMSS: A Great Multivariate Short Signature*. [Online]. Available: https://www-polsys.lip6.fr/Links/NIST/GeMSS.html

[23] National Institute of Standards and Technology. (2019). *Post-Quantum Cryptography Round 2 Submissions*. [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions

[24] 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. (2020). *BIKE: Bit Flipping Key Encapsulation (Spec V3)*. [Online]. Available: https://bikesuite.org/spec.html

[25] 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*. (2020). [Online]. Available: https://classic.mceliece.org/nist.html

[26] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila. (2020). *FrodoKEM: Learning With Errors Key Encapsulation (Round 2 Specification)*. [Online]. Available: https://frodokem.org/#spec

[27] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor. (2020). *Hamming Quasi-Cyclic (HQC) (Round 2)*. [Online]. Available: http://pqc-hqc.org/documentation.html

[28] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. (2021). *CRYSTALS-Kyber (Version 2)*. [Online]. Available: https://pq-crystals.org/kyber/resources.shtml
[29] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, Z. Zhang, Z. Liu, H. Yang, B. Li, and K. Wang. (2021). *LAC: Lattice-Based Cryptosystems*. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
[30] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, D. Stebila, M. R. Albrecht, E. Orsini, V. Osheter, K. G. Paterson, G. Peer, and N. P. Smart. (2020). *NewHope (Version 1.0.2)*. [Online]. Available: https://newhopecrypto.org/
[31] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang. (2020). *NTRU (Second Round)*. [Online]. Available: https://ntru.org/resources.shtml
[32] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal. (2020). *NTRU Prime: Round 2*. [Online]. Available: https://ntruprime.cr.yp.to/nist.html
[33] M. Albrecht, C. Cid, K. G. Paterson, C. J. Tjhai, and M. Tomlinson. (2019). *NTS-KEM (Second Round Submission)*. [Online]. Available: https://nts-kem.io/
[34] 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. (2020). *ROLLO-Rank-Ouroboros, LAKE & LOCKER Full Submission Package (Round 2)*. [Online]. Available: https://pqc-rollo.org/implementation.html
[35] H. Baan, S. Bhattacharya, S. Fluhrer, O. Garcia-Morchon, T. Laarhoven, R. Player, R. Rietman, M.-J. O. Saarinen, Y. Son, L. Tolhuizen, J. L. Torre-Arce, and Z. Zhang. (2020). *Round5: KEM and PKE Based on (Ring) Learning With Rounding*. [Online]. Available: https://round5.org/#spec
[36] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, A. Couvreur, J.-C. Deneuville, P. Gaborit, A. Hauteville, and G. Zémor. (2020). *Rank Quasi-Cyclic (RQC) Full Submission Package (Round 2)*. [Online]. Available: http://pqc-rqc.org/implementation.html
[37] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren. (2020). *SABER: Mod-LWR Based KEM (Round 2 Submission)*. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/resources.html
[38] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. D. Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, D. Urbanik, and G. Pereira. (2021). *Supersingular Isogeny Key Encapsulation*. [Online]. Available: https://sike.org/#nist-submission
[39] M. Hamburg. (2021). *Post-Quantum Cryptography Proposal: THREE-BEARS*. [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions
[40] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. (2021). *CRYSTALS-Dilithium*. [Online]. Available: https://pq-crystals.org/dilithium/resources.shtml
[41] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. (2019). *FALCON: Fast-Fourier Lattice-Based Compact Signatures Over NTRU*. [Online]. Available: https://falcon-sign.info/
[42] W. Beullens, B. Preneel, A. Szepieniec, and F. Vercauteren. (2019). *LUOV*. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/luov/
[43] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe. (2020). *MQDSS Specifications (Version 2.0)*. [Online]. Available: http://mqdss.org/specification.html
[44] G. Zaverucha, M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, J. Katz, X. Wang, V. Kolesnikov, and D. Kales. (2020). *The Picnic Signature Algorithm*. [Online]. Available: https://microsoft.github.io/Picnic/
[45] N. Bindel, S. Akleylek, E. Alkim, P. S. L. M. Barreto, J. Buchmann, E. Eaton, G. G. J. Krämer, P. Longa, H. Polat, J. E. Ricardini, and G. Zanon. (2020). *Submission to NIST's Post-Quantum Project (2nd Round): Lattice-Based Digital Signature Scheme qTESLA*. [Online]. Available: https://qtesla.org/#specs
[46] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, and B.-Y. Yang. (2021). *Rainbow—Algorithm Specification and Documentation*. [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions
[47] J.-P. Aumasson, D. J. Bernstein, C. Dobraunig, M. Eichlseder, 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. (2020). *SPHINCS+* [Online]. Available: https://sphincs.org/resources.html
[48] Intel. (Nov. 2020). *Intel 64 and IA-32 Architectures Software Developer's Manual—Volume 3B System Programming Guide Part 2*. [Online]. Available: https://software.intel.com/en-us/download/intel-64-and-ia-32-architectures-sdm-volume-3b-system-programming-guide-part-2
[49] V. M. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore, "Measuring energy and power with PAPI," in *Proc. 41st Int. Conf. Parallel Process. Workshops*, 2012, pp. 262–268.
[50] GNU Compiler Collection. *×86 Options (Using the GNU Compiler Collection (GCC))*. Accessed: Apr. 4, 2021. [Online]. Available: https://gcc.gnu.org/onlinedocs/gcc/x86-Options.html
[51] D. Branco and P. R. Henriques, "Impact of GCC optimization levels in energy consumption during C/C++ program execution," in *Proc. IEEE 13th Int. Sci. Conf. Informat.*, Nov. 2015, pp. 52–56.
[52] L. Tuura, G. Eulisse, M. Kurkela, and F. Nybäck. (2019). *IgProf, the Ignominous Profiler*. [Online]. Available: https://igprof.org/
[53] K. N. Khan, F. Nyback, Z. Ou, J. K. Nurminen, T. Niemi, G. Eulisse, P. Elmer, and D. Abdurachmanov, "Energy profiling using IgProf," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 1115–1118.
[54] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann, 2012.
[55] J. M. Cardoso, J. G. F. Coutinho, and P. C. Diniz, "High-performance embedded computing," in *Embedded Computing for High Performance*, J. M. Cardoso, J. G. F. Coutinho, and P. C. Diniz, Eds. Boston, MA, USA: Morgan Kaufmann, 2017, ch. 2, pp. 17–56. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128041895000028
[56] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, "RAPL in action: Experiences in using RAPL for power measurements," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 3, no. 2, pp. 1–26, Apr. 2018.
[57] E. Blem, J. Menon, and K. Sankaralingam. (Jan. 2013). *A Detailed Analysis of Contemporary ARM and ×86 Architectures*. [Online]. Available: https://www.researchgate.net/publication/266457125_A_Detailed_Analysis_of_Contemporary_ARM_and_x86_Architectures
[58] A. Akram and L. Sawalha, "A study of performance and power consumption differences among different ISAs," in *Proc. 22nd Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2019, pp. 628–632.
[59] E. Vasilakis, "An instruction level energy characterization of ARM processors," Univ. Crete, Comput. Archit. VLSI Syst. (CARV) Lab., Inst. Comput. Sci. (ICS), Found. Res. Technol. Hellas (FORTH), Heraklion, Greece, Tech. Rep. FORTH-ICS/TR-450, Mar. 2015.
[60] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations: A survey," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–41, Feb. 2019, doi: 10.1145/3292548.
[61] M. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. FIPS PUB 202, Aug. 2015.
[62] J. Kelsey, S. Chang, and R. A. Perlner, "SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST 800-185, Dec. 2016.
[63] Intel Corporation. *Intel 64 and IA-32 Architectures Software Developer's Manual—Volume 2 (2A, 2B, 2C & 2D): Instruction Set Reference, A-Z*. Nov. 2020. [Online]. Available: https://software.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-2a-2b-2c-and-2d-instruction-set-reference-a-z.html
[64] N. Firasta, M. Buxton, P. Jinbo, K. Nasri, and S. Kuo, "Intel AVX: New frontiers in performance improvements and energy efficiency," Intel Corp., Santa Clara, CA, USA, White Paper, Mar. 2008.
[65] K. Czechowski, V. W. Lee, E. Grochowski, R. Ronen, R. Singhal, R. Vuduc, and P. Dubey, "Improving the energy efficiency of big cores," in *Proc. ACM/IEEE 41st Int. Symp. Comput. Archit. (ISCA)*, Jun. 2014, pp. 493–504.
[66] J. M. Cebrián, L. Natvig, and J. C. Meyer, "Performance and energy impact of parallelization and vectorization techniques in modern microprocessors," *Computing*, vol. 96, no. 12, pp. 1179–1193, Dec. 2014, doi: 10.1007/s00607-013-0366-5.
[67] S. Gueron, "Intel's new AES instructions for enhanced performance and security," in *Fast Software Encryption*, O. Dunkelman, Ed. Berlin, Germany: Springer, 2009, pp. 51–66.
[68] K. Akdemir, M. Dixon, W. Feghali, P. Fay, V. Gopal, J. Guiford, E. Ozturk, G. Wolrish, and R. Zohar, "Breakthrough AES performance with Intel AES new instructions," Intel Corp., Santa Clara, CA, USA, White Paper, 2010.

[69] A. Hoban, "Using Intel AES new instructions and PCLMULQDQ to significantly improve IPSec performance on Linux*," Intel Corp., Santa Clara, CA, USA, Aug. 2010.
[70] S. Gulley, V. Gopal, K. Yap, W. Feghali, J. Guilford, and G. Wolrich, "Intel SHA extensions: New instructions supporting the secure hash algorithm on Intel architecture processors," Intel Corp., Santa Clara, CA, USA, Jul. 2013.
[71] S. Kölbl, M. M. Lauridsen, F. Mendel, and C. Rechberger, "Haraka V2—Efficient short-input hashing for post-quantum applications," *IACR Trans. Symmetric Cryptol.*, vol. 2016, no. 2, pp. 1–29, Feb. 2017. [Online]. Available: https://tosc.iacr.org/index.php/ToSC/article/view/563
[72] G. Pinto and F. Castor, "Energy efficiency," *Commun. ACM*, vol. 60, no. 12, pp. 68–75, Nov. 2017.
[73] D. Molka, D. Hackenberg, R. Schöne, and M. S. Müller, "Characterizing the energy consumption of data transfers and arithmetic operations on ×86−64 processors," in *Proc. Int. Conf. Green Comput.*, Aug. 2010, pp. 123–133.
[74] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong, "Efficient online/offline identity-based signature for wireless sensor network," *Int. J. Inf. Secur.*, vol. 9, no. 4, pp. 287–296, Jun. 2010.
[75] A. Menezes, S. Vanstone, and P. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996, ch. 11.
[76] A. Adeel, M. Ali, A. N. Khan, T. Khalid, F. Rehman, Y. Jararweh, and J. Shuja, "A multi-attack resilient lightweight IoT authentication scheme," *Trans. Emerg. Telecommun. Technol.*, Jul. 2019, Art. no. e3676.

CRYSTAL ANDREA ROMA received a bachelor's degree (Hons.) in electrical engineering from the University of Windsor, in 2017, and a master's degree in electrical and computer engineering from the University of Waterloo under the supervision of Dr. M. A. Hasan with a concentration in computer hardware, in 2019. This work was completed while she worked as a Research Associate with Dr. M. A. Hasan with the University of Waterloo from 2019 to 2020. She is currently working with Ford Motor Company in the area of in-vehicle cybersecurity for advanced driver assistance systems. Her research interests include cryptographic computations and embedded systems, RTL design, and applied cryptography.

CHI-EN AMY TAI is currently pursuing a bachelor's degree in management engineering with the University of Waterloo. She worked as an Undergraduate Research Assistant under the supervision of Dr. M. A. Hasan, in 2020, and has interned at several Fortune 100 companies in the past. Her research interests include applied cryptography, security in data systems, and data for good.

M. ANWAR HASAN (Senior Member, IEEE) was the Faculty of Engineering's Associate Dean of Research and External Partnerships, from January 2013 to April 2018. He is currently the Ripple Chair and an Electrical and Computer Engineering Professor with the University of Waterloo. He is also a Faculty Member with the Centre for Applied Cryptographic Research. His research interests include cryptographic computations and embedded systems, dependable and secure computing, and security for cloud and the Internet of Things.