NIST IR 8413-upd1

Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process

Gorjan Alagic
Daniel Apon
David Cooper
Quynh Dang
Thinh Dang
John Kelsey
Jacob Lichtinger
Yi-Kai Liu
Carl Miller
Dustin Moody
Rene Peralta
Ray Perlner
Angela Robinson
Daniel Smith-Tone

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8413-upd1

National Institute of Standards and Technology
U.S. Department of Commerce

NIST IR 8413-upd1

Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process

Gorjan Alagic
Daniel Apon*
David Cooper
Quynh Dang
Thinh Dang
John Kelsey
Jacob Lichtinger
Carl Miller
Dustin Moody
Rene Peralta
Ray Perlner
Angela Robinson
Daniel Smith-Tone

Computer Security Division
Information Technology Laboratory
** Former NIST employee; all work for this publication was done while at or under contract with NIST.*

Yi-Kai Liu
Applied and Computational Mathematics Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8413-upd1

July 2022
Includes updates as of 09-26-2022; see Appendix E

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report
NIST IR 8413-upd1
102 pages (July 2022)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8413-upd1

Submit comments on this publication to: pqc-comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

The National Institute of Standards and Technology is in the process of selecting public-key cryptographic algorithms through a public, competition-like process. The new public-key cryptography standards will specify additional digital signature, public-key encryption, and key-establishment algorithms to augment Federal Information Processing Standard (FIPS) 186-4, *Digital Signature Standard (DSS)*, as well as NIST Special Publication (SP) 800-56A Revision 3, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*, and SP 800-56B Revision 2, *Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography*. It is intended that these algorithms will be capable of protecting sensitive information well into the foreseeable future, including after the advent of quantum computers.

This report describes the evaluation and selection process of the NIST Post-Quantum Cryptography Standardization process third-round candidates based on public feedback and internal review. The report summarizes each of the 15 third-round candidate algorithms and identifies those selected for standardization, as well as those that will continue to be evaluated in a fourth round of analysis. The public-key encryption and key-establishment algorithm that will be standardized is CRYSTALS–KYBER. The digital signatures that will be standardized are CRYSTALS–Dilithium, FALCON, and SPHINCS$^+$. While there are multiple signature algorithms selected, NIST recommends CRYSTALS–Dilithium as the primary algorithm to be implemented. In addition, four of the alternate key-establishment candidate algorithms will advance to a fourth round of evaluation: BIKE, Classic McEliece, HQC, and SIKE. These candidates are still being considered for future standardization. NIST will also issue a new Call for Proposals for public-key digital signature algorithms to augment and diversify its signature portfolio.

Keywords

cryptography; digital signatures; key-encapsulation mechanism (KEM); key-establishment; post-quantum cryptography; public-key encryption; quantum resistant; quantum safe

Supplemental Content

The NIST Post-Quantum Cryptography Standardization Process webpage is available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization

Acknowledgments

NIST would like to thank all of the candidate submission teams who developed, designed, and analyzed post-quantum public-key algorithms and prepared detailed submission packages describing their algorithms.

NIST is also grateful for the efforts of those in the cryptographic community who provided security, implementation, and performance analyses of the candidate algorithms during the first, second, and third rounds. NIST would not be able to select new post-quantum public-key algorithms for standardization without the combined efforts of these individuals and the algorithm submitters.

The authors of this report are also appreciative of the efforts by other members of NIST's Post-Quantum Cryptography team who reviewed candidate algorithms, analyses, and public comments; performed testing; provided technical and administrative support; and participated in numerous meetings to discuss the selection of the second-round candidates. They are Zuzana Bajcsy, Larry Bassham, Lily Chen, Morris Dworkin, Sara Kerman, and Andrew Regenscheid. Finally, the authors also would like to thank NIST legal counsel for helpful discussions.

Contents

List of Tables

List of Figures

1. Introduction

Over the past several years, there has been steady progress toward building quantum computers. The security of many commonly used public-key cryptosystems would be at risk if large-scale quantum computers were ever realized. In particular, this would include key-establishment schemes and digital signatures that are based on factoring, discrete logarithms, and elliptic curve cryptography. In contrast, symmetric cryptographic primitives, such as block ciphers and hash functions, would not be as drastically impacted. As a result, there has been intensified research into finding public-key cryptosystems that would be secure against adversaries with both quantum and classical computers. This field is often referred to as *post-quantum cryptography* (PQC), or sometimes quantum-resistant cryptography. The goal is to develop schemes that can be deployed in existing communication networks and protocols without significant modifications.

In response, the National Institute of Standards and Technology (NIST) initiated a public, competition-like process to select quantum-resistant public-key cryptographic algorithms. The new public-key cryptography standards will specify algorithms for digital signatures, public-key encryption, and key establishment. The new standards will augment Federal Information Processing Standard (FIPS) 186-4, Digital Signature Standard (DSS) [1], Special Publication (SP) 800-56A Revision 3, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography* [2], and SP 800-56B Revision 2, *Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography* [3]. It is intended that these algorithms will be capable of protecting sensitive U.S. Government information well into the foreseeable future, including after the advent of quantum computers. The process will be referred to as the NIST Post-Quantum Cryptography Standardization Process hereafter in this document.

NIST issued a public call for submissions to the PQC Standardization Process in December 2016 [4]. Prior to the November 2017 deadline a total of 82 candidate algorithms were submitted. Shortly thereafter, the 69 candidates that met both the submission requirements and the minimum acceptability criteria were accepted into the first round of the standardization process. Submission packages for the first-round candidates were posted online for public review and comment [5].

After a year-long review of the candidates, NIST selected 26 algorithms to move on to the second round of evaluation in January 2019 [6]. These algorithms were viewed as the most promising candidates for eventual standardization, and were selected based on both internal analysis and public feedback. During the second round, there was continued evaluation by NIST and the broader cryptographic community. After careful deliberation, NIST selected seven finalists and eight alternates to move on to the third round in July 2020 [7]. NIST's intent was to standardize a small number of the finalists at the end of the third round, as well as a small number of the alternate candidates after a fourth round.

The third round began in July 2020 and continued for approximately 18 months. During the third round, there was a more thorough analysis of the theoretical and empirical evidence used to justify the security of the candidates. There was also careful benchmarking

of their performance using optimized implementations on a variety of software and hardware platforms. Similar to the first two rounds, NIST also held the (virtual) Third NIST PQC Standardization Conference in June 2021. Each of the finalists and alternates were invited to present an update on their candidate algorithm. In addition, several researchers presented work that was relevant to the PQC standardization process.

After three rounds of evaluation and analysis, NIST has selected the first algorithms it will standardize as a result of the PQC Standardization Process. The public-key encapsulation mechanism (KEM) that will be standardized is CRYSTALS–KYBER.[1] The digital signatures that will be standardized are CRYSTALS–Dilithium, FALCON, and SPHINCS$^+$. While there are multiple signature algorithms selected, NIST recommends CRYSTALS–Dilithium as the primary algorithm to be implemented. In addition, four of the alternate KEM candidate algorithms will advance to a fourth round of evaluation: BIKE, Classic McEliece, HQC, and SIKE. These candidates will be considered for future standardization at the conclusion of the fourth round.

Table 1 shows a timeline of major events with respect to the NIST PQC Standardization Process to date.

[1] NIST has attempted to format the candidate names as given in their submission documents. We apologize for any mistakes

Table 1. Timeline of the NIST Post-Quantum Cryptography Standardization Process

Date	*Event*
April 2015	Workshop on Cybersecurity in a Post-Quantum World, NIST, Gaithersburg, MD
February 2016	PQC Standardization: Announcement and outline of NIST's Call for Submissions presentation given at PQCrypto 2016
April 2016	NISTIR 8105, *Report on Post-Quantum Cryptography* [8], released
December 2016	Federal Register Notice – Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms [4]
November 30, 2017	Submission Deadline for NIST PQC Standardization Process
December 2017	First-round candidates announced. The public comment period on the first-round candidates began.
April 2018	First NIST PQC Standardization Conference, Ft. Lauderdale, FL
January 2019	Second-round candidates announced. NISTIR 8240, *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process* [6], released. The public comment period on the second-round candidates began.
August 2019	Second NIST PQC Standardization Conference, Santa Barbara, CA
April 2020	NIST invited comments from submitters and the community to inform its decision-making process for the selection of third-round candidates.
July 2020	Third-round finalists and alternate candidates announced. NISTIR 8309, *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process* [7], released. The public comment period on the third-round candidates began.
June 2021	Third NIST PQC Standardization Conference, held virtually
July 2022	Candidate algorithms to be standardized are announced, along with alternate candidates advancing to the fourth round. NIST-IR 8413-upd1, *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*, released.

1.1 Purpose and Organization of this Document

The purpose of this document is to report on the third round of the NIST PQC Standardization Process. The report is organized as follows.

Section 2 enumerates the candidates that were included in the third round. Descriptions of the evaluation criteria and selection process used to ultimately select from the third-round finalists and alternate candidates are then provided. The algorithms that will be standardized are then named, along with the candidates moving into a fourth round of evaluation and analysis.

Section 3 contains some technical material relevant to the candidate algorithms. This includes a brief explanation of the underlying security problems, as well as the definitions of various computational models that NIST used in its evaluation.

Section 4 summarizes each of the third-round candidates. For each candidate, there is a brief description of the algorithm and its characteristics. This report presents reasons why candidate algorithms were either selected for standardization (or the fourth round), as well as reasons why the other candidate algorithms were not selected.

Section 5 describes the next steps in the NIST PQC Standardization Process. More details are provided on standardizing the algorithms selected, and on the process for evaluating candidate algorithms selected for the fourth round. Section 5 also mentions a new Call for Proposals for public-key digital signature algorithms.

2. Evaluation Criteria and the Selection Process

2.1 Acceptance of the Third-Round Candidates

NIST selected 15 candidate algorithms for the third round. Seven of the 15 algorithms were chosen to be 'finalists,' while the other eight algorithms were labelled 'alternates' [7]. The set of finalists included the algorithms that NIST considered to be the most promising to fit the majority of use cases and the most likely to be ready for standardization soon after the end of the third round. The alternate candidates were regarded as potential candidates for future standardization, most likely after another round of evaluation. Some of the alternate candidates have worse performance characteristics than the finalists but might yet be selected for standardization based on NIST's high confidence in their security. Others have acceptable performance but require additional analysis or other work to inspire sufficient confidence in their security for NIST to standardize. In addition, some alternate candidates were selected based either on NIST's desire for diversity in future post-quantum security standards or on their potential for further improvement.

The seven finalists included four key encapsulation mechanisms (KEMs), and three digital signature mechanisms. Of the eight alternates, five were KEMs and three were digital signatures. Submission teams were allowed to make minor modifications and resubmit their packages, which had to meet the same requirements as the original submissions. The complete updated specifications were posted on NIST's PQC website [5] on October 23, 2020, for public review.

Table 2. Third-Round Finalists

Public-Key Encryption/KEMs	Digital Signatures
Classic McEliece	CRYSTALS–Dilithium
CRYSTALS–KYBER	FALCON
NTRU	Rainbow
Saber	

Table 3. Third-Round Alternate Candidates

Public-Key Encryption/KEMs	Digital Signatures
BIKE	G*e*MSS
FrodoKEM	Picnic
HQC	SPHINCS$^+$
NTRU Prime	
SIKE	

2.2 Evaluation Criteria

NIST's Call for Proposals identified three broad aspects of the evaluation criteria that would be used to compare candidate algorithms throughout the NIST PQC Standardization Process: 1) security, 2) cost and performance, and 3) algorithm and implementation characteristics. These criteria are described below, along with a discussion of how they impacted the third-round candidate evaluations.

2.2.1 Security

As was the case for the past Advanced Encryption Standard (AES) and Secure Hash Algorithm 3 (SHA-3) competitions, security is the most important criterion that NIST uses when evaluating candidate post-quantum algorithms. NIST's public-key standards are currently utilized in a wide variety of applications, including internet protocols like TLS, SSH, IKE, IPsec, and DNSSEC, as well as for certificates, software code signing, and secure bootloaders. The new NIST public-key standards will provide post-quantum security for each of these applications.

For the purpose of quantifying the security of candidate algorithms, NIST gave three possible security definitions—two for encryption and one for signatures. NIST also designated five security strength categories for classifying the computational complexity of attacks that violate the security definitions (see [9]).

NIST also mentioned other desirable security properties, such as forward secrecy, resistance to side-channel and multi-key attacks, and resistance to misuse, all of which continue to be of interest. In some cases, NIST has encouraged submitters to make minor tweaks to provide or enhance these additional desirable security properties (e.g., by adding a public salt to ciphertexts to avoid multi-target attacks against KEMs).

For general-use encryption and key-establishment schemes, the Call for Proposals [9] asked for "semantically secure" schemes with respect to adaptive chosen ciphertext attack (equivalently, IND-CCA2 security). For ephemeral use cases, NIST also accepted algorithms that provided semantic security with respect to chosen plaintext attack (equivalently, IND-CPA security). IND-CCA2 security is not required in strictly ephemeral use cases and attempting to meet the more stringent requirements of IND-CCA2 security may incur significant performance penalties for some schemes. Digital signature schemes were required to provide existentially unforgeable signatures with respect to an adaptive chosen message attack (EUF-CMA security). Submitters were encouraged but not required to provide proofs of security in relevant models.

The five security strength categories defined in [9] were based on the computational resources required to perform certain brute-force attacks against the existing NIST standards for AES and SHA in a variety of different models of the cost of computation, both classical and quantum. In some cases, questions have arisen regarding whether various parameter sets meet their claimed security strength categories. The uncertainty arises principally from two distinct considerations.

First, the NIST security strength categories are defined in a way that leaves open the relative cost of various computational resources, including quantum gates, classical gates, quantum memory, classical memory, hardware, energy, and time. The idea is that in order to meet, for example, category 1, the best attack violating the security definition of a parameter set should cost more than a brute-force key search attack on a single instance of AES-128, according to any plausible assumption regarding the relative cost of the various computational resources involved in a real-world attack. Different opinions can therefore arise regarding what constitutes a plausible assumption regarding the relative cost of computational resources.

Second, even if one has agreed upon a model or a range of models for evaluating the relative cost of various computational resources, there may still be uncertainty how much of a given resource an attack actually requires. For example, many parameters of lattice reduction attacks (such as the BKZ block size, the number of required BKZ iterations, or the number of dimensions for free) are not proven optimal values but rather heuristic estimates based on simplified models, simulations, and mathematical conjectures. Additionally, while some submitters have rightly observed that many widely used cost models, such as the RAM model, underestimate the difficulty of certain memory intensive attacks, the comparative lack of published cryptanalysis using more realistic models may bring into question whether sufficient effort has been made to optimize the best-known attacks to perform well in these models.

Submitters were asked to provide a preliminary classification of all proposed parameter

sets according to the definitions of the five security strength categories. While category 1, 2, and 3 parameters were (and continue to be) the most important targets for NIST's evaluation, NIST nevertheless strongly encouraged the submitters to provide at least one parameter set that meets category 5. Aside from NTRU, all of the third-round submission packages contained parameters that claimed to meet category 5. In June 2021, at NIST's request, the NTRU team announced parameters designed to meet category 5 given the state of the art in lattice cryptanalysis [10].

During the first, second, and third rounds of the NIST standardization process, a number of cryptanalytic results dramatically reduced the security assumed for some submitted schemes and undermined NIST's confidence in the maturity of others. These results were the basis for many of NIST's decisions thus far in the process, particularly for Rainbow and G*e*MSS [11–13]. Cryptanalysis has also brought some of the candidates' security category claims into question or shown them to be false. In response, NIST may move some parameter sets down to a lower category (or refrain from standardizing them) if warranted.

Progress was also made in clarifying some outstanding security questions during the third round. In lattice-based cryptography, methods were developed to replace the asymptotic security estimates represented by the core SVP methodology with concrete security estimates expressed as a gate count that can be more directly compared with security estimates for the non-lattice candidates (see [14, 15], as well as discussion on the pqc-forum [16]). Several of the finalists have also been implemented with countermeasures to side-channel attacks (see Section 2.2.3). Additionally, further investigations have been performed to determine whether the BIKE submission's estimate of its decryption failure rate is accurate enough to justify a claim of IND-CCA2 security [17, 18].

NIST continues to see diversity of computational hardness assumptions as an important long-term security goal for its standards. NIST will standardize practically efficient schemes from different families of cryptosystems to reduce the risk that a single breakthrough in cryptanalysis will leave the world without a viable standard for either key-establishment or digital signatures. Nonetheless, NIST does not feel the need to establish these standards all at once but will rather prioritize those schemes that seem closest to being ready for standardization and wide adoption. NIST feels that this strategy balances the desire for diversity with the need for all standards to be thoroughly vetted before they are released.

2.2.2 Cost and Performance

The original call for proposals [9] identified cost as the second most important criterion when evaluating candidate algorithms. Cost includes the computational efficiency of key generation and public and private key operations, the transmission costs for public keys and signatures or ciphertexts, and the implementation costs in terms of RAM (random-access memory) or gate counts.

During the third round of the NIST PQC Standardization Process, more information about the computational efficiency of the finalists became available. Faster, constant-time

implementations were provided for many of the algorithms (e.g., [19–26]), as were implementations that focused on limiting memory usage (e.g., [27–31]). More information about many of the alternate candidates became available as well. This section focuses on the cost and performance considerations that factored into NIST's selections.

When comparing the overall performance of the algorithms, both computational cost and data transfer cost were considered.[2] For general-purpose use, the evaluation of overall performance considered the cost of transferring the public key in addition to the signature or ciphertext during each transaction. For KEMs, the cost of key generation was also taken into account, since many applications use a new KEM key pair for each transaction to provide forward secrecy. For signature algorithms, the cost of key generation was considered less important.

At the end of the second round of the NIST PQC Standardization Process, NIST selected KYBER, NTRU, and Saber as finalists for the selection of a general-purpose KEM and indicated an intention to select at most one of them [7]. All three have good performance on both x86-64 processors with AVX2 extensions [33, 34] and the ARM Cortex-M4 [35]. The overall performance of NTRU is not quite as good as KYBER or Saber as a result of its slower key generation and somewhat larger public keys and ciphertexts. However, the overall performance of any of these KEMs would be acceptable for general-use applications.

Figure 1 shows the computational performance numbers from [33] for the x86-64 processor with AVX2 extensions for KYBER, NTRU, and Saber for security categories 1 and 3.[3] Figure 2 shows the "total costs" for KYBER, NTRU, and Saber when the cost of data transmission is added. Figure 2 was generated using an estimated cost of 2000 cycles/byte.

Encapsulation and decapsulation is very fast with all three schemes. While Saber has the lowest total cost due to its smaller public keys and ciphertexts, the cost difference between KYBER and Saber was not large enough to be considered significant.

The cost of key generation for ntruhps2048677 or ntruhrss701 is about 11 times as much as for KYBER512. However, as Figure 2 shows, the total cost for using these schemes tends to be dominated by the cost of data transmission, and so most of the difference in the total cost of the NTRU parameter sets compared to KYBER and Saber is because of NTRU's somewhat larger public keys and ciphertexts. As a result, the total cost for ntruhps2048677 is less than 30% greater than for KYBER512. In addition, since the public keys and ciphertexts for the category 1 and 3 parameter sets for all three of the schemes are likely to fit within a single internet packet, their performance numbers may be considered comparable. It may also be noted that, according to [33], the cost for key generation for ntruhps2048677 or ntruhrss701 is comparable to the cost of key generation for the elliptic curve cryptography curve P-256, which is widely used for ephemeral key exchange.[4]

[2] The figures below use an estimate of 2000 cycles/byte for data transmission costs as an example; however, the most appropriate conversion factor will vary greatly depending on the use case [32], so the costs of the different candidates were considered using several different cycles/byte cost estimates.

[3] [34] reports similar computational performance numbers for the candidates.

[4] This was also highlighted by Daniel J. Bernstein on the PQC Forum mail list [36].

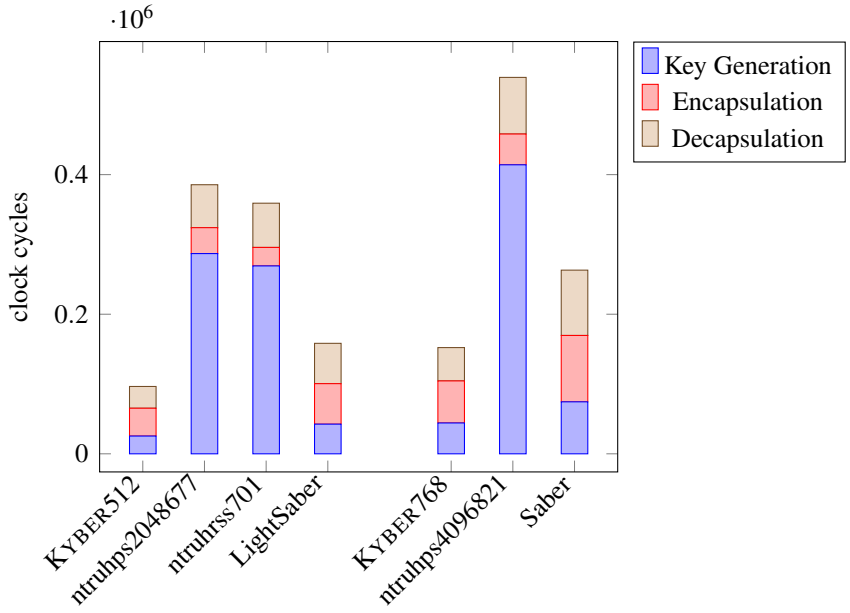

Figure 1. KEM Benchmarks on x86-64 processors with AVX2 extensions

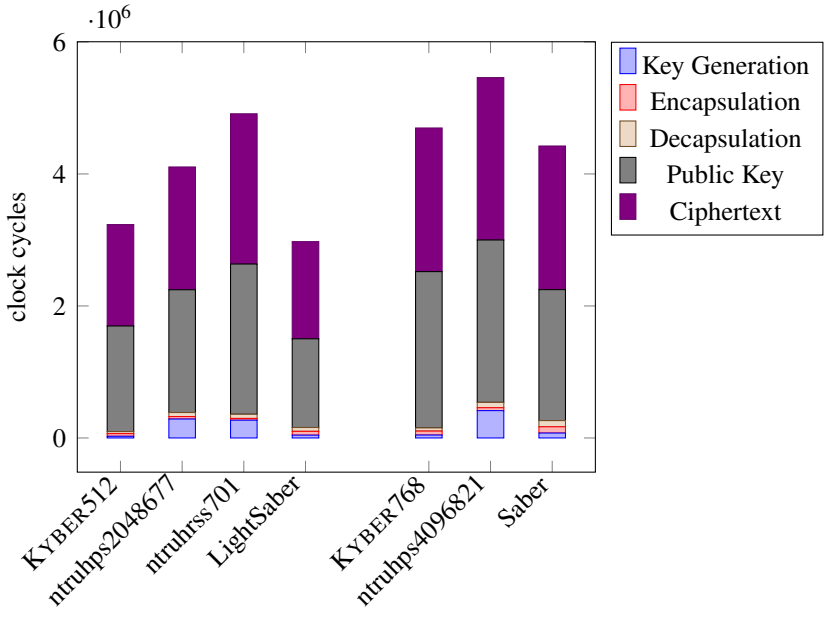

Figure 2. KEM Benchmarks on x86-64 processors with AVX2 extensions with 2000 cycles/byte transmission costs

Figure 3 shows the computational performance numbers from [35] for the ARM Cortex-M4 and Figure 4 shows the "total costs" when an estimated 2000 cycles/byte transmission cost is added. While in the case of the x86-64 processor the total cost is dominated by the cost of transmitting data, with the ARM Cortex-M4, using the same cycles/byte estimate, the cost of computation is a much more significant part of the total cost, especially the cost of key generation with the NTRU parameter sets. As a result, the total costs for ntruhps2048677 and ntruhrss701 are more than twice as much as for KYBER512. However, most of the extra cost is a result of NTRU's slower key generation, and constrained devices are less likely to be used to perform a new key generation for every transaction. If the cost of key generation were removed from the total cost, then the total cost of ntruhps2048677 would be less than 30% greater than for KYBER512. Consequently, the performance difference between NTRU and KYBER or Saber that would actually be experienced on constrained devices would likely be much less than is depicted in Figures 3 and 4.

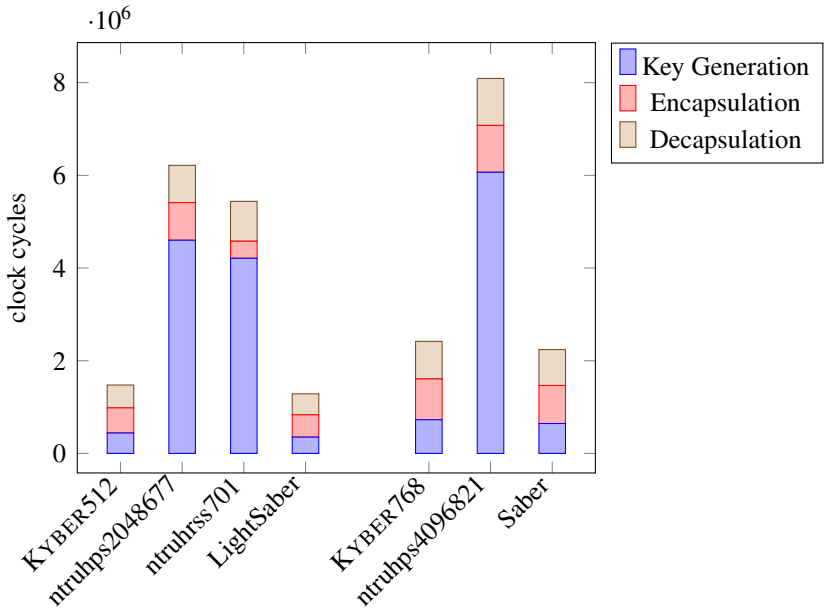

Figure 3. KEM Benchmarks on ARM Cortex-M4 processor

The pqm4 benchmark results [35] show that both KYBER and Saber are suitable for use on constrained devices, as each of these can be implemented (at least without protections against side-channel attacks) using less than 4 KiB of RAM with less than 20 KiB of storage for the code. While the specific implementation of NTRU in [35] may not be suitable for use on constrained devices, it is likely that efficient implementations for constrained devices of the NTRU parameter sets submitted to the NIST PQC Standardization Process are possible given that other NTRU parameter sets have been efficiently implemented on constrained devices [37–39].

There have been many hardware and hybrid hardware-software implementations of various candidates in the third round [40–53]. The benchmarks from [46] show results from

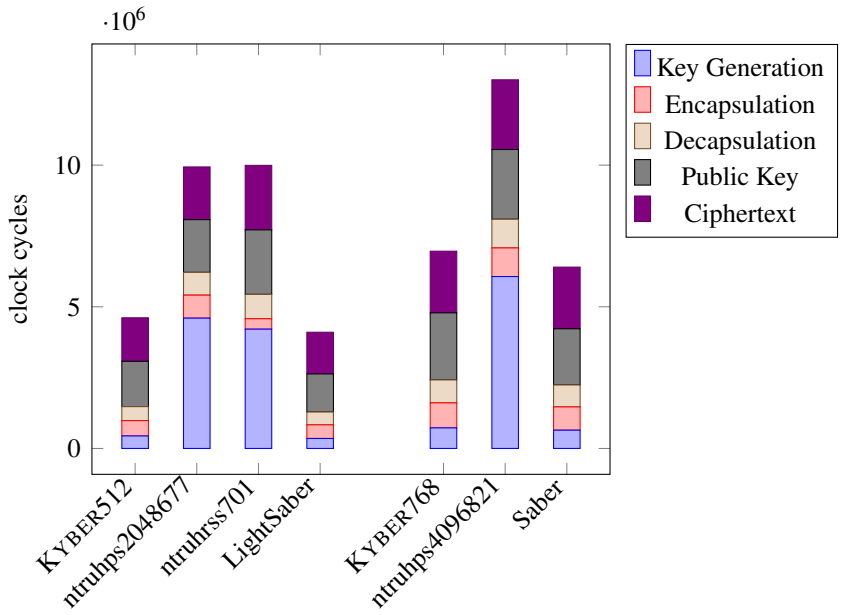

Figure 4. KEM Benchmarks on ARM Cortex-M4 processor with 2000 cycles/byte transmission costs

multiple high-speed FPGA implementations of KYBER, NTRU, and Saber. As with the benchmark results for the x86-64 and ARM Cortex-M4, KYBER and Saber have fairly comparable performance, with KYBER requiring somewhat fewer hardware resources. NTRU requires similar hardware resources to Saber and has comparable encapsulation speed, but decapsulation is a little slower and key generation much slower. Overall, however, the performance numbers for all three schemes again show that they would be suitable for most use cases.

Classic McEliece was also selected as a finalist at the end of the second round [7]. Classic McEliece has a performance profile that differs from the other KEMs under consideration and, as a result, its performance was not directly compared to the performance of the other KEMs. Classic McEliece has slow key generation and very large public keys, but its encapsulation and decapsulation speeds are comparable to those of the structured-lattice KEMs, and it has very small ciphertexts. As a result, Classic McEliece may provide the best performance in applications where the cost of key generation and public key transmission are not considered part of the transaction cost (e.g., [54]), but its total cost would be much greater than any of the other candidate KEMs if the cost of transmitting the public key were included.

The Second Round Status Report selected Dilithium and FALCON as finalists for a general-purpose signature scheme and indicated an intention to select at most one of them [7]. The third finalist, Rainbow, while having an attractive performance profile for applications requiring small signatures or fast verification, suffered security losses which are described in Section 4.5.3; thus, the performance numbers of Rainbow will be omitted in

the following.

Figure 5 shows the computational performance numbers from [33] for the x86-64 processor with AVX2 extensions for Dilithium and FALCON. Unlike Figure 1, the figure does not include the cost of key generation since signature keys are not generated on a per-transaction basis. Figure 6 shows the "total costs" for Dilithium and FALCON when the cost of transmitting the public key and signature is added. As with Figures 2 and 4, an estimated cost of 2000 cycles/byte is used. When using the x86-64 processor, signature generation with Dilithium is slightly faster than with FALCON. However, data transmission dominates the total costs of using these schemes, so FALCON's total cost is lower due to its smaller public key and signature sizes. For most applications using an x86-64 or similar processor, the performance numbers for either Dilithium or FALCON should be acceptable. However, unlike FALCON signatures, Dilithium signatures cannot fit within a single internet packet, so this may make adapting some applications to use Dilithium more difficult than adapting them to use FALCON (e.g., [55, 56]).

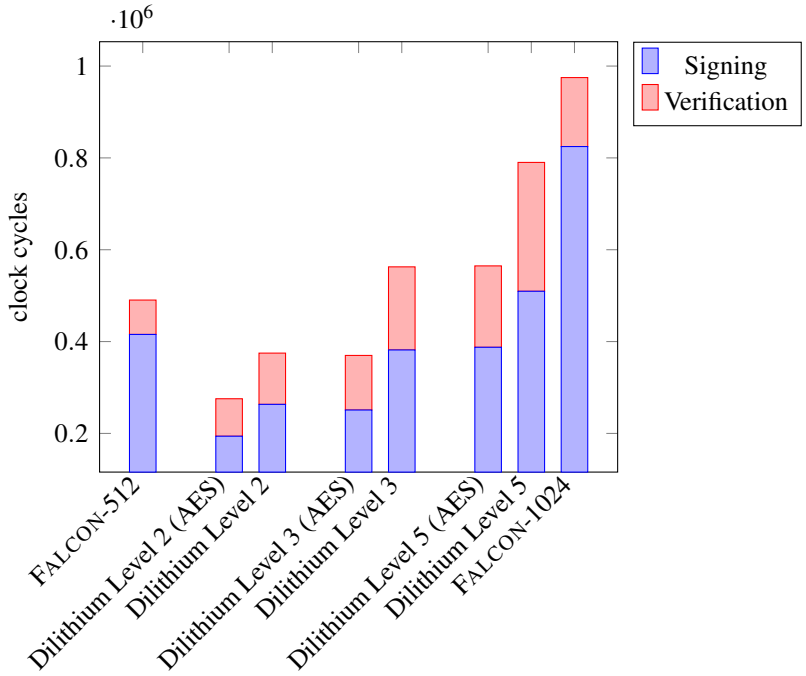

Figure 5. Signature Benchmarks on x86-64 processor with AVX2 extensions

Figure 7 shows the computational performance numbers from [35] for the ARM Cortex-M4 processor for the security category 1, 2, and 3 parameter sets of Dilithium and FALCON parameter sets. Figure 8 shows the "total costs" when an estimated 2000 cycles/byte transmission cost is added. As the ARM Cortex-M4 does not have support for floating-point operations, signature generation using FALCON is much slower than signature generation using Dilithium, and the difference is great enough that the total cost of using Dilithium is lower even when Dilithium's higher data transmission costs are taken into account.

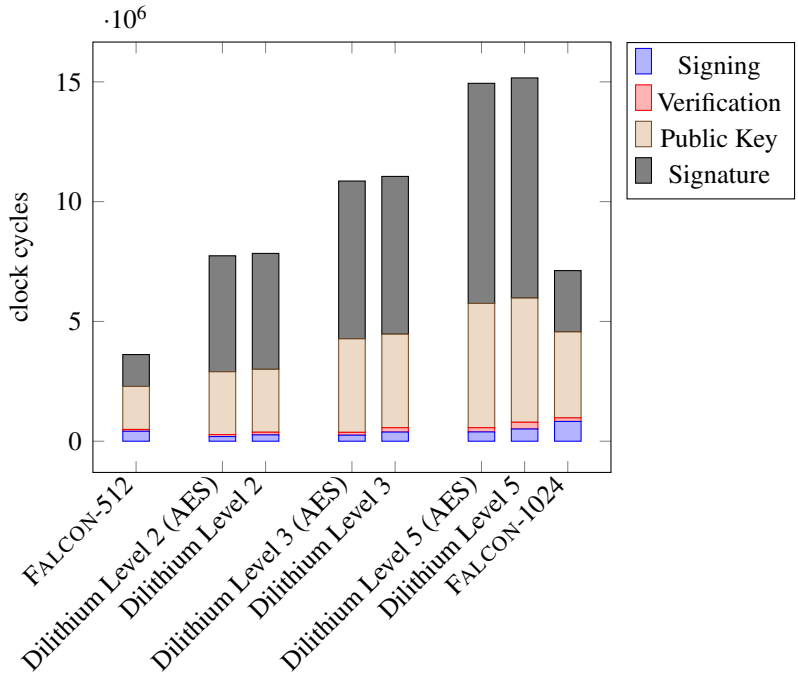

Figure 6. Signature Benchmarks on x86-64 processor with AVX2 extensions with 2000 cycles/byte transmission costs

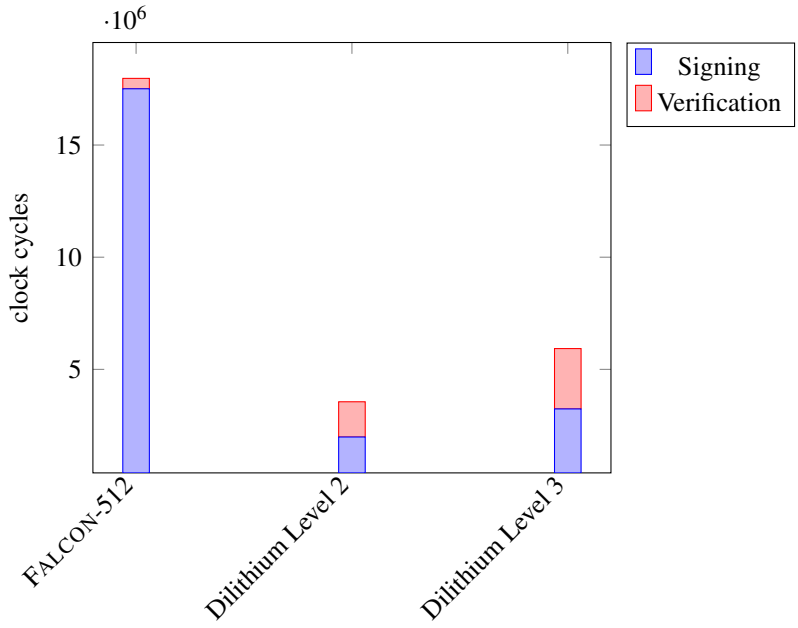

Figure 7. Signature Benchmarks on ARM Cortex-M4 processor

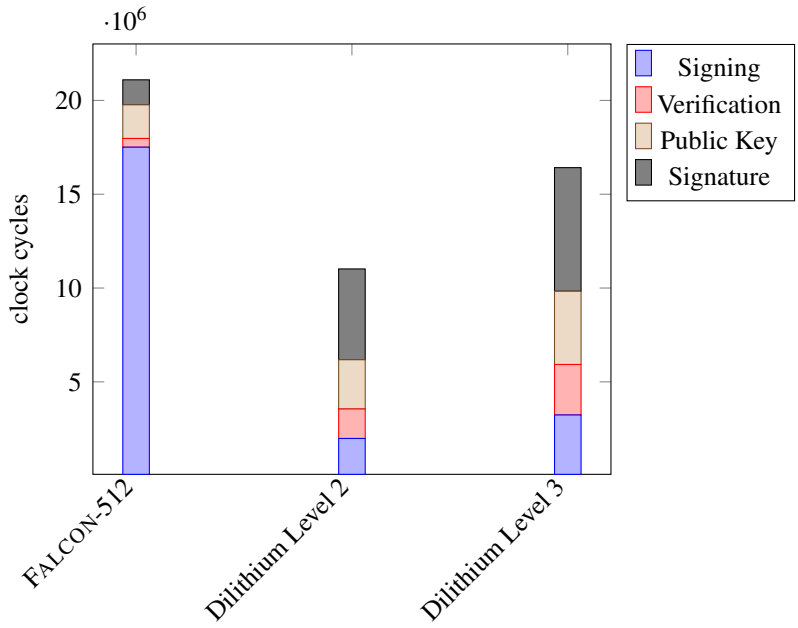

Figure 8. Signature Benchmarks on ARM Cortex-M4 processor with 2000 cycles/byte transmission costs

For the digital signature schemes, [27] demonstrated that signature verification for each of the finalists could be implemented using less than 8 KiB of RAM and with less than 8 KiB of storage for the code, and [57] presented FPGA implementations of signature verification for both Dilithium and FALCON. However, whereas key generation and signing with Dilithium may be implemented using less than 9 KiB of RAM [30], FALCON appears to require significantly more RAM [58], which may make FALCON infeasible to implement on constrained devices, such as smart cards [59]. Furthermore, while a few hardware implementations of Dilithium were developed during the third round [22–24, 57], [22] notes that FALCON lacks any reported hardware implementations, which suggests that FALCON key and signature generation may be relatively difficult to implement in constrained environments.

Figures 9 and 10 show the benchmark numbers from [34] for security categories 1 and 3 for the KEM alternate candidates BIKE, FrodoKEM, HQC, NTRU Prime, and SIKE.[5] As with KYBER, NTRU, and Saber (see Figure 2), with the exception of SIKE, the total cost for using these schemes on x86-64 processors is dominated by the cost of data transmission. NTRU Prime's performance is comparable to that of NTRU. In general, BIKE and HQC have faster overall performance than either FrodoKEM or SIKE. Using a metric of 2000 cycles/byte, SIKE has somewhat better overall performance than FrodoKEM. However, for many use cases the cost of data transmission relative to computation will be lower, and

[5] According to [60], the NTRU Prime parameter sets ntrulpr857 and sntrup857 may belong in either security category 2 or 3.

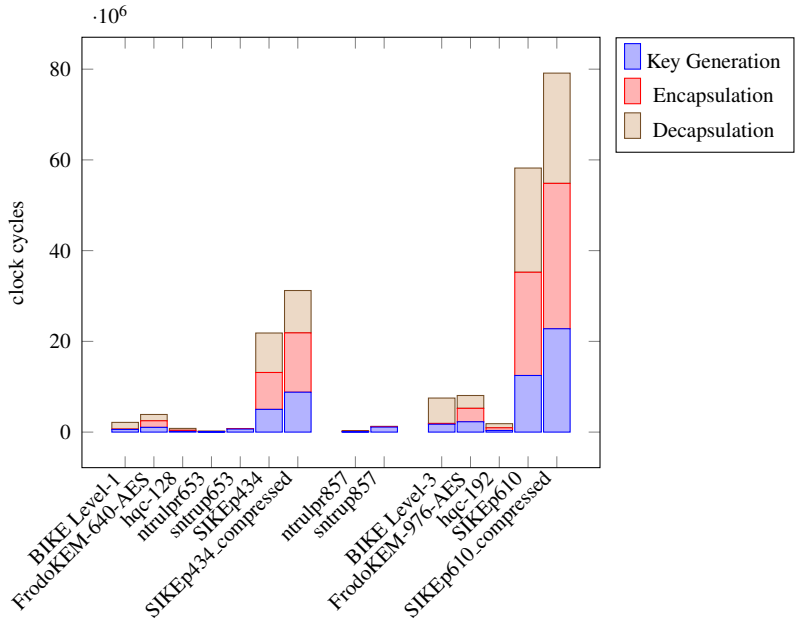

Figure 9. KEM Alternates Benchmarks on x86-64 processor

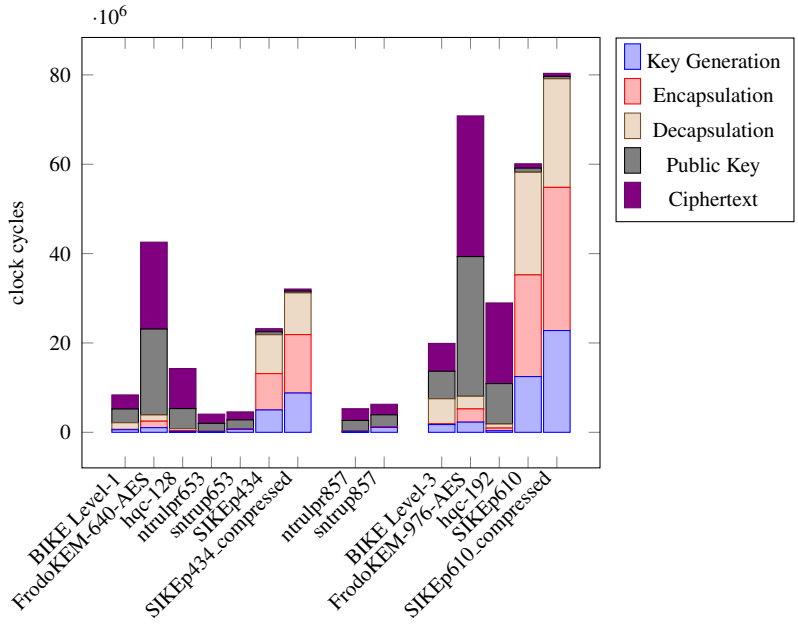

Figure 10. KEM Alternates Benchmarks on x86-64 processor with 2000 cycles/byte transmission costs

FrodoKEM will provide better overall performance.

2.2.3 Algorithm and Implementation Characteristics

In considering other evaluation criteria beyond security and cost and performance, the original Call for Proposals [9] also listed various desirable algorithm and implementation characteristics. The specific characteristics mentioned were flexibility, simplicity, and adoption (the absence of factors that could hinder adoption). Note that this list was not meant to be all-encompassing. NIST hoped that careful attention would be paid to the finalists, as they were the algorithms that would most likely be ready for standardization at the conclusion of the third round.

The third-round candidates were allowed to make small changes to their specifications. Most of these changes were geared toward fixing minor issues that had been noticed during the second round, or to clarify or simplify the submission specification. In addition, some algorithms introduced additional parameter sets to demonstrate greater flexibility. No major redesigns or changes were allowed.

The Status Report on the Second Round [7] made particular mention of side-channel analysis. Historically (dating back to the AES standardization process), most side-channel analyses have been performed in the decades after the point of standardization. However, for the post-quantum cryptography standardization process, NIST asked the community to contribute side-channel analyses earlier in the standards cycle. During the third round (and before), the community responded with a large number of papers and other technical works that considered both side-channel attacks on the candidates, as well as ways of defending implementations against these attacks (see, for example, [61–92], or for a survey, see [93]). At the Third NIST PQC Standardization Conference, there were also several presentations on side-channels that mostly focused on the lattice-based KEM candidates KYBER, NTRU, and Saber [94–100].

NIST notes that future engineers and researchers will undoubtedly benefit from this initial study into post-quantum side-channel analyses. An initial desire had been to find, where possible, any algorithmic characteristics that would facilitate (or harm) the future deployment of side-channel-resistant implementations of any candidate-algorithm. In particular, NIST sought out any "distinguishing information" in the realm of side-channel analyses that would especially indicate a reason for NIST to prefer one of the finalists over the others. However, after extended study, the differences in the difficulty of protecting the candidate algorithms against side-channels appear to be small. NIST strongly appreciates the community's efforts in this line of work. It is NIST's hope and expectation that more such work will continue, especially with regard to protecting the implementations of the algorithms announced for standardization.

Another important characteristic of candidates is their potential performance impact in existing widely used protocols (e.g., TLS, IPSec, and SSH) and certificates. The 3rd Round saw some real-world experiments to see if there would be any performance problems arising from any of the algorithms (see, for example, [101–108]). NIST observed that the

structured lattice finalists for both KEMs and signatures could be substituted into these protocols for existing public-key algorithms with relatively small (or no) performance loss.

While it is hard to measure simplicity concretely, simpler designs are preferable when comparing two similar schemes. In particular, simplicity was an important factor in NIST's evaluation of FALCON, with the concern that the use of floating point arithmetic and more complex implementation could lead to errors that might affect security. In contrast, the simpler design of Dilithium was viewed positively.

NIST believes it is important to select cryptographic standards that will be capable of protecting sensitive government information as well as being widely adopted for use in industry. In selecting a cryptographic algorithm for standardization, an evaluation factor is whether a patent might hinder adoption of the cryptographic standard. All submission teams were required to submit statements regarding knowledge of patents involving their algorithms and implementations. Such statements are available at the NIST PQC website, https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

In addition, NIST has engaged with third parties that own various patents directed to cryptography, and NIST acknowledges cooperation of ISARA, Philippe Gaborit, Carlos Aguilar Melchor, the laboratory XLIM, the French National Center for Scientific Research (CNRS), the University of Limoges, and Dr. Jintai Ding. NIST and these third parties are finalizing agreements such that the patents owned by the third parties will not be asserted against implementers (or end-users) of a standard for the selected cryptographic algorithm. NIST appreciates the efforts of those who helped obtain this outcome and the cooperation of the third parties.

2.3 Selection of the Candidates for Standardization and Fourth Round

This section describes how we made decisions for standardization, and for the algorithms moving to the fourth round. During the third round, there were some cryptanalytic results that had a significant effect on NIST's selections. An attack on G*e*MSS [109] dramatically reduced its security and undermined NIST's confidence in its maturity. This result led to the elimination of G*e*MSS from being considered for standardization by NIST.

Rainbow also suffered significant attacks during the third round [11, 13]. The first attack, early in the third round, caused parameter sets to lose between 20 to 55 bits of security in the RAM model, with the higher security parameter sets losing more bits of security. This was followed by a more severe attack late in the third round that yielded private key recovery for security category 1 parameters in a little over two days of computation time on a single laptop. Lacking confidence in its security, NIST did not select Rainbow for standardization.

NIST also decided to remove FrodoKEM, NTRU Prime and Picnic from consideration for standardization. FrodoKEM is a lattice-based candidate that had been chosen as an alternate during the second round. FrodoKEM is mainly distinguished by the fact that it does not rely on structured lattices (in contrast to the finalists KYBER, NTRU, and Saber). While NIST intends to select at least one additional KEM not based on structured lattices

for standardization after the fourth round, three other KEM alternates (BIKE, HQC, and SIKE) are better suited than FrodoKEM for this role. FrodoKEM has generally worse performance than these three and so will not be considered further for standardization. NTRU Prime was also advanced as an alternate since it was viewed as less promising in comparison to the finalists. There were no results during the third round that significantly altered that view. As NIST will standardize one of the (structured lattice) finalist KEMs, NTRU Prime was not selected to continue on in the process. There was a similar situation for the signatures. Picnic was not selected because NIST is choosing to standardize SPHINCS$^+$. Picnic and SPHINCS$^+$ have similar performance profiles (small public keys and large signatures) and would be suitable for the same use cases. SPHINCS$^+$ and Picnic both have several versions, making a direct comparison of cost and performance more involved (see Figures 11 and 12, and Table 9 for a comparison of some parameter sets). However, they each have much higher cost and much worse performance in comparison to Dilithium and FALCON, making these criteria less important. The security of Picnic is not better than that of SPHINCS$^+$, and NIST feels that while SPHINCS$^+$ is a mature design, Picnic and related schemes would continue to benefit from future research and improvements.

When choosing between similar KEM algorithms, cost and performance were significant selection criteria. As noted in Section 2.2.2, both data transmission costs and computational efficiency were taken into account when comparing candidates. NIST considered benchmarks provided by the community (see, for example, [33, 35, 110–112]) across multiple platforms when determining computational efficiency.

One of the difficult choices NIST faced was deciding between KYBER, NTRU, and Saber. All three were selected as finalists and were very comparable to each other. NIST is confident in the security that each provides. Most applications would be able to use any of them without significant performance penalties. At stated at the conclusion of the second round, NIST intended to standardize only one of these finalists, as all three were based on structured lattices. Issues relating to patents were a factor in NIST's decision during the third round as NIST became aware of various third-party patents. As noted in Section 2.2.3, NIST negotiated with several third parties to enter into various agreements to overcome potential adoption challenges posed by third-party patents.[6] One of the differences between KYBER, Saber, and NTRU is the specific security assumption each relies upon for security. NIST finds the MLWE problem, which KYBER depends upon, marginally more convincing than the other assumptions like MLWR or the NTRU problem. NIST also appreciated the KYBER team's specification, which included a thorough and detailed security analysis. With regard to performance, KYBER was near the top (if not the top) in most benchmarks.

The rest of the KEM candidates selected (BIKE, Classic McEliece, HQC, SIKE) will all continue to be evaluated in the fourth round. Both BIKE and HQC are based on structured codes and would be suitable as a general-purpose KEM that is not based on lattices. NIST may select at most one of these two candidates for standardization at the conclusion of the

[6] NIST expects to execute the various agreements prior to publishing the standard. If the agreements are not executed by the end of 2022, NIST may consider selecting NTRU instead of KYBER. NTRU was proposed in 1996, and U.S. patents were dedicated to the public in 2007.[113].

fourth round. SIKE remains an attractive candidate for standardization because of its small key and ciphertext sizes. NIST hopes that further study will continue on SIKE during the fourth round. Classic McEliece was a finalist, but is not being standardized by NIST at this time. Although it is widely regarded as secure, NIST does not yet anticipate it being widely used because of its large public key size . Thus, there is no urgency to standardize Classic McEliece yet.

In [7], NIST indicated an intent to select at most one of Dilithium and FALCON, as both are based on structured lattices and could be used in most applications. Ultimately, however, NIST decided to select both schemes for standardization. As noted in Section 2.2.2, key and signature generation for FALCON appears to require more resources (gates and RAM) than Dilithium, which may make FALCON unsuitable for implementation on constrained devices, particularly in cases in which protection against side-channel attacks is required. In addition, NIST recognizes that the simpler design of Dilithium's key and signature generation will help ensure secure implementations. For these reasons, NIST selected Dilithium as the primary signature algorithm that it will recommend for general use and will prioritize its standardization.

NIST understands that some applications will not work as they are currently designed if the signature and the data being signed cannot fit in a single internet packet. For these applications, the implementation complexity of FALCON's signature generation may not be a concern, but the difficulty of modifying the applications to work with Dilithium's larger signature size may create a barrier to the transition to post-quantum signature schemes. For this reason, NIST decided to standardize FALCON as well. Given FALCON's overall better performance when signature generation does not need to be performed on constrained devices, many applications may prefer to use FALCON over Dilithium, even in cases in which Dilithium's signature size would not be a barrier to implementation.

In order to not rely entirely on the security of lattices, NIST is also standardizing SPHINCS$^+$. The security of SPHINCS$^+$ is well-understood, although it is much larger and slower than the lattice signatures. SPHINCS$^+$ is a mature scheme, and standardizing it creates a fallback option that helps minimize the risk that a single breakthrough in cryptanalysis would leave NIST without a viable signature. NIST recognizes that SPHINCS$^+$ may not be suitable for many applications, given its performance profile. NIST made the choice to select SPHINCS$^+$ now instead of perhaps including it in the fourth round. As such, this means the end of the current process for signature schemes. All signature candidates have either been selected for standardization or removed from consideration for standardization. NIST may standardize more signatures in the future,[7] but this will take several years and there is no guarantee of better algorithms.

In summary, NIST has selected four of the third-round candidates for standardization and four to advance to a fourth round for further evaluation and study. See Tables 4 and 5 for a list of these algorithms.

[7] NIST plans to issue a new Call for Proposals for post-quantum signatures later in 2022.

Table 4. Algorithms to be Standardized

Public-Key Encryption/KEMs	Digital Signatures
CRYSTALS–KYBER	CRYSTALS–Dilithium
	FALCON
	SPHINCS$^+$

Table 5. Candidates advancing to the Fourth Round

Public-Key Encryption/KEMs	Digital Signatures
BIKE	
Classic McEliece	
HQC	
SIKE	

3. Preliminary Information

The following preliminary information is given in advance of the summary of candidates to introduce some computational and security concepts (and history) that will be referenced throughout the subsequent section. This section will also serve to reduce redundancy as some of the candidates' security analyses have properties in common. This section is not intended to be an exhaustive security or literature review.

3.1 Computational Models

When selecting secure parameters for cryptosystems, the cost of the best-known attacks must be understood and estimated. There are several variables involved in assessing the cost of an actual attack, such as monetary cost of equipment and energy, number of operations needed to complete the attack, size of required memory, and time to read from or write to memory. Thus, the cost of an attack varies depending on the metric(s) selected for evaluation. Appendix B describes several cost models used in the literature and discusses assumptions and considerations for each.

3.2 Underlying Security Problems

This section presents some of the hard computational problems that are common to multiple code-based, multivariate-based or lattice-based schemes examined in the course of the NIST PQC Standardization Process. Other hard computational problems will be mentioned as needed in the individual candidate summaries in Section 4.

3.2.1 Code-based

The difficulty of the general- and syndrome-decoding problems (and some variants thereof) is a component of the security argument for the three code-based KEMs moving to the 4th round: BIKE, Classic McEliece, and HQC. All three schemes provide an IND-CPA secure PKE with proofs that depend on (a variant of) one of these two computational problems.

Let C be an (n,k) binary linear code. Let $\mathbb{F}_2$ denote the finite field of two elements. Then the set of 2^k codewords of C form a k-dimensional subspace of $\mathbb{F}_2^n$. For any vector $v \in \mathbb{F}_2^m, m \in \mathbb{N}$, let $|v|$ denote the Hamming weight of v.

Problem 3.1 ((Decisional) Syndrome Decoding problem) *Given an $(n-k) \times n$ parity-check matrix H for C, a vector $y \in \mathbb{F}_2^{n-k}$, and a target $t \in \mathbb{N}$, determine whether there exists $x \in \mathbb{F}_2^n$ that satisfies $Hx^T = y$ and $|x| \leq t$.*

Problem 3.2 ((Decisional) Codeword Finding problem) *Given an $(n-k) \times n$ parity-check matrix H for C and a target $w \in \mathbb{N}$, determine whether there exists $x \in \mathbb{F}_2^n$ that satisfies $Hx^T = 0$ and $|x| = w$.*

For a general binary linear code C, these two problems were shown to be NP-complete by Berlekamp, McEliece, and van Tilborg [114]. This does not guarantee that any given cryptographic instantiation of the problem is hard.

The most effective known attacks against code-based KEMs are based on information set decoding (ISD). This approach ignores the structure of the binary code and seeks to recover the error vector based on its low Hamming weight. These techniques originated with Prange's algorithm in 1962 [115] and have since undergone a series of improvements (e.g., [116–129]). The net effect of all these improvements has been fairly modest, and most of the changes in concrete security were due to results from more than 30 years ago. Quantum versions of ISD algorithms have also been studied [130–133]. These results represent a generic Grover-based speedup of classical ISD algorithms and indicate that ISD can be sped up nearly as much as brute-force search.

A few recent papers [134–136] have attempted to provide concrete security estimates for the parameter sets submitted to the NIST PQC Standardization Process based on the above classical and quantum ISD attack papers. The analysis of [134] gave an anomalously low estimate for the cost of the MMT attack [122]. The subsequent analysis of [135] determined that the previous analysis was in error and gave corrected estimates for the cost of several attacks (including MMT) in a variety of memory cost models. In [136] a software implementation was used to attempt to determine an appropriate memory cost model; however, since computationally intensive tasks typically benefit more from specialized hardware support than memory intensive tasks, this approach may underestimate the relative cost of memory access.

In a multi-ciphertext setting, a further improvement is possible, reducing the cost of decoding a single ciphertext by a factor equal to approximately the square root of the number of ciphertexts [137].

3.2.2 Multivariate-based

The security arguments for the two multivariate signature schemes, G*e*MSS and Rainbow, depend on the difficulty of the $\mathcal{MQ}$ problem and the MinRank problem.

Problem 3.3 ((Decisional) Multivariate Quadratic ($\mathcal{MQ}$) polynomial problem) *Given a finite field $\mathbb{F}$ and a system of m quadratic polynomials of n variables x_i:*

$$f_k(x_1,\ldots,x_n) = \sum_{1\le i\le j\le n} a_{ij}^{(k)} x_i x_j + \sum_{1\le i\le n} b_i^{(k)} x_i + c^{(k)} = 0,$$

for k from 1 to m, where $a_{ij}^{(k)}, b_i^{(k)}, c^{(k)}$ are all in $\mathbb{F}$, determine if there exists a solution in $\mathbb{F}^n$.

Problem 3.4 ((Decisional) MinRank problem) *Given a finite field $\mathbb{F}$, k matrices M_i of size $m \times n$ with entries in $\mathbb{F}$, and a rank bound r, determine if there exist values $c_i \in \mathbb{F}$ to satisfy the following equation:*

$$rank\left(\sum_{i=1}^{k} c_i M_i\right) \le r.$$

The $\mathcal{MQ}$ problem was shown to be NP-hard in all fields in [138]. The MinRank problem was shown to be NP-hard in [139]. It is important to note that when the target rank r is fixed that the MinRank problem has polynomial complexity; thus, multivariate cryptosystems typically require a large value of r for any associated MinRank instance. Neither problem is known to be hard in the average case[8], nor does the NP-hardness imply that instances arising from cryptographic schemes are intractable.

The most effective known generic attacks on the $\mathcal{MQ}$ problem include Gröbner basis algorithms such as F4/F5, see [140, 141], and linearization algorithms such as XL, see [142]. The most effective attacks for MinRank vary depending on the size and number of matrices and the target rank. The main methods include combinatorial search methods, pioneered in [143], and the support minors method, see [144].

3.2.3 Lattice-based

Seven of the 15 third-round candidates are lattice-based cryptosystems.[9] These cryptosystems are connected to a large body of academic research, which emphasizes (asymptotic) provable security based on the worst-case hardness of lattice problems (via worst-case-to-average-case reductions). An early milestone in this line of research was a 1996 paper by Ajtai [145], which defined the short integer solution (SIS) problem, and related its average-case complexity to the worst-case hardness of finding short vectors in every integer lattice, giving lattice-based one-way functions and lattice-based trapdoor functions.

[8] In this document, "hard in the average" means hard with overwhelming probability on random inputs.
[9] Dilithium, FALCON, FrodoKEM, KYBER, NTRU, NTRU Prime, and Saber

Concurrently in 1996, Hoffstein, Pipher, and Silverman [146] (with publication in 1998) described the NTRU public-key encryption system and the related ring-based NTRU problem from which it draws its security. As observed in that early work, the most direct mechanism by which to attack the system is based on lattice algorithms.

Later, in 2005, the complexity-theoretic connection between public key encryption candidates and computationally hard problems on lattices was formalized in a seminal paper by Regev [147]. There, Regev defined the learning with errors (LWE) problem as a basis for a public-key encryption scheme, and asymptotically related the quantum security of that system to the worst-case hardness of finding short vectors in lattices, a problem known as the (approximate) Shortest Vector Problem (SVP). Solving SVP in general lattices (with sufficiently small approximation factors) is NP-hard under randomized reductions. However, practical lattice-based cryptosystems involve approximate SVP instances that are outside the regime that is known to be NP-hard [148]. In addition, these NP-hardness results only describe the worst-case asymptotic complexity of the problem and are not known to apply to algebraically structured lattices.

Miccancio [149] introduced a ring-based analogue of Ajtai's SIS problem in 2002. A search variant of ring-based LWE (and an associated public-key encryption scheme, relying on the Goldreich-Levin hardcore function [150]) was introduced by Stehle, Steinfeld, Tanaka, and Xagawa [151] in 2009. A decisional variant of Ring-LWE with associated public-key encryption scheme (and an associated search-to-decision reduction) was introduced by Lyubashevsky, Peikert, and Regev [152] in 2010. Further, an algebraically-structured (and in particular, module-based) formulation of SIS/LWE-type problems – which can be syntactically viewed as interpolating between the original integer-based presentation and the later polynomial-ring-based presentations – was first introduced by Brakerski, Gentry, and Vaikuntanathan [153] in 2011 under the name *General Learning With Errors*.

In 2012, an efficient reconciliation-based mechanism for constructing a simple and provably secure key exchange scheme from LWE was discovered by Ding, Xie, and Lin [154]. This work can be viewed as discovering an analogue of the Diffie-Hellman key exchange but with errors. Later work by Peikert [155] and Alkim et al. [156] also proposed a reconciliation mechanism and a generalization, respectively.

Finally, the learning with rounding (LWR) problem was introduced by Banerjee, Peikert, and Rosen [157] in 2012 in order to construct the first (non-generic) pseudorandom functions from LWE. This has since been re-purposed to construct efficient candidate key exchange systems.

In the following, the various underlying security problems for each of these systems are briefly described:[10]

Problem 3.5 (The Short Integer Solution ($SIS_{n,m,q,\beta}$) problem) *Let n, m, q be positive integers, and let β be a positive real number. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, chosen uniformly*

[10]For a more detailed explanation, see for instance [148].

at random, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ *of Euclidean norm* $||\mathbf{z}|| \leq \beta$ *such that* $\mathbf{Az} = \mathbf{0} \in \mathbb{Z}_q^n$.

Several variants of the "NTRU problem" have been studied in the literature [158–161] (see also references to the "DSPR assumption" [162]). The precise definition of this problem varies somewhat, depending on the context, but a typical definition is as follows:

Problem 3.6 (The $Search-NTRU_{R,q,\mathcal{D},\gamma}$ **problem)** *Let* q *be a positive integer,* γ *be a positive real number, and* R *be a ring of the form* $R = \mathbb{Z}_q[x]/\Phi$ *(where* Φ *is a monic polynomial). Given an element* $h \in R$ *drawn from some distribution* $\mathcal{D}$*, such that there exists nonzero* $(f,g) \in R^2$ *that satisfy* $h \cdot f = g \bmod q$ *and have small Euclidean norms* $||f||, ||g|| \leq \sqrt{q}/\gamma$*, find such a pair* (f,g).

The next few problems are all types of *Learning With Errors (LWE)* problems. For a vector $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution χ, define the Learning with Errors (LWE) distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \leftarrow \chi$ over $\mathbb{Z}$, and outputting the pair $(\mathbf{a}, b)$ where $b = \langle \mathbf{s}, \mathbf{a} \rangle + e \bmod q$.

Problem 3.7 (The $Search\text{-}LWE_{n,m,q,\mathcal{B},\chi}$ **problem)** *Let* $\mathbf{s} \in \mathbb{Z}_q^n$ *be chosen from some distribution* $\mathcal{B}$*. Given* m *samples* $(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_m, b_m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *drawn independently at random from the distribution* $A_{\mathbf{s},\chi}$*, find* $\mathbf{s}$.

Problem 3.8 (The $Decision\text{-}LWE_{n,m,q,\mathcal{B},\chi}$ **problem)** *Let* $\mathbf{s} \in \mathbb{Z}_q^n$ *be chosen from some distribution* $\mathcal{B}$*. Without knowing* $\mathbf{s}$*, given* m *samples* $(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_m, b_m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*, distinguish between the following two cases: (i) the samples are drawn independently from the distribution* $A_{\mathbf{s},\chi}$*, or (ii) the samples are drawn independently from the uniform distribution on* $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

We further define some algebraically-structured SIS/LWE problems. Typically in these algebraically-structured variants, a ring R is taken to be a degree-n polynomial ring of the form $R = R_q = \mathbb{Z}_q[X]/(f(X))$, for some positive integer q. Broadly speaking, the choices of $f(X)$ considered in the third round take the form $f(X) = X^{2^d} + 1$ as in KYBER, Saber, Dilithium, and FALCON. Separately, $f(X) = X^n - 1$ and $f(X) = X^{n-1} + X^{n-2} + \ldots + X + 1$ are used by NTRU, and $f(X) = X^p - X - 1$ for a prime p is chosen by NTRU LPrime and sNTRU Prime. In the third round, the uses of algebraic-SIS/LWE mostly took on a module-based formulation as follows.

Problem 3.9 (The $Module\text{-}SIS_{R,m,k,q,\beta}$ **problem)** *Given* m *vectors of polynomials* $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_q^k$*, chosen uniformly at random, let us view them as the rows of a matrix* $\mathbf{A} \in R_q^{m \times k}$*. Then find a nonzero polynomial vector* $\mathbf{z} \in R_q^k$ *of norm* $||\mathbf{z}|| \leq \beta$ *such that* $\mathbf{Az} = \mathbf{0}$.

Problem 3.10 (The decisional $Module\text{-}LWE_{R,m,k,q,\mathcal{B},\chi}$ **problem)** *Let* $\mathbf{s} \in R_q^k$ *be chosen from some distribution* $\mathcal{B}$*. Without knowing* $\mathbf{s}$*, given* m *samples* $(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_m, b_m) \in R_q^k \times R_q$*, distinguish between the following two cases: (i) every sample is drawn independently from the distribution* $A_{R,\mathbf{s},\chi}$ *(the analogue of the LWE distribution* $A_{\mathbf{s},\chi}$*, but over* R_q*), or (ii) every sample is drawn independently from the uniform distribution on* $R_q^k \times R_q$.

Finally, we introduce the family of *Learning With Rounding* (LWR) problems. The difference between LWE and LWR is that the samples are formed as rounded inner products rather than independently sampling from an error distribution χ. That is, LWR samples take the form $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ where $b_i = \lfloor \langle \mathbf{s}, \mathbf{a}_i \rangle \rceil_p$, and $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ (for $p < q$) is the modular rounding function defined as $\lfloor x + q\mathbb{Z} \rceil_p := \lfloor x \cdot (p/q) \rceil + p\mathbb{Z}$.

In algebraic settings, replacing each instance of addition by $e \leftarrow \chi$ with an application of an analogous modular rounding function gives a natural way to extend LWR to any algebraically-structured LWE problem (e.g., defines the *Module-LWR* problem).

The known attacks against lattice-based cryptosystems can be organized into a few broad classes, including primal [163], dual [164–166], and hybrid [167, 168] attacks. In most cases, the cost of these attacks depends on the cost of finding sufficiently short vectors in some lattice. Depending on the context, this problem is known as the Shortest Vector Problem (SVP) or the Gap Shortest Vector Problem (gapSVP). SVP asks to find the shortest vector in some presented lattice, whereas gapSVP asks to estimate the length of the shortest vector in the presented lattice. Another version of these problems is the Shortest Independent Vector Problem (SIVP), which asks to find n-many shortest vectors that are linearly independent of each other (n is the dimension of the lattice).

Estimating the cost of solving these critical security problems on real-world lattice instances is highly non-trivial, as it involves selecting the best type of attack, and optimizing the parameters of the attack to find the best possible solution with a specified amount of computational resources. Theoretical bounds and computer simulations are both used, in order to estimate the cost of solving extremely large instances of these problems. This has been a focus of intense research in recent years, leading to credible estimates of the concrete security of lattice-based cryptosystems. See Appendix C for more discussion of the techniques used in these estimates.

3.3 Security Models and Definitions

3.3.1 IND-CPA, IND-CCA2, and EUF-CMA Security

In the original CFP [9], NIST gave security definitions, which were to be taken as statements of what NIST considered to be the relevant attack model. NIST planned on standardizing KEMs that would enable "semantically secure" encryption or key encapsulation for general use – in particular, a scheme that provides indistinguishability of ciphertexts under adaptive chosen ciphertext attack. Roughly speaking, a scheme is secure in this definition if no adversary can distinguish "challenge encryptions" of two messages of their choosing, despite having oracle access to both encryption and decryption (the latter not being usable on the challenge.) This property is denoted *IND-CCA2 security* in the academic literature [169]. Throughout this report, the terms *IND-CCA* or *CCA-security* will also be used to refer to this property.

Almost all of the KEM candidates submitted to NIST attained this feature by first specifying an IND-CPA public-key encryption scheme. An *IND-CPA* encryption scheme is one that provides indistinguishability of ciphertexts under chosen plaintext attack; this is the

same definition as above, except the adversary does not have oracle access to decryption. The full IND-CCA2 KEMs were then constructed by combining the IND-CPA encryption schemes with some type of Fujisaki-Okamoto (FO) transform [170–172].

For the signature schemes, the relevant security definition was existential unforgeability under adaptive chosen message attack. Roughly speaking, in this definition the adversary is granted oracle access to the signing function and must produce a valid signature for a message that has not previously been signed by the oracle. This property is denoted *EUF-CMA security* in the academic literature [169].

In addition to these security definitions, there are additional security properties that have been discussed in the literature (see, for example, [173, 174]). While not required for submission, such properties may be desirable.

3.3.2 Idealized Security Models

The Random Oracle Model (ROM). Proving security of cryptographic schemes that make use of hashing can be challenging, particularly in the "plain model" in which the adversary simply receives the full description of the hash function as input. For this reason, many of the schemes in the NIST PQC Standardization Process are instead supported by proofs in the idealized Random Oracle Model, or ROM [175]. In this model, a uniformly random function H is sampled at the beginning of time, and all parties are provided black-box access to H; any evaluations of the hash function in the real setting are then replaced with queries to H. Proving security of a cryptographic scheme in the ROM can be interpreted as indicating security against certain kinds of attacks (e.g., ones that do not exploit special structural properties of the hash function). While the ROM has certain shortcomings that are important to keep in mind (see, e.g., [176]), it has a successful history in both theoretical and applied cryptography [177].

The Quantum-accessible Random Oracle Model (QROM). A classical adversary who knows a circuit for some function f can certainly evaluate that function in black-box form (i.e., $x \mapsto f(x)$) by locally implementing the circuit for f. A quantum adversary who knows a circuit for f has the added ability to implement a certain unitary circuit associated to f, enabling queries in superposition (e.g., $\sum_x \alpha_x |x\rangle \mapsto \sum_x \alpha_x |x\rangle |f(x)\rangle$). This is a generic ability that does not require any specific properties of f.

The above observation motivated the definition of the Quantum-accessible Random Oracle Model, or QROM [178]. This model simply expands the ROM (as defined above) by allowing all parties with quantum computers black-box access to the unitary

$$U_H : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle \,.$$

The relevance of this model is justified by the existence of nontrivial quantum attacks that use such quantum queries but no specific properties of the hash function itself. A standard example is the use of Grover's algorithm [179] to find preimages with quadratically fewer queries than is possible in the classical-query model.

4. Summary of Third-Round Candidates

Each of the third-round candidates is discussed below, including summaries of their advantages and disadvantages. In addition, the discussion provides reasons why a scheme was (or was not) selected for either standardization or advancing to the fourth round.

The nine public-key encryption/key-encapsulation mechanisms are discussed first (in Sections 4.1 to 4.3), and the six digital signature schemes follow (in Sections 4.4 and 4.5). For both KEMs and signatures, the algorithms selected for standardization are presented first, followed by candidates selected for the fourth round and, finally, the algorithms not selected to continue on in the NIST PQC Standardization Process.

4.1 KEM Selected for Standardization

NIST has selected one KEM for standardization at this time. Four additional KEMs will continue to be evaluated, and NIST anticipates standardizing at least one of them at the conclusion of the fourth round.

4.1.1 CRYSTALS-Kyber

KYBER is a module learning with errors (MLWE)-based key encapsulation mechanism with its original design presented in [180]. As compared to similar schemes based on unstructured LWE, this design offers significant efficiency advantages.

Design. Like other LWE-style KEM candidates in the third round, KYBER is constructed first as an IND-CPA-secure PKE scheme, then boosted to an IND-CCA-secure KEM by a Fujisaki-Okomoto (FO) type of transform [170].

The base PKE scheme is derived from the MLWE problem. The ring is a cyclotomic power-of-2 ring, $R = \mathbb{Z}[X]/(X^{256}+1)$, and the module rank k is set to $k = 2, 3$, or 4 (corresponding to security categories 1, 3, 5). Other parameters include the integer modulus $q = 3329$, a distribution χ on "short" polynomials of R_q, and a public matrix of polynomials $\mathbf{A} \in R_q^{k \times k}$ pseudorandomly generated from a uniformly random 256-bit string. Two secret vectors of polynomials $\mathbf{s}, \mathbf{e} \in R_q^k$ are sampled independently from χ coefficient-wise. The vector $\mathbf{s}$ is regarded as the secret key, and the vector $\mathbf{e}$ is called the error term. This forms the MLWE public key $pk := (\mathbf{A}, \mathbf{b}) := (\mathbf{A}, \mathbf{As} + \mathbf{e})$.

Encryption and decryption instantiate the Lindner-Peikert paradigm [181]. To encrypt a message m (a 256-bit string), one samples two vectors of polynomials $\mathbf{r}, \mathbf{e}_1 \in R_q^k$ as well as a polynomial $e_2 \in R_q$, with all coefficients of each polynomial chosen independently from χ. Then, the ciphertext c is formed as

$$c := (c_1, c_2) := \left(\mathbf{rA} + \mathbf{e}_1, \mathbf{rb} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m\right) \in R_q^k \times R_q,$$

where $\left\lceil \frac{q}{2} \right\rfloor \cdot m$ should be interpreted in the natural way – as the vector of coefficients of a single polynomial in R_q (with padding as needed). In the actual KYBER PKE scheme,

some of the low-order bits of the ciphertexts are discarded; that is, the ciphertexts are "compressed" in a precise way.

To decrypt a ciphertext c using the secret key $\mathbf{s}$, after first "decompressing" the ciphertext, one computes the intermediate value $v = c_2 - \mathbf{c}_1\mathbf{s}$ then rounds each coefficient of the polynomial v modulo 2 to extract the transmitted bit-string m.

Security. KYBER inherits a strong theoretical security foundation from decades of lattice cryptography literature. Moreover, a series of results over the past decade support the notion that the Module version of LWE is suitable for high-performance cryptosystems without sacrificing security. In particular, a 2012 work by Langlois and Stehlé [182] provides a relatively tight reduction from worst-case Module-SIVP to average-case Module-LWE. Additional results have given evidence that, roughly speaking, transitioning from rank one (i.e., Ring-LWE) to constant rank (i.e., Module-LWE) is likely to increase performance and unlikely to sacrifice security [183–185].

Beyond discussion of lattice cryptographic theory, it was mentioned above that KYBER employs a particular variant of the FO transform to achieve CCA security. The security proofs hold tightly in the ROM [171, 172] and non-tightly in the QROM. Yet under various other natural assumptions, KYBER may also achieve a tight security reduction in the QROM [186].

In the third round, the KYBER team also provided an extensive analysis of the system's concrete security ([14, Sections 5.2 and 5.3]). While many of the details in this section remain somewhat speculative, the overall conclusions appear consistent with the state of the art in lattice cryptanalysis. In addition to a best guess concrete security estimate for all the parameter sets, that section contains a list of open questions in lattice cryptanalysis, and gives a range of estimates for the possible effect on concrete security corresponding to each open question. According to the analysis provided in the KYBER specification, in the very worst case, if every open question is resolved in the worst case for KYBER, some of the parameter sets may fall below their targeted security level in the gate-count model, although even in this case, it is likely the submitted parameter sets will still meet their targeted security levels in any cost model which realistically models the cost of memory access.

Performance. Like the other structured lattice KEMs, KYBER's public key and ciphertext sizes are on the order of a thousand bytes, which should be acceptable for most applications (see Table 6). In comparison, KYBER's bandwidth is smaller than NTRU but about 10% larger than Saber.

KYBER has fast key generation, encapsulation and decapsulation in software [33] (see Section 2.2.2). There have been several works on optimizing implementations of KYBER in both software and hardware, as well as in hybrid hardware/software settings [35, 40–45, 80]. For high-speed FPGA implementations, [46] shows that in terms of speed and resource realization, KYBER is a leading performer for all operations: key generation, encapsulation and decapsulation (among the finalist lattice KEMs).

Overall, the performance data reported from these referred works indicate that KYBER

has sufficient performance in many different environments.

Significant events since Round 2. At the beginning of the third round, the KYBER team increased the binomial noise parameter η from 2 to 3 for the centered binomial distribution used to sample public-key components in its category 1 parameter set. This was partly due to a suggestion from the NIST PQC team. Mildly increasing the noise resulted in a stronger defense against lattice reduction attacks without raising the decryption failure rate above the requisite threshold for security.

To compensate for the increase in decryption failure probability, the number of dropped bits for each coefficient of the second component of the ciphertext was changed from 4 to 3 for the category 1 parameters (KYBER512). In addition, during key generation the uniform sampling was made more efficient by using rejection sampling on 12-bit integers instead of 2-byte integers.

To provide a more precise core SVP estimate of KYBER512, the KYBER team accounted for the noise added from the rounding operation of the ciphertext. Assuming a weak version of LWR, the added noise yields 6 more bits of core SVP hardness for KYBER512.

During the third round, some improvements to the dual attack were proposed [165, 166], leading to lower estimated security in the RAM model than was claimed in the KYBER specification. These results suggest that all three KYBER parameter sets fall slightly below the security targets for their claimed security levels when the cost of memory access for the attacker is not explicitly taken into account.

Overall assessment. The security of KYBER has been thoroughly analyzed and is based on a strong framework of results in lattice-based cryptography. KYBER has excellent performance overall in software, hardware and many hybrid settings.

While the three structured lattice finalists are all strong candidates, NIST has selected KYBER for standardization. A significant factor in the decision to choose KYBER over NTRU was NTRU's performance (particularly key generation), which was not quite as efficient as that of KYBER. There is arguably more evidence to support the MLWE problem (which KYBER is based upon) than the MLWR or NTRU assumptions which Saber and NTRU respectively rely upon.

4.2 KEMs Advancing to the 4th Round

4.2.1 BIKE

BIKE (Bit Flipping Key Encapsulation) is a KEM based on binary linear quasi-cyclic moderate density parity check (QC-MDPC) codes [187]. The BIKE cryptosystem was initially designed for ephemeral key use but has now been claimed to also support static key use.

Design. The binary linear QC-MDPC code $C(n,k)$ used in BIKE is constructed as follows. The secret key is a parity check matrix $H_{r\times 2r}$ for a quasi-cyclic moderate density parity check code, composed of two circulant blocks, where r is prime and chosen so that x^{r-1} has

only two irreducible factors modulo 2. Each row of H has Hamming weight $w \approx \sqrt{n}$, where $w \equiv 2 \bmod 4$. All matrix operations in BIKE can be viewed as polynomial operations due to the isomorphism between the ring of $v \times v$ circulant matrices and the polynomial ring $\mathbb{F}_2[x]/(x^v+1)$, for any $v \in \mathbb{N}$. The secret key may then be thought of as a 1×2 module, (h_0, h_1). The public key $H_{\text{pub}} = (1, h_0^{-1}h_1)$ is the secret key in systematic form, which is computed by multiplying H by h_0^{-1}.

The underlying BIKE PKE follows Neiderreitter-style encryption. At a high level, a message is encoded as an error vector e of weight t and the corresponding ciphertext is computed as $H_{\text{pub}}e^T$. Decryption is accomplished by multiplying the ciphertext by h_0 to produce the syndrome He^T and then using the recommended Black-Grey-Flip bit-flipping decoder [188] to recover e.

Security. The proof of IND-CPA security of the underlying PKE in the ROM depends on the difficulty of solving the decisional Quasi-cyclic Syndrome Decoding (QCSD) and the decisional Quasi-cyclic Codeword Finding (QCCF) problems. These problems are as defined below. Let $\mathcal{R} = \mathbb{F}_2[x]/(x^r-1)$.

Problem 4.1 (Decisional QCSD) *Given $h \in \mathcal{R}$, a vector $y \in \mathcal{R}$, and target $t > 0$, determine whether there exists $(e_0, e_1) \in \mathcal{R}^2$ such that $|e_0| + |e_1| = t$ and $e_0 + e_1 h = y$.*

Problem 4.2 (Decisional QCCF) *Given $h \in \mathcal{R}$ and target $v > 0$, determine whether there exists $(c_0, c_1) \in \mathcal{R}^2$ such that $|c_0| + |c_1| = v$ and $c_0 + c_1 h = \boldsymbol{0}$.*

To avoid trivial distinguishers, the parity of $|h|$ is restricted to be odd, the parity of $|y|$ is restricted to equal the parity of t. [189]. The best known algorithms for solving these problems are information set decoding (ISD) and its variants, as described in Section 3.2.1.

To achieve λ bits of security against an IND-CPA attacker, the cost of breaking both problems 4.1 and 4.2 must exceed 2^λ. The work factor for solving linear decoding problems using ISD was shown to be asymptotically equivalent across all variants of ISD [127] and was used to derive the following approximation:

$$\lambda \approx t - \frac{1}{2}\log_2 r \approx w - \log_2 r, \tag{1}$$

where t, r, w are as described above. The BIKE parameters for each security level were selected according to (1).

The FO$^{\not\perp}$ transform, as described in [172], is applied to the CPA-secure PKE to achieve a claimed IND-CCA KEM. The PKE must be δ-correct,[11] for $\delta \leq 2^{-\lambda}$, to apply this transformation. The maximum decryption failure rate over all messages is difficult to compute in BIKE's case as certain messages (near codewords, etc.) are known to cause more decoding failures than others. To avoid this issue, BIKE updated the specification to randomize

[11] A KEM is δ-correct if the decapsulation fails (i.e., disagrees with encapsulation) with probability at most δ on average over all keys and messages. Similarly, a decoder will be δ-correct if its failure rate is at most δ on average when the input is drawn uniformly.

the message [17]. The decryption failure rate must also be sufficiently low in the static-key scenario to prevent the key recovery attack in [190].

Performance. The quasi-cyclic structure of BIKE enables public key and ciphertext sizes comparable to – though slightly larger – than the structured lattice KEMs. In comparison to HQC, BIKE has smaller bandwidth. See Tables 6 and 7.

Figure 9 shows, we see that BIKE is one of the more efficient alternate KEM candidates. This is especially true when considering the overall performance measures in Figure 10, as the smaller bandwidth of BIKE is significant. It can be noted that BIKE's key generation algorithm runs significantly slower than the other structured code- and lattice-based schemes. In addition, the computation of $10r$ inner products during the decoding procedure results in a decapsulation that runs 6 to 9 times slower than that of HQC. Several hardware benchmarks also confirm that performance of BIKE would be suitable for most applications [191–194].

Significant events since round 2. At the beginning of the third round, the BIKE team narrowed down the included variants to just one and updated the recommended decoder to the Black-Grey-Flip [188]. Security category 5 parameters were added, at NIST's encouragement. BIKE no longer uses the Parallel-Hash algorithm; all random oracles are now implemented as SHA-3-based constructions to improve hardware performance and to avoid any IP issues.

The BIKE specification now claims IND-CCA security, citing additional analysis to support their claim [195, 196]. Iterative, bit-flipping decoders are not characterized by a bounded decoding radius; thus, there is an expected nonzero probability of decoding failure. Vasseur's work on the classification of BIKE weak keys and classes of near codewords expected to disrupt decoding does not disprove IND-CCA security of BIKE [195, 196]. However, these classes are not known to be exhaustive and an upper bound on the decoding failure rate has yet to be found.

Overall assessment. BIKE has the most competitive performance among the non-lattice-based KEMs. The recent, explicit claim of IND-CCA security by the BIKE team is encouraging. NIST anticipates that additional time in the fourth round will allow more vetting by the community of BIKE's security claims.

NIST intends to select at least one additional KEM for standardization at the end of the fourth round. BIKE remains under consideration due to its overall performance and substantially different security assumption from the currently selected KEM.

4.2.2 Classic McEliece

Design. Classic McEliece is a code-based KEM that uses a binary Goppa code in the Niederreiter variant of the McEliece cryptosystem combined with standard techniques to achieve CCA security. Due to the use of Goppa codes, the KEM has perfect correctness.[12]

[12] A perfectly correct KEM or PKE is one for which every ciphertext generated using the encapsulation/encryption function may be correctly decrypted using the decapsulation/decryption function. In con-

It is a merger of the second-round submissions Classic McEliece and NTS-KEM. The original McEliece cryptosystem was published in [197] and was also based on a binary Goppa code.

Security. The Classic McEliece submission cites [198] and other results as giving a tight proof of the submitted KEM's IND-CCA2 security in the quantum random oracle model, based on the assumption that the 1978 McEliece scheme provides one-way under chosen-plaintext attacks (OW-CPA) security. Confidence in the security of the 1978 scheme is mostly established based on the scheme's long history of surviving cryptanalysis with only minor changes in the complexity of the best-known attack. Alternatively, the security of the scheme could be established under the assumptions that row-reduced parity check matrices for the binary Goppa codes used by Classic McEliece are indistinguishable from row-reduced parity check matrices for random linear codes of the same dimensions and that the syndrome decoding problem is hard for random linear codes with those dimensions. The state of the art in cryptanalysis does not contradict these assumptions, although binary Goppa codes with very different dimensions from those used by the Classic McEliece submission have been shown to be distinguishable from random codes [199].

A number of approaches to the cryptanalysis of Classic McEliece have been studied. The most effective known attacks, and those used to set the parameters of Classic McEliece, are information set decoding attacks, as described in Section 3.2.1. Key recovery attacks have also been studied. These either attempt to find the private key by algebraic techniques or brute force search. While algebraic techniques have been used to break variants of McEliece based on other algebraic codes [200–204] or based on Goppa codes with additional structure imposed [205], these techniques appear to be significantly more costly than information set decoding for attacking Classic McEliece.

Performance. Classic McEliece has a very large public key size and fairly slow key generation. This is likely to make Classic McEliece undesirable in many common settings. However, in settings where a public key is reused many times and does not be need to be retransmitted for each new communication, it is possible that the performance profile of Classic McEliece could have some advantages. In particular, Classic McEliece has the smallest ciphertext sizes of any of the NIST PQC candidates.

Significant events since Round 2. While there has been no significant cryptanalysis on Classic McEliece, it did spark a large amount of discussion on the pqc-forum. Much of this discussion concerned issues that are generally applicable to code-based schemes or even KEMs in general. However, a few issues specific to the Classic McEliece submission were uncovered. In particular, based on the concrete analyses of [135], at least one of the parameter sets (targeting category 3) appears to fall slightly short of its target security level (probably meeting category 2 instead). The submission document also contains a potentially misleading implementation note that NIST recommends be removed. A misuse scenario was also brought up, where reusing the same error vector when encapsulating for

trast, some KEMs and PKEs have a very small decryption failure rate.

multiple public keys can result in a significant security loss. This scenario should not happen assuming the random number generator is functioning properly, but it could be made even less likely through fairly simple countermeasures like incorporating the public key in the derivation of the error vector. A similar misuse scenario with similar countermeasures also applies to BIKE, HQC, and NTRU.

Overall assessment. NIST is confident in the security of Classic McEliece and would be comfortable standardizing the submitted parameter sets (under a different claimed security strength in some cases). However, it is unclear whether Classic McEliece represents the best option for enough applications to justify standardizing it at this time. For general-purpose systems wishing to base their security on codes rather than lattices, BIKE or HQC may represent a more attractive option. For applications that need a very small ciphertext, SIKE may turn out to be more attractive. NIST will, therefore, consider Classic McEliece in the fourth round along with BIKE, HQC, and SIKE. NIST would like feedback on specific use cases for which Classic McEliece would be a good solution.

4.2.3 HQC

HQC (Hamming Quasi-Cyclic) is a KEM based on QC-MDPC codes, where no trapdoor is hidden in the code [206]. The motivation for the HQC framework was to generate a code-based scheme that could benefit from a quasi-cyclic structure but have a more direct security reduction to the problem of decoding a random linear code. In particular, the submitters contend that it is difficult to reduce the security of a code-based scheme to a general decoding problem (like Problems 3.1 or 3.2) when the public key masks the secret key by scrambling or permutation operations. [206, 207].

Design. HQC is based on QC-MDPC codes and follows an LWE-like encryption protocol. The IND-CPA secure PKE can be described as follows. Let $\mathcal{R} = \mathbb{F}_2[x]/(x^n - 1)$ for n prime such that x^{n-1} has only two irreducible factors modulo 2. The secret key is a randomly sampled pair $(x, y) \in \mathcal{R}^2$, and the public key is the pair $(h, s = x + h \cdot y)$ where h is randomly sampled from $\mathcal{R}$ and used to construct the generator matrix $G \in \mathbb{F}_2^{k \times n}$ of the code. Because the secret key is generated independently of the code, there is no hidden structure in the HQC public parity-check matrix. This enables the security reduction to be independent of the decoding algorithm used for decryption [206].

To encrypt a message $m \in \mathbb{F}_2^k$, the sender randomly samples three polynomials $e, r_1, r_2 \in \mathcal{R}$ of appropriate weights and responds with the ciphertext

$$c = (u, v) := (r_1 + h \cdot r_2, mG + s \cdot r_2 + e). \tag{2}$$

To decrypt, the receiver uses the decoding algorithm to decode $(v - u \cdot y)$. The HQC decoder is a concatenation of Reed-Solomon Reed-Muller codes (RMRS).

Security. The IND-CPA security of HQC relies on the difficulty of the QCSD with parity problem, a close variant of Problem 4.1. The FO$^{\not\perp}$ transform [172] is applied to the CPA-secure PKE to achieve an IND-CCA KEM.

The decoder used in HQC has a well-defined minimum distance d and, consequently, a determinable error-correction capability $\delta = \lfloor \frac{d-1}{2} \rfloor$. The probability that an HQC ciphertext includes error e such that $|e| > \delta$ is captured in a closed-form analysis and used to produce an upper bound on the decryption failure rate. The provably and sufficiently low decryption failure rate is required for proper application of the $FO^{\not\perp}$ transform [172] and to resist key recovery attacks [190].

As with the other code-based schemes, the best known attacks are based on information set decoding; see Section 3.2.1.

Performance. The quasi-cyclic structure of HQC enables small public key and ciphertext sizes, although they are noticeably larger than the structured lattice KEMs. HQC ciphertexts and public keys are roughly 2.9 and 1.5 times the size of BIKE ciphertexts and public keys, respectively. See Tables 6 and 7.

Although the bandwidth of HQC exceeds that of BIKE, HQC's key generation and decapsulation only require a fraction of the kilocycles required by BIKE. When factoring in the bandwidth with performance numbers, HQC is one of the top two alternate KEMs advancing for overall performance in software (see Figures 9 and 10).

The HQC submission included some benchmarks for a hardware implementation, but there do not seem to be any other implementations in the literature.

Significant events since Round 2. The Round 2 submission included three parameter sets for security category 5: HQC-256-1, HQC-256-2, and HQC-256-3, each targeting different decryption failure rates. The parameter set HQC-256-1 was broken during the second round [208]. The updated HQC specification now contains only one parameter set for each security category, and each has a sufficiently low decryption failure rate to avoid the attack [208].

Side-channel attacks were found against HQC [209, 210], but the current implementations of HQC are said to run in constant time and avoid secret-dependent memory access.

Another significant change to the HQC specification (after the second round) was the removal of the BCH-repetition decoder due to overall improvements offered by the RMRS decoder [207].

Overall assessment. HQC offers strong security assurances and a mature decryption failure rate analysis. Although the quasi-cyclic structure of HQC enables reasonable sizes for public keys and ciphertexts, HQC public keys and ciphertexts are larger than all of the other remaining structured code- and structured lattice-based KEMs (see Tables 6 and 7). The overall performance of HQC is acceptable, though not optimal.

NIST intends to select at least one additional KEM for standardization at the end of the fourth round. HQC remains under consideration due to the rigorous security analysis and substantially different security assumption from the currently selected KEM.

4.2.4 SIKE

SIKE (Supersingular Isogeny Key Encapsulation) is a specific realization of the SIDH (Supersingular Isogeny Diffie-Hellman) protocol first proposed by de Feo, Jao and Plût [211, 212]. SIDH is a Diffie-Hellman-like key exchange protocol whose security is based on the hardness of finding isogenies between supersingular elliptic curves. SIKE is a key exchange mechanism with security against chosen-ciphertext attacks that is built around an optimized implementation of SIDH.

The motivation for designing post-quantum cryptosystems based on isogenies is as follows. In some sense, the isogeny-finding problem can be viewed as a loose analogue of the discrete log problem but using a large graph (the isogeny graph) rather than an abelian group. However, while there is a polynomial-time quantum algorithm for computing discrete logs over elliptic curves, the currently known quantum algorithms for finding isogenies are much slower: they take subexponential time over ordinary elliptic curves [213] and exponential time over supersingular elliptic curves.

Design. There are two main challenges in the design of the SIDH protocol. The first is how to describe and compute isogenies efficiently. The second is how to make Alice and Bob's operations "commute" so that one can construct a Diffie-Hellman-like protocol where the same shared key can be computed by applying Alice's operations followed by Bob's operations or vice versa.

In the SIDH protocol, isogenies are described by specifying their kernels, and only isogenies whose kernels can be generated by single points that are of "smooth order" are used (that is, the order of the point is a number whose prime factors are all small). The second issue is addressed by having Alice and Bob use different torsion groups $E[\ell]$ and $E[\ell']$, where ℓ and ℓ' are relatively prime, and having Alice and Bob exchange some additional torsion point information over a public channel (roughly speaking, Alice reveals the action of her isogeny on Bob's torsion group and vice versa). See [211, 212] for more details.

SIKE consists of an optimized implementation of the SIDH protocol combined with a modified transformation of [172] (an extension of the FO transform). The optimized implementation reduces the amount of communication and computation needed to run the protocol and also protects against side-channel attacks. The transformation is needed to provide security against chosen-ciphertext attacks.

Security. In essence, the security of SIKE follows from the hardness of finding isogenies between supersingular elliptic curves. This problem can be solved using a meet-in-the-middle algorithm or by using quantum algorithms for claw-finding and collision-finding. The cost of running these algorithms is fairly well-studied [214]. However, there is a technical question about how to measure the cost of using large amounts of memory in these attacks. Previous estimates assumed that an attacker could use – at most – 2^{96} bits of memory, which is unreasonably low for a hypothetical adversary capable of threatening security categories 3 or 5 [7]. While this error in analysis is unlikely to lead to a practical break, the parameters currently claimed by SIKE to meet categories 3 and 5 should most likely be considered to fall short of their security targets, meeting instead categories 2 and

4 respectively.

In addition, while there is a subexponential-time quantum algorithm for finding isogenies between ordinary elliptic curves [213], there are some obstacles to applying this algorithm in the supersingular case because the endomorphism ring of a supersingular elliptic curve is non-commutative. Finally, there has been recent progress in understanding how isogeny-finding is related to other computational problems involving endomorphism rings of supersingular elliptic curves [215].

However, the above picture becomes more complicated when one considers attacks that make use of the torsion point information that is revealed by the SIDH and SIKE protocols. Some progress in these torsion-point attacks have weakened the security of some variants of the SIDH protocol, although there has been no impact on SIKE itself [216]. There are also some plausible countermeasures to these torsion-point attacks [217]. There is some recent evidence that one can exploit the torsion point information revealed by the SIDH protocol to get a subexponential-time quantum attack on certain overstretched parameterizations of SIDH (bypassing the obstacle mentioned earlier, that is, the non-commutative structure of the endomorphism ring) [218]. There is no direct impact of this work on SIKE.

Finally, there has been a good amount of research on side-channel attacks and countermeasures for SIKE [219–221]. Certain countermeasures for SIKE were already known from previous work on implementing elliptic-curve cryptography [222, 223].

Performance. SIKE has relatively low communication costs on the order of hundreds of bytes (see Table 7). However, SIKE requires both parties to perform computations that are relatively expensive. To improve performance, one can use specialized algorithms for performing calculations with elliptic curves, and one can implement certain critical operations (such as finite field arithmetic) in x64 assembly code. Using such an implementation, SIKE encapsulation and decapsulation take on the order of tens of millions of cycles, which is still relatively slow compared to other post-quantum schemes (see Figure 9).

SIKE's performance on embedded devices may be an issue because the time to perform a single key encapsulation/decapsulation (on a low-end 32-bit ARM processor, for instance) can be noticeable. Implementing SIKE in FPGAs may be a good route to achieving better performance in embedded devices [224, 225]. In addition, it may be attractive to construct hybrid protocols that use SIKE together with pre-quantum-secure ECDH (elliptic curve Diffie-Hellman key exchange), since SIKE and ECDH can share some common subroutines.

Significant events since Round 2. There has been additional progress in developing faster implementations of SIKE on small ARM processors and FPGAs [226], as well as more refined analyses of the concrete security of SIKE, using budget-based models to estimate the cost of using large amounts of memory for cryptanalysis [227]. In addition, the SIKE team has announced some public challenges with cash prizes to encourage practical cryptanalysis of SIKE [217].

Overall assessment. SIKE is an unusual candidate, as it relies on a different hard problem than all of the other post-quantum cryptosystems being evaluated by NIST. In terms of

performance, it has both advantages (small key sizes) and disadvantages (slow running times). SIKE seems promising but needs further study, as it is still a relatively new scheme.

4.3 KEMs no longer being considered

4.3.1 FrodoKEM

FrodoKEM is an LWE-based key encapsulation mechanism. Unlike the other LWE-based candidate KEMs, it relies only on the hardness of the "plain" or "unstructured" variant of LWE. While this offers a potential security advantage, it also comes with a significant cost in performance.

Design. The decisional LWE problem (see subsection 3.2.3) naturally leads to a public-key encryption scheme: the secret vector $\mathbf{s}$ is the secret key, and a collection of LWE samples $[\mathbf{A} \mid \mathbf{As}+\mathbf{e}]$ (organized as a matrix) is the public key. To encrypt a bit b, one sums a random subset of the samples and then adds $(0,0,\ldots,0,b\cdot q/2)$. Here $q \leq 2^{16}$ is the integer modulus and is selected to be a power of 2. Distinguishing ciphertexts then amounts to distinguishing "nearly true" from "far from true" equations mod q in the unknown variables $\mathbf{s}$, a problem which is as hard as the decisional LWE problem.

FrodoPKE is an IND-CPA-secure PKE that relies on an optimized version of the above concept due to Lindner and Peikert [181]. The private key is now a matrix $\mathbf{S}$ and the public key is $(\mathbf{A},\mathbf{B} := \mathbf{AS}+\mathbf{E})$ with the entries of $\mathbf{S}$ and $\mathbf{E}$ sampled from a discrete Gaussian distribution χ on $\mathbb{Z}_q$. To encrypt a message encoded into a matrix $\mathbf{M}$ over $\mathbb{Z}_q$, the sender chooses random Gaussian matrices $\mathbf{S}',\mathbf{E}',\mathbf{E}''$ and sends the ciphertext

$$(\mathbf{C}_1,\mathbf{C}_2) := (\mathbf{S}'\mathbf{A}+\mathbf{E}',\mathbf{S}'\mathbf{B}+\mathbf{E}''+\mathbf{M}) \tag{3}$$

To decrypt, the receiver computes $\mathbf{C}_2-\mathbf{C}_1\mathbf{S} \approx \mathbf{M}$. Provided that the receiver's original encoding is robust to noise in the lower order bits of $\mathbf{M}$, the sender can then recover the receiver's message. Note that, in the FrodoPKE implementation, the matrix $\mathbf{A}$ above is pseudorandomly generated using AES-128 or SHAKE128.

From FrodoPKE, the authors apply a certain Fujisaki-Okamoto (FO) transform [171] to obtain FrodoKEM, an IND-CCA secure key encapsulation mechanism. The specific FO transform is (a slightly adapted version of) the "implicit rejection" transform from [172].

Security. The cryptanalysis history relevant to Frodo is largely positive. Despite some marginal progress, both the LWE problem and KEMs in the style above seem resistant to classical and quantum attacks. Security is also supported by theoretical asymptotic proofs: a series of reductions show that breaking Frodo (for large parameter choices) would imply a fast algorithm for certain worst-case lattice problems (e.g., bounded distance decoding) that are believed to be hard [228]. As is typical, these theorems do not hold for the concrete parameter choices used in Frodo. However, they do indicate some fundamental soundness in the core idea underlying the Frodo approach.

A notable strength of Frodo is that the random matrix $\mathbf{A}$ is completely unstructured, and as a consequence, the security of FrodoKEM depends on the plain LWE problem rather than

on its structured variants (Module-LWE or Ring-LWE). This means that FrodoKEM could remain secure even in a future world where structured lattices are broken.

Performance. Unfortunately, the conservative security choices of FrodoKEM also make it the lattice scheme with the worst performance overall. Roughly speaking, the structural LWE assumption on the matrix $\mathbf{A}$ made by other lattice schemes results in a quadratic savings. As a result, Frodo is clearly not an immediate drop-in general-purpose scheme. For example, its best-performing parameter set would mean a public key + ciphertext package of roughly 20 000 bytes (see Table 7).

Significant events since Round 2. Around the start of the third round, an attack was found on the implementation of Frodo, which turned out not to be constant time [229]. This issue has since been fixed by the Frodo team.

Overall assessment. In terms of security, Frodo's conservative design choices are laudable. At the same time, these choices mean that Frodo's performance is significantly worse than schemes based on structured lattices. While NIST does intend to select at least one additional KEM for standardization at the end of the fourth round, three KEMs (BIKE, HQC and SIKE) are better placed than Frodo for this role. They have generally better performance, and they are based on substantially different assumptions from the KEM being standardized at present. Therefore, NIST did not select FrodoKEM to continue into the fourth round.

4.3.2 NTRU

The NTRU encryption scheme was first presented in 1996 [146, 230]. It was among the first publicly known lattice-based encryption schemes. While there have been a few versions of NTRU considered over the years, the central design features have remained consistent and are present in the NTRU submission. NTRU is based on a different computational hardness assumption from LWE- or LWR-based cryptosystems like KYBER and Saber.

Design. The third-round finalist NTRU is a merger of two earlier submissions: NTRU-HRSS-KEM [231] and NTRUEncrypt [232]. NTRU includes parameter sets based on each of the earlier submissions, which are denoted NTRU-HPS and NTRU-HRSS. All parameters of the merged submissions are perfectly correct (i.e., they have a decryption failure of 0 for honestly generated ciphertexts).

Informally, the basic version of NTRU encryption is implemented using polynomials from the ring $\mathcal{R} = \mathbb{Z}_q[x]/(x^n - 1)$, where q is a power of two. Two polynomials f and g are generated with coefficients in the set $\{-1, 0, 1\}$, and $h = g \cdot f^{-1}$ in $\mathcal{R}$. The public key is h, while the polynomials f and g are private. To encrypt a uniformly random message m represented by a polynomial in $\mathcal{R}$ with $\{-1, 0, 1\}$-coefficients, the sender computes $c = 3hr + m$, where $r \in \mathcal{R}$ is a polynomial with coefficients chosen uniformly at random from the set $\{-1, 0, 1\}$. To decrypt, the private key holder calculates $e = cf \bmod q$ and then recovers the message m from $e \cdot f^{-1} \bmod 3$.

The NTRU version called NTRU-HPS uses fixed weight sample spaces for generating polynomials. Here, fixed weight means that when looking at the coefficients, which are all drawn from $\{-1,0,1\}$, the number of the total 1s and -1s is a fixed value. In comparison, the NTRU-HRSS version uses arbitrary random weight sample spaces, meaning that each coefficient is chosen uniformly at random from the set $\{-1,0,1\}$.

As specified, the NTRU PKEs are not IND-CCA secure. Like the other KEMs in the NIST PQC Standardization Process, a version of the Fujisaka-Okamoto transform is used to convert the PKEs into IND-CCA2 secure KEMs. Specifically, NTRU uses the SXY transform [186], which basically re-encrypts to check the output from decryption and to output a random value when the check fails. As a consequence, the attacker would not get any noticeable information from seeing the output when an ineligible ciphertext is input into the decapsulation function.

Security. In addition to RLWE, the security of NTRU is also based on the NTRU assumption described in Section 3.2.3. The NTRU KEMs have tight CCA-security reductions to the underlying PKEs in the ROM, and non-tight security reductions in the QROM. Making some additional non-standard assumptions, one of the QROM security proofs can be made tight. The CCA security proofs are obtained from the OW-CPA assumption for the PKEs, thus relating the security of the NTRU submission to the original 1996 NTRU design.

The submission specification uses both local and non-local cost models for determining the security category of their parameter sets. For a more direct comparison with the other KEM finalists, the assignment of security categories according to the non-local cost model is appropriate. This is what NIST used for NTRU in the figures and tables in this report.

The design and parameter choices of NTRU protect against all the attacks known today. In their specification, analysis is provided for the primal and dual lattice attacks. The specification analyzes quantum versions of the above attacks as well but notes that all existing claims of a quantum speedup for lattice reduction algorithms rely on the Quantum-RAM model of computation, which the submission describes as sufficiently unrealistic to be irrelevant to the security of NTRU in practice.

The NTRU problem was first posed in 1996 and it remains unbroken despite many research advances in lattice attacks over the past few decades. This long security analysis provides confidence in the security of NTRU.

Performance. The public key and ciphertext sizes for NTRU are comparable to the other structured lattice KEM candidates, although about 25% larger (see Table 6). NTRU KEMs have very good performance in software, especially on an AVX2 machine (see Figure 1). NTRU key generation is noticeably slower than that of the other two lattice KEM finalists.

Significant events since Round 2. After being selected to be a finalist in the third round, NTRU made a few minor changes. This included an updated security analysis, as well as some patches to some reported bugs. In addition, the NTRU team created a large number of potential parameter sets to illustrate the flexibility of being able to easily make security/performance trade-offs. Later on during the third round, NTRU officially provided parameter sets for the security category 5 level after a request from NIST [10].

Overall assessment. One important feature of NTRU is that because it has been around for longer, its IP situation is more clearly understood. The original designers put their patents into the public domain [113], in addition to most of them having expired.

As noted by the submitters, NTRU may not be the fastest or smallest among the lattice KEM finalists, and for most applications and use cases, the performance would not be a problem. Nonetheless, as NIST has selected KYBER for standardization, NTRU will therefore not be considered for standardization in the fourth round.

4.3.3 NTRU Prime

The NTRU Prime submission [60], which consists of two structured-lattice-based cryptosystems, was first proposed in [233] as an exploration of the design space of "NTRU-like" cryptosystems, with the goal of reducing the attack surface with only minor loss of efficiency.

Design. NTRU Prime has several unusual design features. It has two variants: Streamlined NTRU Prime, which is modeled after the original NTRU, and NTRU LPRime, which combines some aspects of NTRU with some aspects of Ring-LWE cryptosystems (in the style of Lyubashevsky-Peikert-Regev [152]). In addition, NTRU Prime is constructed over a different ring: the "NTRU Prime ring," $\mathbb{Z}_q[x]/(x^p - x - 1)$. Finally, certain key parts of NTRU Prime are designed to operate deterministically (e.g., using rounding rather than random noise and eliminating the possibility of random decryption failures). The submitters have argued that these features improve the security of the scheme.

Security. The current version of NTRU Prime has performance and concrete security estimates (e.g., quantitative estimates of the computational resources required for usage and cryptanalysis) that are roughly comparable to other lattice-based cryptosystems.[13] As a result, the current version of NTRU Prime is notable more for its unusual design features, and claims that it offers higher security in a qualitative sense.

In order to state these claims, the designers of NTRU Prime have advocated for a specific approach to security analysis, based on a taxonomy of security risks [15]. This taxonomy is used to justify various design decisions, such as using rounding rather than random noise, and eliminating the possibility of decryption failures. However, some care is needed when reading this taxonomy, as it is a matter of subjective judgement which risks are the most serious and what is the best way of mitigating those risks.

One particular issue is the choice of the NTRU Prime ring (rather than a cyclotomic ring), which is claimed to eliminate the possibility of certain kinds of algebraic attacks. To date, most work on the cryptanalysis of algebraically structured lattices (see Appendix C) has focused on cyclotomic rings, because they are widely used and simpler to analyze. Relatively little is known about the security of cryptographic schemes that use the NTRU Prime ring.

[13] For example, two typical NTRU Prime parameter sets, sntrup761 and ntrulpr761, are roughly comparable to KYBER768, although there are some differences.

Another topic of interest is acquiring accurate estimates of the cost of running lattice basis reduction algorithms, which are used for cryptanalysis. The NTRU Prime team has used a variety of different methods to estimate these costs, leading to different estimates of the concrete security strength of NTRU Prime.

Performance. Streamlined NTRU Prime's performance profile is fairly similar to that of NTRU. In particular, Streamlined NTRU Prime's key generation is relatively slow. NTRU LPRime's key generation is much faster, resulting in a performance profile more similar to that of KYBER and Saber. The choice of the NTRU Prime ring prevents the use of certain fast algorithms for polynomial multiplication. However, the use of a ring whose degree is not a power of 2 allows for more flexibility in tuning the parameters of the cryptosystem to reach the desired security levels.

NTRU Prime's public keys and ciphertexts are each on the order of 1000-2000 bytes (see Table 7). On an Intel x86-64 processor, depending on the desired security level, encryption takes on the order of 50-100 thousand cycles, decryption takes on the order of 50-150 thousand cycles, and key generation takes on the order of 500-2500 thousand cycles (for Streamlined NTRU Prime) and 50-100 thousand cycles (for NTRU LPRime). Faster performance can be obtained by generating many keys simultaneously in batches and implementing the scheme in an FPGA [234, 235].

Significant events since Round 2. Recent work on NTRU Prime has focused on adding new parameter sets for NIST security categories 4 and 5, detailed security analysis of other lattice-based cryptosystems [15], demonstrating a post-quantum TLS protocol (integrating NTRU Prime with the OpenSSL software library) [234], and faster FPGA hardware implementations [235].

Overall assessment. The case for NTRU Prime relies substantially on the claim that its unusual choice of ring provides a security benefit over the algebraic structures used by the other lattice candidates, i.e., the claim that (1) there is likely to be an attack that significantly diminishes the security of NTRU, KYBER, and Saber, and (2) no similar attack is likely to affect NTRU Prime. At the end of the third round, the evidence for these two points is not particularly convincing. No algebraic attack has been published that directly impacts the concrete or asymptotic security of any of the third-round structured lattice candidates.[14] From a practical perspective, it seems likely that an unexpected breakthrough in cryptanalysis of *any* structured lattice scheme would reduce the community's confidence in *all* such schemes, including NTRU Prime.

For these reasons, NIST is not moving NTRU Prime to the fourth round of the evaluation process. In order to hedge against the possibility of a security vulnerability involving structured lattice KEMs, NIST will consider standardizing a KEM that is not based on lattices, after a fourth round of evaluation.

[14] See the discussion in Appendix C.

4.3.4 Saber

Saber is an IND-CCA2 KEM based on module learning with rounding (MLWR). Saber was selected as a finalist at the end of the second round.

Saber can be thought of as a variant of Regev's LWE encryption scheme [147] and differs in that it uses a module structured lattice and Learning with Rounding (LWR) instead of LWE. The LWR problem was defined by Banerjee, Peikert, and Rosen [157], with Saber citing [182], [236] as precedent for the use of modules in lattice cryptography.

Design. Like the LWE-style KEM candidate KYBER in the third round, Saber is constructed first as an IND-CPA-secure PKE scheme, then boosted to an IND-CCA-secure KEM by a version of the Fujisaki-Okamoto (FO) transform [170].

The base PKE scheme is derived from the MLWR problem. The ring is a cyclotomic power-of-2 ring, $R = \mathbb{Z}[X]/(X^{256}+1)$, and the module rank k is set to $k = 2, 3$, or 4 (corresponding to security categories 1, 3, 5). For each parameter set, Saber uses three integer moduli, p, q, and T, all powers of 2, $q = 2^{13}$ and $p = 2^{10}$ for all parameter sets, while T is 2^3, 2^4 or 2^6 (corresponding to security categories 1, 3, 5). Saber also uses a rounding operation Round_p, which can be thought of (roughly) as taking elements of $\mathbb{Z}_q$ and mapping them to $\mathbb{Z}_p$ by rounding to the nearest multiple of $\frac{q}{p}$ and dropping the $\log_2(q) - \log_2(p)$ lowest order bits to produce an $\log_2(p)$ bit quantity that can be thought of as an element of $\mathbb{Z}_p$. Similar operations Round_T and Round_2 map to elements of $\mathbb{Z}_T$ and $\mathbb{Z}_2$, respectively.

In key generation, a matrix $\mathbf{A} \in R_q^{k \times k}$ is sampled uniformly at random, while a short vector $\mathbf{s} \in R^k$ is sampled at random coefficient-wise from a centered binomial distribution. The public key is $pk := (\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \text{Round}_p(\mathbf{A}^{\mathrm{T}}\mathbf{s}))$, while the secret key is $\mathbf{s}$. Encryption and decryption instantiate a variant of the Lindner-Peikert paradigm [181]. To encrypt a message m (a 256-bit string), one samples coefficient-wise from a centered binomial distribution, a vector of polynomials $\mathbf{s}' \in R^k$. Then, the ciphertext c is formed as

$$c := (c_m, \mathbf{b}') := \left(\text{Round}_T(\mathbf{b}^{\mathrm{T}}\mathbf{s}') + mT/2, \text{Round}_p(\mathbf{A}\mathbf{s}')\right)$$

To decrypt a ciphertext c using the secret key $\mathbf{s}$, one computes $m = \text{Round}_2(\mathbf{b}'^{\mathrm{T}}\mathbf{s} - c_m)$.

Security. Saber's submission document gives a tight IND-CCA security proof in the random oracle model based on the decisional MLWR assumption and a loose proof in the quantum random oracle model. The Saber specification further suggests that it may be possible to provide a tighter security proof in the quantum random oracle model using the techniques of [237].

While MLWR does not have as extensive a network of security reductions as MLWE, there have been some results such as [238]. Moreover, all experimental investigations to date have indicated that MLWR (at least the MLWR instances relevant to cryptosystems like Saber) does not differ from MLWE in terms of the cryptanalytic techniques that are applicable or in terms of how successful those techniques are. Likewise, similar techniques, like the core SVP methodology, are used to estimate the concrete security of parameters for both MLWE and MLWR cryptosystems.

Performance. Saber's use of power of 2 moduli and rounding is intended to make implementation easier relative to other designs, such as KYBER, that use prime moduli and variants of LWE. This is particularly true for masked implementations which protect against side-channel attacks. The disadvantage of power of 2 moduli is that they do not allow an NTT implementation of polynomial multiplication. Despite these differences, Saber has a very similar performance profile to KYBER. It has fast key generation, encryption and decryption. Both schemes are typically the fastest or second fastest among the third-round candidates depending on the platform. Additionally, Saber has keys and ciphertexts that are about 10% smaller than those of KYBER for all of the 3 targeted security levels (see Table 6).

Significant events since Round 2. As with all of the lattice submissions, the best estimates of concrete security have been affected by ongoing research progress in lattice cryptanalysis. Nonetheless, Saber's parameters have stayed the same in the third round, as they have throughout the NIST PQC standardization process. In its third-round submission, Saber gave updated security estimates for its parameter sets, correcting an error pointed out on the pqc-forum during the second round. Saber also added some variants, a "90s version," modeled after KYBER's "90s version" and a uniform sampling version, which were described in their appendix. The Saber team also added discussion of side-channel attacks to their submission document citing a masked implementations of Saber [99].

During the third round, some improvements to the dual attack were proposed [166], leading to lower estimated security in the RAM model than was claimed in the Saber specification. These results suggest that all three Saber parameter sets fall slightly below the security targets for their claimed security levels when the cost of memory access for the attacker is not explicitly taken into account.

Overall assessment. Like the other structured lattice KEMs under consideration, Saber is a very efficient scheme whose security is supported by a large body of cryptographic research. Nonetheless, NIST determined that there was no compelling reason to standardize multiple different structured lattice KEMs and chose KYBER instead of Saber. One factor that led to this decision was NIST's assessment that the MLWE problem, which accounts for most of the security of KYBER, is better studied than the MLWR problem on which the security of Saber is entirely based. While it did not seem particularly likely that the use of MLWR as opposed to MLWE would result in a significant loss of security, KYBER and Saber were similar enough in security and performance profile that factors such as this could determine the decision.

4.4 Signatures Selected for Standardization

4.4.1 CRYSTALS-Dilithium

Dilithium is a lattice-based digital signature algorithm based on the Fiat-Shamir paradigm.

Design. Dilithium uses the ring $R_q := \mathbb{Z}_q[X]/(X^{256}+1)$, where q is the prime number

$2^{23} - 2^{13} + 1$. The public key for Dilithium is essentially a Module-LWE sample of the form $(\mathbf{A}, \mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$, where $\mathbf{A}$ is a matrix over R_q and $\mathbf{s}_1$ and $\mathbf{s}_2$ are error vectors over R_q. One distinctive feature of Dilithium is its error distribution: whereas lattice-based signature algorithms typically use a truncated Gaussian distribution to compute the coefficients in their error vectors, Dilithium uses a uniform distribution over $\{-\eta, -\eta + 1, \ldots, \eta\}$, where η is a small positive integer.

Dilithium is based on the "Fiat-Shamir with aborts" approach of Lyubashevsky [239]. At the core of this approach is a three-message lattice-based identification scheme that enables a prover to convince a verifier that they hold the secret key $(\mathbf{s}_1, \mathbf{s}_2)$ without revealing it. This begins with the prover computing a vector $\mathbf{w}$ consisting of the high-order bits of $\mathbf{A}\mathbf{y}$ (for random $\mathbf{y}$) and sending it to the verifier. The verifier responds with a random challenge polynomial $c \in R_q$ with small coefficients. The prover then responds with the vector $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$. The catch is that $\mathbf{z}$ may actually leak information about $\mathbf{s}_1$, so a careful rejection sampling step has to be added to ensure that $\mathbf{z}$ has coefficients of appropriate magnitude. In the end, the verifier accepts only if $\mathbf{A}\mathbf{z} \approx \mathbf{w} + c\mathbf{t}$.

To get a signature scheme, one applies the Fiat-Shamir transform. This amounts to having the prover generate c by hashing the commitment $\mathbf{w}$ together with the message μ. The actual Dilithium scheme involves a few additional optimizations. Notably, the public key is compressed by both the use of pseudorandomness and by omitting more than half of the low-order bits of $\mathbf{t}$. To make up for these dropped bits, the signer provides "hints" as part of each signature. These hints are essentially certain carries that allow the verifier to still correctly perform the check described above.

Security. The starting point for establishing the security of Dilithium is the decisional Module-LWE assumption, which suffices to show that the public key does not leak any information about the secret key. With an additional assumption called SelfTargetMSIS (a variant of the Module-SIS assumption) [240], one can show that Dilithium is strongly unforgeable (i.e., SUF-CMA) in the QROM.[15] An alternative version of Dilithium has been proved secure in the QROM based only on Module-LWE but at the cost of increasing the size of public keys by $\approx 5\times$ and signatures by $\approx 2\times$ [241]. Dilithium also satisfies several desirable "beyond unforgeability" security properties [173]. Notably, it satisfies a strong binding property that may be useful for non-repudiation: a given Dilithium signature can be identified with a unique public key and message.

As with other lattice-based schemes, the best-known attacks on Dilithium (not exploiting side-channels) amount to applying generic algorithms for finding short vectors in lattices. Under fairly conservative estimates, the core SVP security of Dilithium is 124, 186, and 265 for NIST levels 2, 3, and 5, respectively. Dilithium offers a number of options for varying parameters in order to increase security at the cost of either increased sizes and/or slower performance.

Performance. As noted above, pseudorandomness and truncated storage techniques are used to improve the performance of Dilithium. Additionally, for efficiency, elements of

[15] In the classical ROM, Dilithium can be proven secure using the standard Module-SIS assumption.

R_q are computed and stored using an NTT-based implementation for fast multiplication of polynomials. Dilithium is, along with FALCON, one of the two most efficient signature protocols in Round 3. FALCON generally has shorter keys and signatures than Dilithium (see Table 8), although Dilithium has the benefit of not requiring floating-point arithmetic. See subsection 2.2.2 for a detailed comparison between FALCON and Dilithium.

Significant events since Round 2. The Dilithium team made some minor changes and slightly adjusted parameter sets to better match NIST security levels.

During the third round, some improvements to the dual attack were proposed [165, 166], leading to lower estimated security in the RAM model than was claimed in the Dilithium specification. These results suggest that two of the three Dilithium parameter sets fall slightly below the security targets for their claimed security levels when the cost of memory access for the attacker is not explicitly taken into account.

Overall assessment. Dilithium is a signature scheme with high efficiency, relatively simple implementation, a strong theoretical security basis, and an encouraging cryptanalytic history. It is an excellent choice for a broad range of cryptographic applications and is, thus, the primary signature algorithm selected by NIST for standardization at this time.

4.4.2 Falcon

FALCON (Fast Fourier Lattice-based Compact Signatures over NTRU) is a lattice-based signature scheme utilizing the "hash-and-sign" paradigm.

Design. FALCON follows the GPV framework, introduced by Gentry, Peikert, and Vaikuntanathan in 2007 [242], for constructing hash-and-sign signature schemes from lattice-based trapdoor functions with preimage sampling. The FALCON submission builds on a sequence of works whose aim is to instantiate the GPV approach efficiently in NTRU lattices [243–245], with a particular focus on the compactness of the package consisting of one public key and one signature.

The instantiation of NTRU lattices in FALCON is relatively straightforward. Specifically, the secret is a set of polynomials $f, g, F, G \in \mathbb{Z}[x]/(x^n + 1)$ such that $fG - gF \equiv q$, and the public key is $h \equiv g \cdot f^{-1}$. For appropriately generated secrets, h will appear random while the bases

$$\begin{bmatrix} 1 & h \\ 0 & q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} f & g \\ F & G \end{bmatrix} \tag{4}$$

generate the same lattice.

Unlike the instantiation of NTRU lattices, the trapdoor preimage sampling algorithm of FALCON is fairly involved. In particular, its implementation requires the use of operations such as floating-point arithmetic, which leads to difficulties in secure implementations (e.g., for achieving constant-time signing) [246]. FALCON also has complex data structures, like the FALCON tree. This makes FALCON significantly more challenging to implement than

other lattice signature schemes (notably, Dilithium) [58]. NIST encourages further work on how to best implement FALCON, as well as on how to verify implementations.

Security. The theoretical security of FALCON is established by a proof of unforgeability in the QROM, based on the hardness of the SIS Problem over NTRU lattices (see subsection 3.2.3) [178]. Conservative estimates place the core SVP hardness of forging a FALCON signature at roughly the same level as for Dilithium (see Table 11). Parameterizing FALCON for intermediate security levels is possible but may require a different choice of modulus and ring, which could further complicate implementation.

It should be noted that FALCON does not offer certain desirable "beyond unforgeability" security properties [173]. However, a relatively simple transformation can add these properties to FALCON at a minimal performance cost [173].

As is the case with Dilithium, a secure implementation of FALCON will require side-channel protections (see [58, 247]).

Performance. FALCON has the smallest bandwidth (public key size plus signature size) among the third-round digital signature schemes (see Tables 8 and 9). FALCON is also fast when verifying a signature. Signing is somewhat slower than Dilithium and key generation is significantly slower. Due to its low bandwidth and fast verification, FALCON may be a superior choice in some constrained protocol scenarios.

Significant events since Round 2. The FALCON team has made some minor adjustments to parameters and algorithms in the FALCON specification. One notable change is that the signature encoding is now non-malleable and constant-size. The team has also expanded on the formal specification of the trapdoor sampling algorithm mentioned above.

Overall assessment. FALCON was chosen for standardization because NIST has confidence in its security (under the assumption that it is correctly implemented) and because its small bandwidth may be necessary in certain applications.

4.4.3 SPHINCS$^+$

SPHINCS$^+$ is a stateless hash-based signature scheme.

Design. The scheme combines the use of one-time signatures, few-times signatures, Merkle trees, and hypertrees to construct a digital signature scheme that is suitable for general use. It does not require the user to keep track of any state between signatures. In contrast, there are also stateful hash-based signature schemes which are faster and produce smaller signatures but require the user to keep state across signatures with disastrous consequences if the state is mismanaged. Like Picnic, the security of SPHINCS$^+$ is based only on the security of the underlying symmetric primitives. However, unlike Picnic, SPHINCS$^+$ is defined to use a standard hash function like SHA-256, rather than a new block cipher optimized for efficient multiparty computation. SPHINCS$^+$ includes parameter sets based on three different hash functions: SHAKE256, SHA-256, and Haraka.

SPHINCS$^+$ is a complex scheme that involves many different parameters for each security category. Each set of parameters determines some trade-off between the complexity of different steps of the signing and verification process and the size of the final signature. The designers of SPHINCS$^+$ have considered a wide range of parameter set choices and have proposed two sets for each security category. One set makes the signatures faster at the cost of larger signatures, and the other set makes the signatures smaller at the cost of slower signatures. While these parameter sets are well-suited for most general-purpose uses of SPHINCS$^+$, it is possible to make other more extreme trade-offs (e.g., making signatures very slow in order to make the signature a couple thousand bytes shorter) that might be sensible in some cases.

The design of SPHINCS$^+$ imposes a limit on the number of allowable signatures from a given public key. For any number of signatures using the public key, g, there is some very low probability that they will reveal enough of the private key to allow an attacker to forge a signature. As g grows, so does the probability of this disaster. The total number of signatures must be kept low enough that this probability remains negligible. NIST's call for proposals [9] required the ability to securely perform 2^{64} signatures, which imposes requirements on the parameters of SPHINCS$^+$. A smaller maximum number of signatures would result in somewhat smaller and faster signatures. NIST intends to ask for public feedback on whether such a version of SPHINCS$^+$ would be beneficial.

Security. The complexity of SPHINCS$^+$ is a potential issue for implementation security and also for evaluating the security of the whole scheme (since an error in the specification or design is easier to miss in a more complicated algorithm). In contrast, the cryptographic security of SPHINCS$^+$ relies only on the security of the underlying hash functions used [248–250]. This security assumption is independent of the ones on which other finalist signature schemes (like Dilithium and FALCON) are based, so SPHINCS$^+$ provides a useful fallback in case of unforeseen cryptanalytic attacks. The difficulty of protecting SPHINCS$^+$ from side-channel attacks is mostly determined by the difficulty of protecting a keyed hash implementation from side-channel attacks.

Performance. Because of the way SPHINCS$^+$ signatures are formed, key generation and verification are much faster than signing. SPHINCS$^+$ public keys are very short, but SPHINCS$^+$ signatures are quite long. Even for Category 1 security, the smallest (and slowest) parameter choices yield a signature of about 8 KiB – far larger than alternative signature schemes such as FALCON or Dilithium. See Table 9.

Significant events since Round 2. At the beginning of the Round 3, new parameter sets were selected for security categories 1 and 3. In addition, a flaw was discovered in the security reduction for SPHINCS$^+$, which was corrected during the third round [250]. In January 2022, the SPHINCS$^+$ team announced tweaks to the key generation and signing procedures in order to protect against multi-user attacks [251].

During the third round, two attacks were discovered which called into question the claimed category 5 security of parameter sets using SHA-256. The first attack was described on the pqc-forum in February 2021 [252], and a month later the SPHINCS$^+$ team

proposed a patch [253]. The second attack [254] was announced on the pqc-forum in April 2022. This attack similarly affects claimed category 5 parameters of SPHINCS$^+$ using SHA-256, but is not mitigated by the previously proposed patch.

Overall Assessment. While our existing stateful hash-based signature standards, XMSS and LMS, are based on similar assumptions to SPHINCS$^+$, the requirement to keep state in XMSS and LMS makes them more difficult to implement in a way that avoids misuse (see [255]). SPHINCS$^+$ was selected for standardization because it provides a workable (albeit rather large and slow) signature scheme whose security seems quite solid and is based on an entirely different set of assumptions than those of our other signature schemes to be standardized.

The two attacks related to SHA-256-based parameters claiming category 5 security will need to be carefully considered when selecting which parameters of SPHINCS$^+$ to standardize. In both cases, the underlying issue is that, due to its 256-bit internal state, SHA-256 is not well designed to provide more than category 2 security in a wide variety of circumstances. While some applications of SHA-256 do appear to provide more security strength than this, gaining confidence in a SHA-256-based construction claiming more than category 2 security will require a security proof that explicitly considers that SHA-256 is a Merkle-Damgård hash with a Davies-Meyer compression function. The existing security proof for SPHINCS$^+$ does not analyze the internal structure of the hash functions it uses. Ignoring the internal structure of the hash function is better motivated for the SHAKE256 parameter sets, due to results such as [256].

4.5 Signatures no longer being considered

4.5.1 G*e*MSS

G*e*MSS (a Gr*e*at Multivariate Short Signature) is a signature scheme that follows the hash-and-sign paradigm with the application of Feistel–Patarin iterations. G*e*MSS uses a trapdoor function based on Hidden Field Equation with Vinegar variables and the Minus modifier (HFEv-).

Design. G*e*MSS belongs to the big field family of multivariate cryptosystems. The basic idea of these schemes is to use a bijective mapping between $GF(q^n)$ and $GF(q)^n$ so that the multivariate trapdoor function (expressed in terms of the small field $GF(q)$) can be re-expressed as a univariate function over the big field, $GF(q^n)$. So expressed, the function can be efficiently inverted. To produce a public key, the function is composed with linear maps over the small field, which is presumed to hide the structure. The HFE cryptosystem [257] was introduced after the original big field scheme of Matsumoto and Imai [258] was broken by [259]. However, HFE with secure parameters has very slow signing. The Vinegar and Minus modifier were added by [260] in an effort to increase security at little cost in performance.

Security. The security of G*e*MSS depends on multiple assumptions. It is assumed that, on

average, instances of G*e*MSS produce hard instances of the $\mathcal{MQ}$ problem in, the context of directly inverting the public key, and of the MinRank problem (see Section 3.2.2).

Performance. Like most other multivariate schemes, G*e*MSS produces small signatures but has a very large public key (see Table 9). Compared to Rainbow, the submitted parameters for G*e*MSS yield slightly smaller signatures, but the public key is significantly larger, and the signing and key generation operations are significantly slower. G*e*MSS defines six sets of parameters: G*e*MSS, BlueG*e*MSS, RedG*e*MSS, WhiteG*e*MSS, CyanG*e*MSS and MagentaG*e*MSS. The WhiteG*e*MSS, CyanG*e*MSS and MagentaG*e*MSS parameter sets were added in the third round and use fewer rounds in the Feistel-Patarin construction than the G*e*MSS, BlueG*e*MSS and RedG*e*MSS parameter sets. G*e*MSS and WhiteG*e*MSS rely the least (although still significantly) on the vinegar and minus modifiers for their security. These parameter sets have the slowest signing algorithms as a result. RedG*e*MSS and MagentaG*e*MSS rely the most on the vinegar and minus modifiers and are the fastest. BlueG*e*MSS and CyanG*e*MSS are intermediate.

Significant events since Round 2. In Round 3, G*e*MSS suffered a catastrophic key-recovery attack (see [12, 109]). The attack introduces a new MinRank instance whose resolution reveals the structure of the private key. While previous MinRank attacks on HFE schemes model MinRank in essentially the same way (see [261–263]), they are all exponential in the number of vinegar variables and the number of removed equations. In contrast, the attack of [109] is polynomial in the number of vinegar variables and is not affected greatly by the number of removed equations. This attack is further improved by the techniques in [264], where it is shown how to implement the much more efficient support minors MinRank approach [144] in the case that the solution is in an extension field.

Overall Assessment. This cryptanalysis effectively establishes that the vinegar and minus modifiers fail to provide any substantial security benefit in an HFEv- construction. The result undermines the basic design principles of HFEv-. Possible modifications to repair the scheme – such as abandoning the vinegar and minus modifiers and increasing the degree of the HFE polynomial to reach the target security level or adding a projection or plus modifier to thwart the new attacks, as suggested in [265] – would both represent too large a change to the original submission and render the performance of the resulting scheme unacceptable, as shown in [264]. Therefore, NIST decided not to advance G*e*MSS.

4.5.2 Picnic

A Picnic signature is a non-interactive zero-knowledge proof of knowledge of a secret key bound to the message being signed. Picnic was an alternate signature scheme during the third round.

Design. Picnic uses a symmetric block cipher called *LowMC*. A circuit C takes as input a plaintext block p and a secret key sk, and outputs LowMC(sk, p). A randomly chosen plaintext block p serves as a public key. LowMC was designed so as to allow an XOR-

AND circuit with fewer AND gates than other ciphers such as AES. The "number of AND gates required" metric is called *multiplicative complexity* [266]. Zero-knowledge proofs of knowledge of the input to an AND-XOR circuit, given its output, are of length proportional to the number of AND gates of the circuit. AES can be computed with 32 AND gates per S-box [267] (it is not known if it can be done with fewer AND gates). This results in over 5000 AND gates in AES-128. A comparable LowMC parametrization uses under 1000 AND gates [268].

A Picnic signature is a non-interactive zero-knowledge proof of knowledge of the secret key. The message being signed is incorporated (via hashing) into the challenges of the proof of knowledge in such a way that only the holder of the secret key can produce the proof. The length of the signature depends on the multiplicative complexity of the encryption scheme and the MPCitH (multi-party computation in the head) technique to construct a zero-knowledge proof of knowledge from the field of secure multi-party computation (see [269]).

Picnic is a highly modular design. The cryptographic primitives – a hash function and block cipher – could be instantiated in different ways. LowMC has not been studied as much as AES and hence needs much more analysis before it can be standardized by NIST. However, the security requirements for the underlying block cipher in Picnic are much less stringent than the general security requirements of a block cipher, as only a single plaintext/ciphertext pair is ever revealed, and an attacker needs to find a key that maps that plaintext to that ciphertext in order to forge Picnic signatures.

Security. Picnic uses no number-theoretic or structured hardness assumptions. Its security depends on the underlying hash function behaving as a random oracle (a common cryptographic assumption) and on the security of the LowMC block cipher [268] against an adversary given a single plaintext/ciphertext pair. The security of LowMC has not been as extensively studied as that of older symmetric-key ciphers, although recent attempts to analyze LowMC's security have found weaknesses [270–275].

As with other candidates, a straightforward implementation of Picnic would have significant side-channel issues (see [97, 276]).

Performance. Picnic has a small public key size and relatively large signatures. Signing speed is much faster than SPHINCS$^+$, and verification is somewhat slower.

Significant events since Round 2. As noted in the Security section above, there were several papers which cryptanalyzed LowMC [270–275].

Variants of Picnic based on AES have been proposed [277]. The signature scheme Banquet [278] uses AES and achieves performance close to that of Picnic. Obtaining further improvements under the same paradigm as Picnic is an active area of research (see, for example, [279, 280]) and may eventually lead to a signature scheme with significantly better performance than the current design.

Overall assessment. Picnic and SPHINCS$^+$ were the two candidate signature schemes that relied mostly on the security of symmetric primitives. NIST chose SPHINCS$^+$ largely

because it could not confidently quantify the security of LowMC (see, for example, [271–273]) and because future cryptosystems that evolve out of the multi-party-computation-in-the-head paradigm may eventually prove significantly superior to the third-round Picnic design.

4.5.3 Rainbow

Rainbow is a multivariate signature scheme using the hash-and-sign paradigm with the modification of [281]. Rainbow is a layered generalization of the unbalanced oil-vinegar (UOV) scheme.

Design. Rainbow belongs to the small field family of multivariate cryptosystems and to the lineage of oil-vinegar schemes such as UOV (see [282]). UOV schemes use two types of variables – oil variables and vinegar variables – to generate a multivariate quadratic map for which preimages are easily computed. Specifically, this map contains terms that are quadratic in the vinegar variables and terms that are bilinear in the oil and vinegar variables but contains no terms that are quadratic in exclusively the oil variables. In this way, the owner of the private key can randomly assign values to the vinegar variables and solve linearly for values of the oil variables.

Rainbow generalizes this basic construction by defining layers with differing sets of oil variables that can be sequentially solved, layer by layer (see [283]). The entire map is then composed with linear maps to hide the structure. The use of layers in the Rainbow construction allows smaller signatures and faster verification than traditional UOV at the cost of extra structure.

Security. Rainbow's security depends on several hardness assumptions. It is assumed that, on average, instances of Rainbow produce hard instances of several problems including the $\mathcal{MQ}$ problem, in the context of directly inverting the public key, and the MinRank problem (see subsection 3.2.2).

Performance. Rainbow has efficient signing and verification, and produces very short signatures. Rainbow key generation, however, is significantly slower than signing or verifying. Still, key generation is comparable to that of FALCON. The key sizes of Rainbow parameters are quite large in comparison to other finalists but are still significantly smaller than G*e*MSS (see Tables 8 and 9).

Significant events since Round 2. Parameters were updated between Round 2 and Round 3 due to updated analysis in [284] showing that the second round parameters of Rainbow were very slightly below the NIST security categories. A new method of generating a MinRank problem from the Rainbow public key was discovered in [11]. Together with the support minors method of solving MinRank instances (see [144]), this new "rectangular MinRank attack" reduced the complexity of the best known attacks on the various Rainbow parameter sets by 20 to 55 bits of security in the gate metric, showing that all of the Rainbow parameter sets fall significantly below their security targets, when memory costs

are ignored. Subsequently, an analysis incorporating the significant memory access cost of this attack [264] suggests that all of the Rainbow parameters fail to meet their purported security levels, even when memory costs are taken into account. Finally, a new attack [13] provides a new hybrid combinatorial/algebraic attack and an improvement of the rectangular MinRank attacks that further reduces the security of all parameter sets to the extent that an attack on Rainbow-I has become practical.

Overall assessment. The best known attacks of Rainbow have significantly affected the security of the scheme. In fact, in light of the new attacks, it is not clear that secure instantiations of Rainbow can offer any performance advantage in comparison to the less structured UOV scheme without significant re-engineering. Therefore, NIST decided not to advance Rainbow.

5. Conclusion

NIST greatly appreciates the participation in the NIST PQC Standardization Process. It has been a long and complex process so far. Six years have passed since NIST issued its Call for Proposals for PQC algorithms, and there have been significant efforts from submitters, researchers, implementers, industry, and the cryptographic community. With the conclusion of the third round, NIST is pleased to announce the first public-key algorithms that will provide protection from quantum attacks to be standardized.

The primary algorithms NIST recommends for most use cases are CRYSTALS–KYBER (key-establishment) and CRYSTALS–Dilithium (digital signatures). In addition, the signature schemes FALCON and SPHINCS$^+$ will also be standardized. The candidates BIKE, Classic McEliece, HQC, and SIKE will all continue for further study in a fourth round of evaluation. The reasons for these choices were provided earlier in this report.

NIST will create new draft standards for these algorithms, with coordination of the submission teams to ensure that the standards are in agreement with the specifications. As part of the drafting process, NIST will seek input on which specific parameter sets to include, particularly for any at security category 1. When finished, the standards will be posted for public comment. After the close of the comment period, NIST will revise the draft standards as appropriate based on the feedback received. A final review, approval, and promulgation process will then follow. NIST hopes to publish the completed standard by 2024.

The fourth round of evaluation and analysis will proceed similar to the earlier rounds. As before, the four candidate algorithms will be allowed to make relatively minor modifications to their submissions, which must be submitted to NIST, and must meet the same requirements as defined in [9]. Further details and instructions will be provided on the pqc-forum. After the fourth round concludes, NIST may decide to select some of the fourth round candidates for standardization.

As first indicated in [7] and emphasized during the third round:

> "NIST is pleased with the progress of the PQC standardization effort but rec-

> ognizes that current and future research may lead to promising schemes which were not part of the NIST PQC Standardization Project. NIST may adopt a mechanism to accept such proposals at a later date. In particular, NIST would be interested in a general-purpose digital signature scheme which is not based on structured lattices."

NIST plans to issue a new Call for Proposals for public-key (quantum-resistant) digital signature algorithms in the summer of 2022. NIST primarily seeks to diversify its signature portfolio with non-structured lattice signature schemes. NIST may also be interested in signature schemes that have short signatures and fast verification. Submissions in response to this call will be due in 2023. Submitters are encouraged to communicate with NIST ahead of time. NIST will decide which (if any) of the submitted signature algorithms to accept and initiate a new process for evaluation. NIST expects this process to be much smaller in scope than the current PQC process. The signature schemes accepted to this process will need to be thoroughly analyzed, which will similarly take several years.

Even though the third round is ending and NIST will begin to draft the first PQC standards, standardization efforts in this area will continue for some time. This should not be interpreted to mean that users should wait to adopt post-quantum algorithms. NIST hopes for rapid adoption of these first standardized algorithms and will issue future guidance on the transition. The transition will undoubtedly have many complexities, and there will be challenges for some use cases, such as IoT devices or Certificate Transparency [285]. The National Cybersecurity Center of Excellence has initiated a project to develop practices to ease some of the anticipated migration challenges [286, 287].

NIST plans to host a 4th NIST PQC Standardization Conference in the winter of 2022. More details will be provided at a later date.

Once again, NIST is grateful to the community for all of the research, support, and analysis provided during the first three rounds. These efforts have been indispensable in helping NIST during the PQC standardization process.

References

[1] National Institute of Standards and Technology (2013) Digital signature standard (DSS) (U.S. Department of Commerce, Washington, D.C.), Federal Information Processing Standards (FIPS) Publication 186-4. https://doi.org/10.6028/NIST.FIPS.186-4

[2] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R (2018) Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography (U.S. Department of Commerce, Washington, D.C.), Special Publication 800-56A Revision 3. https://doi.org/10.6028/NIST.SP.800-56Ar3

[3] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R, Simon S (2019) Recommendation for pair-wise key-establishment using integer factorization cryptography (U.S. Department of Commerce, Washington, D.C.), Special Publication 800-56B Revision 2. https://doi.org/10.6028/NIST.SP.800-56Br2

[4] National Institute of Standards and Technology (2016) Announcing request for nominations for public-key post-quantum cryptographic algorithms. *Federal Register* 81(244):92787–92788. https://federalregister.gov/a/2016-30615.

[5] National Institute of Standards and Technology (2016) NIST post-quantum cryptography standardization. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

[6] Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Liu YK, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D (2019) Status report on the first round of the NIST post-quantum cryptography standardization process (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8240. https://doi.org/10.6028/NIST.IR.8240

[7] Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Kelsey J, Liu YK, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D (2020) Status report on the second round of the NIST post-quantum cryptography standardization process (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8309. https://doi.org/10.6028/NIST.IR.8309

[8] Chen L, Jordan S, Liu YK, Moody D, Peralta R, Perlner R, Smith-Tone D (2016) Report on post-quantum cryptography (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8105. https://doi.org/10.6028/NIST.IR.8105

[9] National Institute of Standards and Technology (2016) Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. Available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[10] Schank J (2021) Category 5 NTRU parameters. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/t1JCgzSS-uk/m/VXXQaJgFCQAJ.

[11] Beullens W (2021) Improved cryptanalysis of UOV and Rainbow. *Advances in Cryptology – EUROCRYPT 2021*, eds Canteaut A, Standaert FX (Springer International

Publishing, Cham), pp 348–373.

[12] Tao C, Petzoldt A, Ding J (2021) Efficient key recovery for all HFE signature variants. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 70–93.

[13] Beullens W (2022) Breaking Rainbow takes a weekend on a laptop, Cryptology ePrint Archive, Report 2022/214. https://ia.cr/2022/214.

[14] Avanzi R, Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, Schwabe P, Seiler G, Stehlé D (2020) CRYSTALS-KYBER algorithm specifications and supporting documentation, 3rd Round submission to the NIST's post-quantum cryptography standardization process. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[15] NTRU Prime Risk-Management Team (2021) Risks of lattice KEMs. https://ntruprime.cr.yp.to/latticerisks-20211031.pdf.

[16] (2022) NIST pqc-forum mailing list. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum.

[17] Drucker N, Gueron S, Kostic D, Persichetti E (2021) On the applicability of the Fujisaki–Okamoto transformation to the BIKE KEM. *International Journal of Computer Mathematics: Computer Systems Theory* 6(4):364–374. https://doi.org/10.1080/23799927.2021.1930176

[18] Kostic D (2020) *Analysis of the BIKE post-quantum cryptographic protocols and the Legendre pseudorandom function*. Ph.D. thesis. École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. https://doi.org/10.5075/epfl-thesis-7212

[19] Becker H, Hwang V, Kannwischer MJ, Yang BY, Yang SY (2021) Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1, Cryptology ePrint Archive, Report 2021/986. https://ia.cr/2021/986.

[20] Nguyen DT, Gaj K (2021) Fast NEON-based multiplication for lattice-based NIST post-quantum cryptography finalists. *Post-Quantum Cryptography*, eds Cheon JH, Tillich JP (Springer International Publishing, Cham), pp 234–254.

[21] Chung CMM, Hwang V, Kannwischer MJ, Seiler G, Shih CJ, Yang BY (2021) NTT multiplication for NTT-unfriendly rings: New speed records for Saber and NTRU on Cortex-M4 and AVX2. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(2):159–188. https://doi.org/10.46586/tches.v2021.i2.159-188

[22] Beckwith L, Nguyen DT, Gaj K (2021) High-performance hardware implementation of CRYSTALS-Dilithium, Cryptology ePrint Archive, Report 2021/1451. https://ia.cr/2021/1451.

[23] Land G, Sasdrich P, Güneysu T (2021) A hard crystal - implementing Dilithium on reconfigurable hardware, Cryptology ePrint Archive, Report 2021/355. https://ia.cr/2021/355.

[24] Ricci S, Malina L, Jedlicka P, Smékal D, Hajny J, Cibik P, Dzurenda P, Dobias P (2021) Implementing CRYSTALS-Dilithium signature scheme on FPGAs. *The 16th International Conference on Availability, Reliability and Security* ARES 2021 (Association for Computing Machinery, New York, NY, USA), pp 1–11. https://doi.org/

10.1145/3465481.3465756

[25] Sanal P, Karagoz E, Seo H, Azarderakhsh R, Mozaffari-Kermani M (2021) Kyber on ARM64: Compact implementations of Kyber on 64-bit ARM Cortex-A processors, Cryptology ePrint Archive, Report 2021/561. https://ia.cr/2021/561.

[26] Wang B, Gu X, Yang Y (2020) Saber on ESP32. *Applied Cryptography and Network Security*, eds Conti M, Zhou J, Casalicchio E, Spognardi A (Springer International Publishing, Cham), pp 421–440.

[27] Gonzalez R, Hülsing A, Kannwischer MJ, Krämer J, Lange T, Stöttinger M, Waitz E, Wiggers T, Yang BY (2021) Verifying post-quantum signatures in 8 kB of RAM. *Post-Quantum Cryptography*, eds Cheon JH, Tillich JP (Springer International Publishing, Cham), pp 215–233.

[28] Chen MS, Chou T (2021) Classic McEliece on the ARM Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(3):125–148. https://doi.org/10.46586/tches.v2021.i3.125-148

[29] Roth J, Karatsiolis E, Krämer J (2021) Classic McEliece implementation with low memory footprint. *Smart Card Research and Advanced Applications*, eds Liardet PY, Mentens N (Springer International Publishing, Cham), pp 34–49.

[30] Greconici DOC, Kannwischer MJ, Sprenkels D (2020) Compact Dilithium implementations on Cortex-M3 and Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(1):1–24. https://doi.org/10.46586/tches.v2021.i1.1-24

[31] Niederhagen R, Roth J, Wälde J (2021) Streaming SPHINCS+ for embedded devices using the example of TPMs, Cryptology ePrint Archive, Report 2021/1072. https://ia.cr/2021/1072.

[32] (2020) Round 2 embedded energy results and general comments. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/UH-8hkFODac/m/ydY1y_5-AAAJ.

[33] Bernstein D, Lange T (eds.), eBACS: ECRYPT Benchmarking of Cryptographic Systems - SUPERCOP (2020). Available at https://bench.cr.yp.to/supercop.html.

[34] Open quantum safe (OQS) algorithm performance visualizations. Available at https://openquantumsafe.org/benchmarking.

[35] pqm4: Post-quantum crypto library for the ARM Cortex-M4 (2020). Available at https://github.com/mupq/pqm4.

[36] Bernstein DJ (2021) Kyber's inefficiency: some data points. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ik1pXoAZk8s/m/hTt5Z1v8AQAJ.

[37] Cheng H, Großschädl J, Rønne PB, Ryan PYA (2021) A lightweight implementation of NTRUEncrypt for 8-bit AVR microcontrollers, https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/grobschadl-lighteight-implmentation-NTRUE.pdf. Third NIST PQC Standardization Conference.

[38] Guillen OM, Pöppelmann T, Mera JMB, Bongenaar EF, Sigl G, Sepulveda J (2017) Towards post-quantum security for IoT endpoints with NTRU. *Proceedings of the*

Conference on Design, Automation & Test in Europe DATE '17 (European Design and Automation Association, Leuven, BEL), p 698–703.

[39] Boorghany A, Sarmadi SB, Jalili R (2015) On constrained implementation of lattice-based cryptographic primitives and schemes on smart cards. *ACM Transactions on Embedded Computing Systems* 14(3). https://doi.org/10.1145/2700078

[40] Xin G, Han J, Yin T, Zhou Y, Yang J, Cheng X, Zeng X (2020) VPQC: A domain-specific vector processor for post-quantum cryptography based on RISC-V architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers* 67(8):2672–2684.

[41] Banerjee U, Ukyab TS, Chandrakasan AP (2019) Sapphire: A configurable crypto-processor for post-quantum lattice-based protocols. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019(4):17–61. https://doi.org/10.13154/tches.v2019.i4.17-61

[42] Basu K, Soni D, Nabeel M, Karri R (2019) NIST post-quantum cryptography- a hardware evaluation study, Cryptology ePrint Archive, Report 2019/047. https://ia.cr/2019/047.

[43] Fritzmann T, Sigl G, Sepúlveda J (2020) RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020(4):239–280. https://doi.org/10.13154/tches.v2020.i4.239-280

[44] Alkim E, Evkan H, Lahr N, Niederhagen R, Petri R (2020) ISA extensions for finite field arithmetic - accelerating Kyber and NewHope on RISC-V, Cryptology ePrint Archive, Report 2020/049. https://ia.cr/2020/049.

[45] Dang VB, Farahmand F, Andrzejczak M, Gaj K (2019) Implementing and benchmarking three lattice-based post-quantum cryptography algorithms using software/hardware codesign. *2019 International Conference on Field-Programmable Technology (ICFPT)*, pp 206–214. https://doi.org/10.1109/ICFPT47387.2019.00032

[46] Dang VB, Mohajerani K, Gaj K (2021) High-speed hardware architectures and FPGA benchmarking of CRYSTALS-Kyber, NTRU, and Saber, Cryptology ePrint Archive, Report 2021/1508. https://ia.cr/2021/1508.

[47] Farahmand F, Dang VB, Nguyen DT, Gaj K (2019) Evaluating the potential for hardware acceleration of four NTRU-based key encapsulation mechanisms using software/hardware codesign. *Post-Quantum Cryptography*, eds Ding J, Steinwandt R (Springer International Publishing, Cham), pp 23–43.

[48] Bisheh-Niasar M, Azarderakhsh R, Mozaffari-Kermani M (2021) A monolithic hardware implementation of Kyber: Comparing apples to apples in PQC candidates. *Progress in Cryptology – LATINCRYPT 2021*, eds Longa P, Ràfols C (Springer International Publishing, Cham), pp 108–126.

[49] Jati A, Gupta N, Chattopadhyay A, Sanadhya SK (2021) A configurable Crystals-Kyber hardware implementation with side-channel protection, Cryptology ePrint Archive, Report 2021/1189. https://ia.cr/2021/1189.

[50] Yarman F, Mert AC, Öztürk E, Savaş E (2021) A hardware accelerator for poly-

nomial multiplication operation of CRYSTALS-KYBER PQC scheme. *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp 1020–1025. https://doi.org/10.23919/DATE51398.2021.9474139

[51] Xing Y, Li S (2021) A compact hardware implementation of CCA-secure key exchange mechanism CRYSTALS-KYBER on FPGA. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(2):328–356. https://doi.org/10.46586/tches.v2021.i2.328-356

[52] Qin Z, Tong R, Wu X, Bai G, Wu L, Su L (2021) A compact full hardware implementation of PQC algorithm NTRU. *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp 792–797. https://doi.org/10.1109/CISCE52179.2021.9446042

[53] Sinha Roy S, Basso A (2020) High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: Saber in hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020(4):443–466. https://doi.org/10.13154/tches.v2020.i4.443-466

[54] Hülsing A, Ning K, Schwabe P, Weber F, Zimmermann PR (2021) Post-quantum WireGuard. *2021 IEEE Symposium on Security and Privacy (SP)* (IEEE Computer Society, Los Alamitos, CA, USA), pp 304–321. https://doi.org/10.1109/SP40001.2021.00030

[55] Shulman H, Goodman J, Housley R, Kaliski B, Risk V, Stebila D, van Rijswijk-Deij R (2021) PANEL: PQC considerations for DNSSEC, Third PQC Standardization Conference. Available at https://www.nist.gov/video/third-pqc-standardization-conference-session-v-applications.

[56] Bindel N (2021) Suitability of 3rd round signature candidates for vehicle-to-vehicle communication, Workshop Record of the Third PQC Standardization Conference. Available at https://csrc.nist.gov/Presentations/2021/suitability-of-3rd-round-signature-candidates-for.

[57] Beckwith L, Nguyen DT, Gaj K (2022) High-performance hardware implementation of lattice-based digital signatures, Cryptology ePrint Archive, Report 2022/217. https://ia.cr/2022/217.

[58] Pornin T (2019) New efficient, constant-time implementations of Falcon, Cryptology ePrint Archive, Report 2019/893. https://ia.cr/2019/893.

[59] Greuet A (2021) Smartcard and post-quantum crypto, Workshop Record of the Third PQC Standardization Conference. Available at https://csrc.nist.gov/Presentations/2021/smartcard-and-post-quantum-crypto.

[60] Bernstein DJ, Brumley BB, Chen MS, Chuengsatiansup C, Lange T, Marotzke A, Peng BY, Tuveri N, van Vredendaal C, Yang BY (2020) NTRU Prime: round 3, Submission to the NIST's post-quantum cryptography standardization process.

[61] Gross H, Schaffenrath D, Mangard S (2017) Higher-order side-channel protected implementations of KECCAK, Cryptology ePrint Archive, Report 2017/395. https://ia.cr/2017/395.

[62] Albrecht MR, Deo A, Paterson KG (2018) Cold boot attacks on ring and module

LWE keys under the NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018(3):173–213. https://doi.org/10.13154/tches.v2018.i3.173-213

[63] Migliore V, Gérard B, Tibouchi M, Fouque PA (2019) Masking Dilithium. *Applied Cryptography and Network Security*, eds Deng RH, Gauthier-Umaña V, Ochoa M, Yung M (Springer International Publishing, Cham), pp 344–362.

[64] Ravi P, Roy DB, Bhasin S, Chattopadhyay A, Mukhopadhyay D (2019) Number "not used" once - practical fault attack on pqm4 implementations of NIST candidates. *Constructive Side-Channel Analysis and Secure Design*, eds Polian I, Stöttinger M (Springer International Publishing, Cham), pp 232–250.

[65] Liu Y, Zhou Y, Sun S, Wang T, Zhang R, Ming J (2021) On the security of lattice-based Fiat-Shamir signatures in the presence of randomness leakage. *IEEE Transactions on Information Forensics and Security* 16:1868–1879. https://doi.org/10.1109/TIFS.2020.3045904

[66] Ravi P, Jhanwar MP, Howe J, Chattopadhyay A, Bhasin S (2019) Exploiting determinism in lattice-based signatures: Practical fault attacks on pqm4 implementations of NIST candidates. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security* Asia CCS '19 (Association for Computing Machinery, New York, NY, USA), p 427–440. https://doi.org/10.1145/3321705.3329821

[67] Schneider T, Paglialonga C, Oder T, Güneysu T (2019) Efficiently masking binomial sampling at arbitrary orders for lattice-based crypto. *Public-Key Cryptography – PKC 2019*, eds Lin D, Sako K (Springer International Publishing, Cham), pp 534–564.

[68] Lahr N, Niederhagen R, Petri R, Samardjiska S (2020) Side channel information set decoding using iterative chunking. *Advances in Cryptology – ASIACRYPT 2020*, eds Moriai S, Wang H (Springer International Publishing, Cham), pp 881–910.

[69] Dachman-Soled D, Ducas L, Gong H, Rossi M (2020) LWE with side information: Attacks and concrete security estimation. *Advances in Cryptology – CRYPTO 2020*, eds Micciancio D, Ristenpart T (Springer International Publishing, Cham), pp 329–358.

[70] Apon D (2020) Passing the final checkpoint! NIST PQC 3rd round begins. https://www.scribd.com/document/474476570/PQC-Overview-Aug-2020-NIST.

[71] Ravi P, Bhasin S, Sinha Roy S, Chattopadhyay A (2020) On exploiting message leakage in (few) NIST PQC candidates for practical message recovery and key recovery attacks, Cryptology ePrint Archive, Report 2020/1559. https://ia.cr/2020/1559.

[72] Bache F, Paglialonga C, Oder T, Schneider T, Güneysu T (2020) High-speed masking for polynomial comparison in lattice-based KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020(3):483–507. https://doi.org/10.13154/tches.v2020.i3.483-507

[73] Xu Z, Pemberton O, Sinha Roy S, Oswald D (2020) Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber, Cryptology ePrint Archive, Report 2020/912. https://ia.cr/2020/912.

[74] Pessl P, Prokop L (2021) Fault attacks on CCA-secure lattice KEMs. *IACR*

Transactions on Cryptographic Hardware and Embedded Systems 2021(2):37–60. https://doi.org/10.46586/tches.v2021.i2.37-60

[75] Ravi P, Ezerman MF, Bhasin S, Chattopadhyay A, Sinha Roy S (2021) Will you cross the threshold for me? - Generic side-channel assisted chosen-ciphertext attacks on NTRU-based KEMs, Cryptology ePrint Archive, Report 2021/718. https://ia.cr/2021/718.

[76] Howe J, Apon D (2021) Attacks on NIST PQC 3rd round candidates, Real World Crypto 2021. https://iacr.org/submit/files/slides/2021/rwc/rwc2021/22/slides.pdf.

[77] Ngo K, Dubrova E, Guo Q, Johansson T (2021) A side-channel attack on a masked IND-CCA secure Saber KEM implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(4):676–707. https://doi.org/10.46586/tches.v2021.i4.676-707

[78] Bhasin S, D'Anvers JP, Heinz D, Pöppelmann T, Van Beirendonck M (2021) Attacking and defending masked polynomial comparison for lattice-based cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(3):334–359. https://doi.org/10.46586/tches.v2021.i3.334-359

[79] Horlemann AL, Puchinger S, Renner J, Schamberger T, Wachter-Zeh A (2021) Information-set decoding with hints, Cryptology ePrint Archive, Report 2021/279. https://ia.cr/2021/279.

[80] Bos JW, Gourjon M, Renes J, Schneider T, van Vredendaal C (2021) Masking Kyber: First- and higher-order implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(4):173–214. https://doi.org/10.46586/tches.v2021.i4.173-214

[81] Xagawa K, Ito A, Ueno R, Takahashi J, Homma N (2021) Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. *Advances in Cryptology – ASIACRYPT 2021*, eds Tibouchi M, Wang H (Springer International Publishing, Cham), pp 33–61.

[82] Ueno R, Xagawa K, Tanaka Y, Ito A, Takahashi J, Homma N (2021) Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2022(1):296–322. https://doi.org/10.46586/tches.v2022.i1.296-322

[83] Ngo K, Dubrova E, Johansson T (2021) Breaking masked and shuffled CCA secure Saber KEM by power analysis. *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security* (Association for Computing Machinery, New York, NY, USA), p 51–61.

[84] Hamburg M, Hermelink J, Primas R, Samardjiska S, Schamberger T, Streit S, Strieder E, van Vredendaal C (2021) Chosen ciphertext k-trace attacks on masked CCA2 secure Kyber. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(4):88–113. https://doi.org/10.46586/tches.v2021.i4.88-113

[85] Picek S, Perin G, Mariot L, Wu L, Batina L (2021) SoK: Deep learning-based physical side-channel analysis, Cryptology ePrint Archive, Report 2021/1092. https://ia.cr/2021/1092.

[86] Coron JS, Gérard F, Montoya S, Zeitoun R (2021) High-order table-based conversion algorithms and masking lattice-based encryption, Cryptology ePrint Archive, Report 2021/1314. https://ia.cr/2021/1314.

[87] Bettale L, Montoya S, Renault G (2021) Safe-error analysis of post-quantum cryptography mechanisms. *FDTC 2021 - Fault Diagnosis and Tolerance in Cryptographie* (Virtual event, France), pp 39–44.

[88] D'Anvers JP, Heinz D, Pessl P, van Beirendonck M, Verbauwhede I (2021) Higher-order masked ciphertext comparison for lattice-based cryptography, Cryptology ePrint Archive, Report 2021/1422. https://ia.cr/2021/1422.

[89] Coron JS, Gérard F, Montoya S, Zeitoun R (2021) High-order polynomial comparison and masking lattice-based encryption, Cryptology ePrint Archive, Report 2021/1615. https://ia.cr/2021/1615.

[90] Azouaoui M, Bronchain O, Hoffmann C, Kuzovkova Y, Schneider T, Standaert FX (2022) Systematic study of decryption and re-encryption leakage: the case of Kyber, Cryptology ePrint Archive, Report 2022/036. https://ia.cr/2022/036.

[91] D'Anvers JP, Van Beirendonck M, Verbauwhede I (2022) Revisiting higher-order masked comparison for lattice-based cryptography: Algorithms and bit-sliced implementations, Cryptology ePrint Archive, Report 2022/110. https://ia.cr/2022/110.

[92] Bronchain O, Cassiers G (2022) Bitslicing arithmetic/boolean masking conversions for fun and profit with application to lattice-based KEMs, Cryptology ePrint Archive, Report 2022/158. https://ia.cr/2022/158.

[93] Howe J, Prest T, Apon D (2021) SoK: How (not) to design and implement post-quantum cryptography. *Topics in Cryptology – CT-RSA 2021*, ed Paterson KG (Springer International Publishing, Cham), pp 444–477.

[94] Askeland A, Rønjom S (2021) A side-channel assisted attack on NTRU, Cryptology ePrint Archive, Report 2021/790. https://ia.cr/2021/790.

[95] Kamucheka T, Fahr M, Teague T, Nelson A, Andrews D, Huang M (2021) Power-based side channel attack analysis on PQC algorithms, Cryptology ePrint Archive, Report 2021/1021. https://ia.cr/2021/1021.

[96] Heinz D, Kannwischer MJ, Land G, Pöppelmann T, Schwabe P, Sprenkels D (2022) First-order masked Kyber on ARM Cortex-M4, Cryptology ePrint Archive, Report 2022/058. https://ia.cr/2022/058.

[97] Aranha DF, Berndt S, Eisenbarth T, Seker O, Takahashi A, Wilke L, Zaverucha G (2021) Side-channel protections for Picnic signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021(4):239–282. https://doi.org/10.46586/tches.v2021.i4.239-282

[98] Ravi P, Sinha Roy S, Chattopadhyay A, Bhasin S (2020) Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020(3):307–335. https://doi.org/10.13154/tches.v2020.i3.307-335

[99] Beirendonck MV, D'anvers JP, Karmakar A, Balasch J, Verbauwhede I (2021) A side-channel-resistant implementation of SABER. *Journal on Emerging Technolo-*

gies in Computing Systems 17(2). https://doi.org/10.1145/3429983

[100] Fritzmann T, Van Beirendonck M, Basu Roy D, Karl P, Schamberger T, Verbauwhede I, Sigl G (2021) Masked accelerators and instruction set extensions for post-quantum cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2022(1):414–460. https://doi.org/10.46586/tches.v2022.i1.414-460

[101] Kwiatkowski K, Sullivan N, Langley A, Levin D, Mislove A (2019) Measuring TLS key exchange with post-quantum KEM, Workshop Record of the Second PQC Standardization Conference. Available at https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/kwiatkowski-measuring-tls.pdf.

[102] Langley A (2018) CECPQ2. imperial violet, blog. Available at https://www.chromium.org/cecpq2/.

[103] Schwabe P, Stebila D, Wiggers T (2020) Post-quantum TLS without handshake signatures. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* CCS '20 (Association for Computing Machinery, New York, NY, USA), p 1461–1480. https://doi.org/10.1145/3372297.3423350

[104] Weibel A (2020) AWS security blog: Round 2 post-quantum TLS is now supported in AWS KMS. Available at https://aws.amazon.com/blogs/security/round-2-post-quantum-tls-is-now-supported-in-aws-kms/.

[105] Sikeridis D, Kampanakis P, Devetsikiotis M (2020) Post-quantum authentication in TLS 1.3: A performance study. *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020* (The Internet Society), pp –.

[106] Sikeridis D, Kampanakis P, Devetsikiotis M (2020) Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH. *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies* CoNEXT '20 (Association for Computing Machinery, New York, NY, USA), p 149–156. https://doi.org/10.1145/3386367.3431305

[107] Paul S, Kuzovkova Y, Lahr N, Niederhagen R (2021) Mixed certificate chains for the transition to post-quantum authentication in TLS 1.3, Cryptology ePrint Archive, Report 2021/1447. https://ia.cr/2021/1447.

[108] Sizing up post-quantum signatures. Available at https://blog.cloudflare.com/sizing-up-post-quantum-signatures/.

[109] Tao C, Petzoldt A, Ding J (2020) Improved key recovery of the HFEv- signature scheme, Cryptology ePrint Archive, Report 2020/1424. https://ia.cr/2020/1424.

[110] Florida Atlantic University (2020) PQC-Wiki. Available at https://pqc-wiki.fau.edu/w/Special:DatabaseHome.

[111] PQShield (2020) PQCzoo. Available at https://pqczoo.com.

[112] PQClean (2020). Available at https://github.com/PQClean/PQClean.

[113] Security innovation makes NTRUEncrypt patent-free, https://www.globenewswire.com/news-release/2017/03/28/945815/0/en/Security-Innovation-Makes-

NTRUEncrypt-Patent-Free.html. Accessed: 2022-01-25.

[114] Berlekamp E, McEliece R, van Tilborg H (1978) On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory* 24(3):384–386. https://doi.org/10.1109/TIT.1978.1055873

[115] Prange E (1962) The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory* 8(5):5–9. https://doi.org/10.1109/TIT.1962.1057777

[116] Stern J (1989) A method for finding codewords of small weight. *Coding Theory and Applications*, eds Cohen G, Wolfmann J (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 106–113.

[117] Dumer I (1991) On minimum distance decoding of linear codes. *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pp 50–52.

[118] Canteaut A, Chabaud F (1998) A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory* 44(1):367–378. https://doi.org/10.1109/18.651067

[119] Bernstein DJ, Lange T, Peters C (2008) Attacking and defending the McEliece cryptosystem. *Post-Quantum Cryptography*, eds Buchmann J, Ding J (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 31–46.

[120] Bernstein DJ, Lange T, Peters C, van Tilborg HCA (2009) Explicit bounds for generic decoding algorithms for code-based cryptography. *Pre-proceedings of WCC 2009*, pp 168–180.

[121] Finiasz M, Sendrier N (2009) Security bounds for the design of code-based cryptosystems. *Advances in Cryptology – ASIACRYPT 2009*, ed Matsui M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 88–105.

[122] May A, Meurer A, Thomae E (2011) Decoding random linear codes in $\tilde{O}(2^{0.054n})$. *Advances in Cryptology – ASIACRYPT 2011*, eds Lee DH, Wang X (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 107–124.

[123] Bernstein DJ, Lange T, Peters C (2011) Smaller decoding exponents: Ball-collision decoding. *Advances in Cryptology – CRYPTO 2011*, ed Rogaway P (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 743–760.

[124] Becker A, Joux A, May A, Meurer A (2012) Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. *Advances in Cryptology – EUROCRYPT 2012*, eds Pointcheval D, Johansson T (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 520–536.

[125] Hamdaoui Y, Sendrier N (2013) A non asymptotic analysis of information set decoding, Cryptology ePrint Archive, Report 2013/162. https://ia.cr/2013/162.

[126] May A, Ozerov I (2015) On computing nearest neighbors with applications to decoding of binary linear codes. *Advances in Cryptology – EUROCRYPT 2015*, eds Oswald E, Fischlin M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 203–228.

[127] Canto Torres R, Sendrier N (2016) Analysis of information set decoding for a sublinear error weight. *Post-Quantum Cryptography*, ed Takagi T (Springer Interna-

tional Publishing, Cham), pp 144–161.
[128] Both L, May A (2017) Optimizing BJMM with nearest neighbors: full decoding in $2^{2n/21}$ and McEliece security. *The Tenth International Workshop on Coding and Cryptography*, pp –.
[129] Both L, May A (2018) Decoding linear codes with high error rate and its impact for LPN security. *Post-Quantum Cryptography*, eds Lange T, Steinwandt R (Springer International Publishing, Cham), pp 25–46.
[130] Bernstein DJ (2010) Grover vs. McEliece. *Post-Quantum Cryptography*, ed Sendrier N (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 73–80.
[131] Kachigar G, Tillich JP (2017) Quantum information set decoding algorithms. *Post-Quantum Cryptography*, eds Lange T, Takagi T (Springer International Publishing, Cham), pp 69–89.
[132] Kirshanova E (2018) Improved quantum information set decoding. *Post-Quantum Cryptography*, eds Lange T, Steinwandt R (Springer International Publishing, Cham), pp 507–527.
[133] Esser A, Ramos-Calderer S, Bellini E, Latorre JI, Manzano M (2021) An optimized quantum implementation of ISD on scalable quantum resources, Cryptology ePrint Archive, Report 2021/1608. https://ia.cr/2021/1608.
[134] Baldi M, Barenghi A, Chiaraluce F, Pelosi G, Santini P (2019) A finite regime analysis of information set decoding algorithms. *Algorithms* 12(10). https://doi.org/10.3390/a12100209
[135] Esser A, Bellini E (2021) Syndrome decoding estimator, Cryptology ePrint Archive, Report 2021/1243. https://ia.cr/2021/1243.
[136] Esser A, May A, Zweydinger F (2021) McEliece needs a break – solving McEliece-1284 and quasi-cyclic-2918 with modern ISD, Cryptology ePrint Archive, Report 2021/1634. https://ia.cr/2021/1634.
[137] Sendrier N (2011) Decoding one out of many. *Post-Quantum Cryptography*, ed Yang BY (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 51–67.
[138] Patarin J, Goubin L (1997) Trapdoor one-way permutations and multivariate polynominals. *Proceedings of the First International Conference on Information and Communication Security* ICICS '97 (Springer-Verlag, Berlin, Heidelberg), p 356–368.
[139] Buss JF, Frandsen GS, Shallit JO (1996) The computational complexity of some problems of linear algebra. *BRICS Report Series* 3(33). https://doi.org/10.7146/brics.v3i33.20013
[140] Faugére JC (1999) A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra* 139(1):61–88. https://doi.org/https://doi.org/10.1016/S0022-4049(99)00005-5
[141] Faugère JC (2002) A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation* ISSAC '02 (Association for Computing Machinery, New York, NY, USA), p 75–83. https://doi.org/10.1145/780506.780516

[142] Courtois N, Klimov A, Patarin J, Shamir A (2000) Efficient algorithms for solving overdefined systems of multivariate polynomial equations. *Advances in Cryptology — EUROCRYPT 2000*, ed Preneel B (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 392–407.

[143] Goubin L, Courtois NT (2000) Cryptanalysis of the TTM cryptosystem. *Advances in Cryptology — ASIACRYPT 2000*, ed Okamoto T (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 44–57.

[144] Bardet M, Bros M, Cabarcas D, Gaborit P, Perlner R, Smith-Tone D, Tillich JP, Verbel J (2020) Improvements of algebraic attacks for solving the rank decoding and MinRank problems. *Advances in Cryptology – ASIACRYPT 2020*, eds Moriai S, Wang H (Springer International Publishing, Cham), pp 507–536.

[145] Ajtai M (1996) Generating hard instances of lattice problems (extended abstract). *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* STOC '96 (Association for Computing Machinery, New York, NY, USA), p 99–108. https://doi.org/10.1145/237814.237838

[146] Hoffstein J, Pipher J, Silverman JH (1998) NTRU: A ring-based public key cryptosystem. *Algorithmic Number Theory*, ed Buhler JP (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 267–288.

[147] Regev O (2005) On lattices, learning with errors, random linear codes, and cryptography. *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing* STOC '05 (Association for Computing Machinery, New York, NY, USA), p 84–93. https://doi.org/10.1145/1060590.1060603

[148] Peikert C (2016) A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science* 10(4):283–424. https://doi.org/10.1561/0400000074

[149] Micciancio D (2002) Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp 356–365. https://doi.org/10.1109/SFCS.2002.1181960

[150] Goldreich O, Levin LA (1989) A hard-core predicate for all one-way functions. *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pp 25–32.

[151] Stehlé D, Steinfeld R, Tanaka K, Xagawa K (2009) Efficient public key encryption based on ideal lattices. *International Conference on the Theory and Application of Cryptology and Information Security* (Springer), pp 617–635.

[152] Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. *Advances in Cryptology – EUROCRYPT 2010*, ed Gilbert H (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 1–23.

[153] Brakerski Z, Gentry C, Vaikuntanathan V (2012) (Leveled) fully homomorphic encryption without bootstrapping. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* ITCS '12 (Association for Computing Machinery, New York, NY, USA), p 309–325. https://doi.org/10.1145/2090236.2090262

[154] Jintai Ding XL Xiang Xie (2012) A simple provably secure key exchange scheme

based on the learning with errors problem, Cryptology ePrint Archive, Report 2012/688. https://ia.cr/2012/688.

[155] Peikert C (2014) Lattice cryptography for the internet. *Post-Quantum Cryptography*, ed Mosca M (Springer International Publishing, Cham), pp 197–219.

[156] Alkim E, Ducas L, Pöppelmann T, Schwabe P (2016) Post-quantum key exchange: A new hope. *Proceedings of the 25th USENIX Conference on Security Symposium* SEC'16 (USENIX Association, USA), p 327–343.

[157] Banerjee A, Peikert C, Rosen A (2012) Pseudorandom functions and lattices. *Advances in Cryptology – EUROCRYPT 2012*, eds Pointcheval D, Johansson T (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 719–737.

[158] Albrecht M, Bai S, Ducas L (2016) A subfield lattice attack on overstretched NTRU assumptions. *Advances in Cryptology – CRYPTO 2016*, eds Robshaw M, Katz J (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 153–178.

[159] Cheon JH, Jeong J, Lee C (2016) An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics* 19(A):255–266. https://doi.org/10.1112/S1461157016000371

[160] Kirchner P, Fouque PA (2017) Revisiting lattice attacks on overstretched NTRU parameters. *Advances in Cryptology – EUROCRYPT 2017*, eds Coron JS, Nielsen JB (Springer International Publishing, Cham), pp 3–26.

[161] Pellet-Mary A, Stehlé D (2021) On the hardness of the NTRU problem. *Advances in Cryptology – ASIACRYPT 2021*, eds Tibouchi M, Wang H (Springer International Publishing, Cham), pp 3–35.

[162] López-Alt A, Tromer E, Vaikuntanathan V (2012) On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing* STOC '12 (Association for Computing Machinery, New York, NY, USA), p 1219–1234. https://doi.org/10.1145/2213977.2214086

[163] Albrecht MR, Göpfert F, Virdia F, Wunderer T (2017) Revisiting the expected cost of solving uSVP and applications to LWE. *Advances in Cryptology – ASIACRYPT 2017*, eds Takagi T, Peyrin T (Springer International Publishing, Cham), pp 297–322.

[164] Albrecht MR (2017) On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. *Advances in Cryptology – EUROCRYPT 2017*, eds Coron JS, Nielsen JB (Springer International Publishing, Cham), pp 103–129.

[165] Guo Q, Johansson T (2021) Faster dual lattice attacks for solving LWE with applications to CRYSTALS. *Advances in Cryptology – ASIACRYPT 2021*, eds Tibouchi M, Wang H (Springer International Publishing, Cham), pp 33–62.

[166] The Center of Encryption and Information Security – MATZOV IDF (2022) Report on the security of LWE: Improved dual lattice attack. https://doi.org/10.5281/zenodo.6412487.

[167] Howgrave-Graham N (2007) A hybrid lattice-reduction and meet-in-the-middle

attack against NTRU. *Advances in Cryptology - CRYPTO 2007*, ed Menezes A (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 150–169.

[168] Nguyen PQ (2021) Boosting the hybrid attack on NTRU: Torus LSH, permuted HNF and boxed sphere, https://csrc.nist.gov/CSRC/media/Events/third-pqc-standardization-conference/documents/accepted-papers/nguyen-boosting-hybridboost-pqc2021.pdf. Third NIST PQC Standardization Conference.

[169] Katz J, Lindell Y (2020) *Introduction to Modern Cryptography* (Chapman & Hall/CRC), 3rd Ed.

[170] Fujisaki E, Okamoto T (1999) Secure integration of asymmetric and symmetric encryption schemes. *Advances in Cryptology — CRYPTO' 99*, ed Wiener M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 537–554.

[171] Fujisaki E, Okamoto T (2013) Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology* 26:80–101. https://doi.org/10.1007/s00145-011-9114-1

[172] Hofheinz D, Hövelmanns K, Kiltz E (2017) A modular analysis of the Fujisaki-Okamoto transformation. *Theory of Cryptography*, eds Kalai Y, Reyzin L (Springer International Publishing, Cham), pp 341–371.

[173] Cremers C, Düzlü S, Fiedler R, Janson C, Fischlin M (2021) BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. *2021 IEEE Symposium on Security and Privacy (SP)* (IEEE Computer Society, Los Alamitos, CA, USA), pp 1696–1714. https://doi.org/10.1109/SP40001.2021.00093

[174] Grubbs P, Maram V, Paterson KG (2021) Anonymous, robust post-quantum public key encryption, Cryptology ePrint Archive, Report 2021/708. https://ia.cr/2021/708.

[175] Bellare M, Rogaway P (1993) Random oracles are practical: A paradigm for designing efficient protocols. *Proceedings of the 1st ACM Conference on Computer and Communications Security* CCS '93 (Association for Computing Machinery, New York, NY, USA), p 62–73. https://doi.org/10.1145/168588.168596

[176] Canetti R, Goldreich O, Halevi S (2004) The random oracle methodology, revisited. *Journal of the ACM* 51(4):557–594. https://doi.org/10.1145/1008731.1008734

[177] Koblitz N, Menezes AJ (2015) The random oracle model: A twenty-year retrospective. *Designs, Codes, and Cryptography* 77(2):587–610. https://doi.org/10.1007/s10623-015-0094-2

[178] Boneh D, Dagdelen Ö, Fischlin M, Lehmann A, Schaffner C, Zhandry M (2011) Random oracles in a quantum world. *Advances in Cryptology – ASIACRYPT 2011*, eds Lee DH, Wang X (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 41–69.

[179] Grover LK (1996) A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* STOC '96 (Association for Computing Machinery, New York, NY, USA), p 212–219. https://doi.org/10.1145/237814.237866

[180] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, Schwabe P, Seiler G, Stehle D (2018) CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pp 353–367.

https://doi.org/10.1109/EuroSP.2018.00032

[181] Lindner R, Peikert C (2011) Better key sizes (and attacks) for LWE-based encryption. *Topics in Cryptology – CT-RSA 2011*, ed Kiayias A (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 319–339.

[182] Langlois A, Stehlé D (2015) Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography* 75(3):565–599. https://doi.org/10.1007/s10623-014-9938-4

[183] Mukherjee T, Stephens-Davidowitz N (2020) Lattice reduction for modules, or how to reduce moduleSVP to moduleSVP. *Advances in Cryptology – CRYPTO 2020*, eds Micciancio D, Ristenpart T (Springer International Publishing, Cham), pp 213–242.

[184] Peikert C, Pepin Z (2019) Algebraically structured LWE, revisited. *Theory of Cryptography*, eds Hofheinz D, Rosen A (Springer International Publishing, Cham), pp 1–23.

[185] Bolboceanu M, Brakerski Z, Sharma D (2021) On algebraic embedding for unstructured lattices, Cryptology ePrint Archive, Report 2021/053. https://ia.cr/2021/053.

[186] Saito T, Xagawa K, Yamakawa T (2018) Tightly-secure key-encapsulation mechanism in the quantum random oracle model. *Advances in Cryptology – EUROCRYPT 2018*, eds Nielsen JB, Rijmen V (Springer International Publishing, Cham), pp 520–551.

[187] Misoczki R, Tillich JP, Sendrier N, Barreto PSLM (2013) MDPC-McEliece: New McEliece variants from moderate density parity-check codes. *2013 IEEE International Symposium on Information Theory*, pp 2069–2073. https://doi.org/10.1109/ISIT.2013.6620590

[188] Drucker N, Gueron S, Kostic D (2019) QC-MDPC decoders with several shades of gray, Cryptology ePrint Archive, Report 2019/1423. https://ia.cr/2019/1423.

[189] Gligoroski D (2018) OFFICIAL COMMENT: BIKE, NIST PQC Forum. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/BIKE-official-comment.pdf.

[190] Guo Q, Johansson T, Stankovski P (2016) A key recovery attack on MDPC with CCA security using decoding errors. *Advances in Cryptology – ASIACRYPT 2016*, eds Cheon JH, Takagi T (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 789–815.

[191] Richter-Brockmann J, Mono J, Guneysu T (2021) Folding BIKE: Scalable hardware implementation for recongurable devices. *IEEE Transactions on Computers* https://doi.org/10.1109/TC.2021.3078294

[192] Reinders A, Misoczki R, Ghosh S, Sastry M (2020) Efficient BIKE hardware design with constant-time decoder, Cryptology ePrint Archive, Report 2020/117. https://ia.cr/2020/117.

[193] Richter-Brockmann J, Chen MS, Ghosh S, Güneysu T (2021) Racing BIKE: Improved polynomial multiplication and inversion in hardware, Cryptology ePrint Archive, Report 2021/1344. https://ia.cr/2021/1344.

[194] Hu J, Wang W, Cheung RCC, Wang H (2019) Optimized polynomial multiplier over commutative rings on FPGAs: A case study on BIKE. *2019 International Conference on Field-Programmable Technology (ICFPT)*, pp 231–234. https://doi.org/10.1109/ICFPT47387.2019.00035

[195] Vasseur V (2021) QC-MDPC codes DFR and the IND-CCA security of BIKE, Cryptology ePrint Archive, Report 2021/1458. https://ia.cr/2021/1458.

[196] Vasseur V (2021) *Post-quantum cryptography: a study of the decoding of QC-MDPC codes*. Ph.D. thesis. Université de Paris, Paris, France.

[197] McEliece RJ (1978) A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report* 44:114–116.

[198] Bindel N, Hamburg M, Hövelmanns K, Hülsing A, Persichetti E (2019) Tighter proofs of CCA security in the quantum random oracle model. *Theory of Cryptography*, eds Hofheinz D, Rosen A (Springer International Publishing, Cham), pp 61–90.

[199] Faugère JC, Gauthier-Umanã V, Otmani A, Perret L, Tillich JP (2011) A distinguisher for high rate McEliece cryptosystems. *2011 IEEE Information Theory Workshop*, pp 282–286. https://doi.org/10.1109/ITW.2011.6089437

[200] Sidelnikov VM, Shestakov SO (1992) On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications* 2(4):439–444. https://doi.org/doi:10.1515/dma.1992.2.4.439

[201] Wieschebrink C (2010) Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. *Post-Quantum Cryptography*, ed Sendrier N (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 61–72.

[202] Couvreur A, Gaborit P, Gauthier-Umaña V, Otmani A, Tillich JP (2014) Distinguisher-based attacks on public-key cryptosystems using Reed—Solomon codes. *Designs, Codes and Cryptography* 73(2):641–666. https://doi.org/10.1007/s10623-014-9967-z

[203] Minder L, Shokrollahi A (2007) Cryptanalysis of the Sidelnikov cryptosystem. *Advances in Cryptology - EUROCRYPT 2007*, ed Naor M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 347–360.

[204] Borodin MA, Chizhov IV (2014) Effective attack on the McEliece cryptosystem based on Reed-Muller codes. *Discrete Mathematics and Applications* 24(5):273–280. https://doi.org/10.1515/dma-2014-0024

[205] Faugère JC, Otmani A, Perret L, de Portzamparc F, Tillich JP (2014) Structural cryptanalysis of McEliece schemes with compact keys, Cryptology ePrint Archive, Report 2014/210. https://ia.cr/2014/210.

[206] Aguilar-Melchor C, Blazy O, Deneuville JC, Gaborit P, Zémor G (2018) Efficient encryption from random quasi-cyclic codes. *IEEE Transactions on Information Theory* 64(5):3927–3943. https://doi.org/10.1109/TIT.2018.2804444

[207] Aragon N, Gaborit P, Z'emor G (2020) HQC-RMRS, an instantiation of the HQC encryption framework with a more efficient auxiliary error-correcting code. *ArXiv* abs/2005.10741.

[208] Guo Q, Johansson T (2020) A new decryption failure attack against HQC. *Advances*

in Cryptology – ASIACRYPT 2020, eds Moriai S, Wang H (Springer International Publishing, Cham), pp 353–382.

[209] Schamberger T, Renner J, Sigl G, Wachter-Zeh A (2020) A power side-channel attack on the CCA2-secure HQC KEM, Cryptology ePrint Archive, Report 2020/910. https://ia.cr/2020/910.

[210] Hlauschek C, Lahr N, Schröder RL (2021) On the timing leakage of the deterministic re-encryption in HQC KEM, Cryptology ePrint Archive, Report 2021/1485. https://ia.cr/2021/1485.

[211] Jao D, De Feo L (2011) Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Post-Quantum Cryptography*, ed Yang BY (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 19–34.

[212] De Feo L, Jao D, Plût J (2014) Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology* 8(3):209–247.

[213] Childs A, Jao D, Soukharev V (2014) Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology* 8(1):1–29.

[214] Jaques S, Schanck JM (2019) Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. *Advances in Cryptology – CRYPTO 2019*, eds Boldyreva A, Micciancio D (Springer International Publishing, Cham), pp 32–61.

[215] Eisenträger K, Hallgren S, Lauter K, Morrison T, Petit C (2018) Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. *Advances in Cryptology – EUROCRYPT 2018*, eds Nielsen JB, Rijmen V (Springer International Publishing, Cham), pp 329–368.

[216] de Quehen V, Kutas P, Leonardi C, Martindale C, Panny L, Petit C, Stange KE (2021) Improved torsion-point attacks on SIDH variants. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 432–470.

[217] Costello C (2021) The case for SIKE: A decade of the supersingular isogeny problem, Cryptology ePrint Archive, Report 2021/543. https://ia.cr/2021/543.

[218] Kutas P, Merz SP, Petit C, Weitkämper C (2021) One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols. *Advances in Cryptology – EUROCRYPT 2021*, eds Canteaut A, Standaert FX (Springer International Publishing, Cham), pp 242–271.

[219] Gélin A, Wesolowski B (2017) Loop-abort faults on supersingular isogeny cryptosystems. *Post-Quantum Cryptography*, eds Lange T, Takagi T (Springer International Publishing, Cham), pp 93–106.

[220] Koziel B, Azarderakhsh R, Jao D (2018) An exposure model for supersingular isogeny Diffie-Hellman key exchange. *Topics in Cryptology – CT-RSA 2018*, ed Smart NP (Springer International Publishing, Cham), pp 452–469.

[221] Ti YB (2017) Fault attack on supersingular isogeny cryptosystems. *Post-Quantum Cryptography*, eds Lange T, Takagi T (Springer International Publishing, Cham), pp 107–122.

[222] Hankerson D, Menezes AJ, Vanstone S (2006) *Guide to elliptic curve cryptography*

(Springer Science & Business Media), .

[223] Brier É, Joye M (2002) Weierstraß elliptic curves and side-channel attacks. *Public Key Cryptography*, eds Naccache D, Paillier P (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 335–345.

[224] Elkhatib R, Azarderakhsh R, Mozaffari-Kermani M (2021) High-performance FPGA accelerator for SIKE. *IEEE Transactions on Computers* https://doi.org/10.1109/TC.2021.3078691

[225] Tian J, Wu B, Wang Z (2021) High-speed FPGA implementation of SIKE based on an ultra-low-latency modular multiplier. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68(9):3719–3731. https://doi.org/10.1109/TCSI.2021.3094889

[226] Anastasova M, Azarderakhsh R, Kermani MM (2021) Fast strategies for the implementation of SIKE round 3 on ARM Cortex-M4. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68(10):4129–4141. https://doi.org/10.1109/TCSI.2021.3096916

[227] Longa P, Wang W, Szefer J (2021) The cost to break SIKE: A comparative hardware-based analysis with AES and SHA-3. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 402–431.

[228] Bos J, Costello C, Ducas L, Mironov I, Naehrig M, Nikolaenko V, Raghunathan A, Stebila D (2016) Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* CCS '16 (Association for Computing Machinery, New York, NY, USA), p 1006–1018. https://doi.org/10.1145/2976749.2978425

[229] Guo Q, Johansson T, Nilsson A (2020) A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. *Advances in Cryptology – CRYPTO 2020*, eds Micciancio D, Ristenpart T (Springer International Publishing, Cham), pp 359–386.

[230] Hoffstein J, Pipher J, Silverman J (1996) NTRU: a new high speed public key cryptosystem. *presented at the rump session of Crypto 96* .

[231] Hülsing A, Rijneveld J, Schanck JM, Schwabe P (2017) NTRU-HRSS-KEM algorithm specifications and supporting documentation, Submission to the NIST's post-quantum cryptography standardization process.

[232] Chen C, Hoffstein J, Whyte W, Zhang Z (2017) NIST PQ submission: NTRUEncrypt a lattice based encryption algorithm, Submission to the NIST's post-quantum cryptography standardization process.

[233] Bernstein DJ, Chuengsatiansup C, Lange T, van Vredendaal C (2018) NTRU Prime: Reducing attack surface at low cost. *Selected Areas in Cryptography – SAC 2017*, eds Adams C, Camenisch J (Springer International Publishing, Cham), pp 235–260.

[234] Bernstein DJ, Brumley BB, Chen MS, Tuveri N (2022) OpenSSLNTRU: Faster post-quantum TLS key exchange. Available at https://www.usenix.org/conference/usenixsecurity22/presentation/bernstein.

[235] Peng BY, Marotzke A, Tsai MH, Yang BY, Chen HL (2021) Streamlined NTRU Prime on FPGA, Cryptology ePrint Archive, Report 2021/1444. https://ia.cr/2021/1

444.

[236] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehlé D (2018) CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018(1):238–268. https://doi.org/10.13154/tches.v2018.i1.238-268

[237] Kuchta V, Sakzad A, Stehlé D, Steinfeld R, Sun SF (2020) Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. *Advances in Cryptology – EUROCRYPT 2020*, eds Canteaut A, Ishai Y (Springer International Publishing, Cham), pp 703–728.

[238] Liu FH, Wang Z (2020) Rounding in the rings. *Advances in Cryptology – CRYPTO 2020*, eds Micciancio D, Ristenpart T (Springer International Publishing, Cham), pp 296–326.

[239] Lyubashevsky V (2009) Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. *Advances in Cryptology – ASIACRYPT 2009*, ed Matsui M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 598–616.

[240] Bai S, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehlé D (2020) CRYSTALS-Dilithium: Algorithm specifications and supporting documentation, Submission to the NIST's post-quantum cryptography standardization process.

[241] Kiltz E, Lyubashevsky V, Schaffner C (2018) A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. *Advances in Cryptology – EUROCRYPT 2018*, eds Nielsen JB, Rijmen V (Springer International Publishing, Cham), pp 552–586.

[242] Gentry C, Peikert C, Vaikuntanathan V (2008) Trapdoors for hard lattices and new cryptographic constructions. *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* STOC '08 (Association for Computing Machinery, New York, NY, USA), pp 197–206. https://doi.org/10.1145/1374376.1374407

[243] Stehlé D, Steinfeld R (2011) Making NTRU as secure as worst-case problems over ideal lattices. *Advances in Cryptology – EUROCRYPT 2011*, ed Paterson KG (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 27–47.

[244] Ducas L, Lyubashevsky V, Prest T (2014) Efficient identity-based encryption over NTRU lattices. *Advances in Cryptology – ASIACRYPT 2014*, eds Sarkar P, Iwata T (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 22–41.

[245] Ducas L, Prest T (2016) Fast Fourier orthogonalization. *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation* ISSAC '16 (Association for Computing Machinery, New York, NY, USA), p 191–198. https://doi.org/10.1145/2930889.2930923

[246] Andrysco M, Nötzli A, Brown F, Jhala R, Stefan D (2018) Towards verified, constant-time floating point operations. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* CCS '18 (Association for Computing Machinery, New York, NY, USA), p 1369–1382. https://doi.org/10.1145/3243734.3243766

[247] Guerreau M, Martinelli A, Ricosset T, Rossi M (2022) The hidden parallelepiped is

back again: Power analysis attacks on Falcon, Cryptology ePrint Archive, Report 2022/057. https://ia.cr/2022/057.

[248] Bernstein DJ, Hülsing A (2019) Decisional second-preimage resistance: When does SPR imply PRE? *Advances in Cryptology – ASIACRYPT 2019*, eds Galbraith SD, Moriai S (Springer International Publishing, Cham), pp 33–62.

[249] Bernstein DJ, Hülsing A, Kölbl S, Niederhagen R, Rijneveld J, Schwabe P (2019) The SPHINCS$^+$ signature framework. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* CCS '19 (Association for Computing Machinery, New York, NY, USA), p 2129–2146. https://doi.org/10.1145/3319535.3363229

[250] Hülsing A, Kudinov M (2022) Recovering the tight security proof of SPHINCS$^+$, Cryptology ePrint Archive, Report 2022/346. https://ia.cr/2022/346.

[251] Hülsing A (2022) Changes to SPHINCS+ specification to prevent multi-user attacks. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/F9ZUtWCij54.

[252] Stern M (2021) Re: Diversity of signature schemes. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2LEoSpskELs/m/LkUdQ5mKAwAJ.

[253] Hülsing A (2021) SPHINCS+ instantiation of message hash with SHA2. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/2LEoSpskELs/m/E9pwGmngBAAJ.

[254] Antonov S (2022) Round 3 official comment: SPHINCS+. Available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/FVItvyRea28/m/mGaRi5iZBwAJ.

[255] Cooper DA, Apon DC, Dang QH, Davidson MS, Dworkin MJ, Miller CA (2020) Recommendation for stateful hash-based signature schemes (U.S. Department of Commerce, Washington, D.C.), Special Publication 800-208. https://doi.org/10.6028/NIST.SP.800-208

[256] Bertoni G, Daemen J, Peeters M, Van Assche G (2008) On the indifferentiability of the sponge construction. *Advances in Cryptology – EUROCRYPT 2008*, ed Smart N (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 181–197.

[257] Patarin J (1996) Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. *Advances in Cryptology — EUROCRYPT '96*, ed Maurer U (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 33–48.

[258] Matsumoto T, Imai H (1988) Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. *Advances in Cryptology — EUROCRYPT '88*, eds Barstow D, Brauer W, Brinch Hansen P, Gries D, Luckham D, Moler C, Pnueli A, Seegmüller G, Stoer J, Wirth N, Günther CG (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 419–453.

[259] Patarin J (1995) Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. *Advances in Cryptology — CRYPT0' 95*, ed Coppersmith D (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 248–261.

[260] Patarin J, Courtois N, Goubin L (2001) QUARTZ, 128-bit long digital signatures. *Topics in Cryptology — CT-RSA 2001*, ed Naccache D (Springer Berlin Heidelberg,

Berlin, Heidelberg), pp 282–297.

[261] Bettale L, Faugère JC, Perret L (2013) Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Designs, Codes and Cryptography* 69(1):1–52. https://doi.org/10.1007/s10623-012-9617-2

[262] Vates J, Smith-Tone D (2017) Key recovery attack for all parameters of HFE-. *Post-Quantum Cryptography*, eds Lange T, Takagi T (Springer International Publishing, Cham), pp 272–288.

[263] Ding J, Perlner R, Petzoldt A, Smith-Tone D (2018) Improved cryptanalysis of HFEv- via projection. *Post-Quantum Cryptography*, eds Lange T, Steinwandt R (Springer International Publishing, Cham), pp 375–395.

[264] Baena J, Briaud P, Cabarcas D, Perlner R, Smith-Tone D, Verbel J (2021) Improving support-minors rank attacks: applications to G*e*MSS and Rainbow, Cryptology ePrint Archive, Report 2021/1677. https://ia.cr/2021/1677.

[265] Øygarden M, Smith-Tone D, Verbel J (2021) On the effect of projection on rank attacks in multivariate cryptography. *Post-Quantum Cryptography*, eds Cheon JH, Tillich JP (Springer International Publishing, Cham), pp 98–113.

[266] Boyar J, Peralta R, Pochuev D (2000) On the multiplicative complexity of Boolean functions over the basis ($\wedge$, $\oplus$, 1). *Theoretical Computer Science* 235(1):43–57. https://doi.org/10.1016/S0304-3975(99)00182-6

[267] Boyar J, Matthews P, Peralta R (2013) Logic minimization techniques with applications to cryptology. *Journal of Cryptology* 26(2):280–312. https://doi.org/10.1007/s00145-012-9124-7

[268] Albrecht MR, Rechberger C, Schneider T, Tiessen T, Zohner M (2015) Ciphers for MPC and FHE. *Advances in Cryptology – EUROCRYPT 2015*, eds Oswald E, Fischlin M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 430–454.

[269] Ishai Y, Kushilevitz E, Ostrovsky R, Sahai A (2007) Zero-knowledge from secure multiparty computation. *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing* STOC '07 (Association for Computing Machinery, New York, NY, USA), p 21–30. https://doi.org/10.1145/1250790.1250794

[270] Liu F, Isobe T, Meier W (2021) Cryptanalysis of full LowMC and LowMC-M with algebraic techniques. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 368–401.

[271] Banik S, Barooti K, Durak FB, Vaudenay S (2020) Cryptanalysis of LowMC instances using single plaintext/ciphertext pair. *IACR Transactions on Symmetric Cryptology* 2020(4):130–146. https://doi.org/10.46586/tosc.v2020.i4.130-146

[272] Dinur I (2021) Cryptanalytic applications of the polynomial method for solving multivariate equation systems over GF(2). *Advances in Cryptology – EUROCRYPT 2021*, eds Canteaut A, Standaert FX (Springer International Publishing, Cham), pp 374–403.

[273] Banik S, Barooti K, Vaudenay S, Yan H (2021) New attacks on LowMC instances with a single plaintext/ciphertext pair. *Advances in Cryptology – ASIACRYPT 2021*, eds Tibouchi M, Wang H (Springer International Publishing, Cham), pp 303–331.

[274] Liu F, Isobe T, Meier W (2021) A simple algebraic attack on 3-round LowMC, Cryptology ePrint Archive, Report 2021/255. https://ia.cr/2021/255.

[275] Liu F, Wang G, Meier W, Sarkar S, Isobe T (2022) Algebraic meet-in-the-middle attack on LowMC, Cryptology ePrint Archive, Report 2022/019. https://ia.cr/2022/019.

[276] Gellersen T, Seker O, Eisenbarth T (2021) Differential power analysis of the Picnic signature scheme. *Post-Quantum Cryptography*, eds Cheon JH, Tillich JP (Springer International Publishing, Cham), pp 177–194.

[277] de Saint Guilhem CD, De Meyer L, Orsini E, Smart NP (2020) BBQ: Using AES in Picnic signatures. *Selected Areas in Cryptography – SAC 2019*, eds Paterson KG, Stebila D (Springer International Publishing, Cham), pp 669–692.

[278] Baum C, de Saint Guilhem CD, Kales D, Orsini E, Scholl P, Zaverucha G (2021) Banquet: Short and fast signatures from AES. *Public-Key Cryptography – PKC 2021*, ed Garay JA (Springer International Publishing, Cham), pp 266–297.

[279] de Saint Guilhem CD, Orsini E, Tanguy T (2021) Limbo: Efficient zero-knowledge MPCitH-based arguments, Cryptology ePrint Archive, Report 2021/215. https://ia.cr/2021/215.

[280] Dobraunig C, Kales D, Rechberger C, Schofnegger M, Zaverucha G (2021) Shorter signatures based on tailor-made minimalist symmetric-key crypto, Cryptology ePrint Archive, Report 2021/692. https://ia.cr/2021/692.

[281] Sakumoto K, Shirai T, Hiwatari H (2011) On provable security of UOV and HFE signature schemes against chosen-message attack. *Post-Quantum Cryptography*, ed Yang BY (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 68–82.

[282] Kipnis A, Patarin J, Goubin L (1999) Unbalanced oil and vinegar signature schemes. *Advances in Cryptology — EUROCRYPT '99*, ed Stern J (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 206–222.

[283] Ding J, Schmidt D (2005) Rainbow, a new multivariable polynomial signature scheme. *Applied Cryptography and Network Security*, eds Ioannidis J, Keromytis A, Yung M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 164–175.

[284] Perlner R, Smith-Tone D (2020) Rainbow band separation is better than we thought, Cryptology ePrint Archive, Report 2020/702. https://ia.cr/2020/702.

[285] Laurie B, Messeri E, Stradling R (2021) Certificate Transparency version 2.0, Internet Engineering Task Force (IETF) request for comments (RFC) 9162, https://doi.org/10.17487/RFC9162.

[286] National Cybersecurity Center of Excellence (2021) Migration to post-quantum cryptography. Available at https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms.

[287] National Institute of Standards and Technology (2021) National Cybersecurity Center of Excellence (NCCoE) migration to post-quantum cryptography. *Federal Register* 86(195):56898–56900. https://www.federalregister.gov/d/2021-22223.

[288] Brent RP, Kung HT (1981) The area-time complexity of binary multiplication. *Journal of the ACM* 28(3):521–534. https://doi.org/10.1145/322261.322269

[289] Bouillaguet C (2022) Nice attacks — but what is the cost? computational models for cryptanalysis, Cryptology ePrint Archive, Report 2022/197. https://ia.cr/2022/197.
[290] Wiener MJ (2004) The full cost of cryptanalytic attacks. *Journal of Cryptology* 17:105–124. https://doi.org/10.1007/s00145-003-0213-5
[291] van Oorschot PC, Wiener MJ (1994) Parallel collision search with application to hash functions and discrete logarithms. *Proceedings of the 2nd ACM Conference on Computer and Communications Security* CCS '94 (Association for Computing Machinery, New York, NY, USA), p 210–218. https://doi.org/10.1145/191177.191231
[292] van Oorschot PC, Wiener MJ (1999) Parallel collision search with cryptanalytic applications. *Journal of Cryptology* 12:1–28.
[293] Beals R, Brierley S, Gray O, Harrow AW, Kutin S, Linden N, Shepherd D, Stather M (2013) Efficient distributed quantum computing. *Proceedings of the Royal Society A* 469(2153):20120686. https://doi.org/10.1098/rspa.2012.0686
[294] Giovannetti V, Lloyd S, Maccone L (2008) Quantum random access memory. *Physical Review Letters* 100(16). https://doi.org/10.1103/physrevlett.100.160501
[295] Arunachalam S, Gheorghiu V, Jochym-O'Connor T, Mosca M, Srinivasan PV (2015) On the robustness of bucket brigade quantum RAM. *New Journal of Physics* 17(12):123010. https://doi.org/10.1088/1367-2630/17/12/123010
[296] Hann CT, Lee G, Girvin S, Jiang L (2021) Resilience of quantum random access memory to generic noise. *PRX Quantum* 2:020311. https://doi.org/10.1103/PRXQuantum.2.020311
[297] Schnorr CP, Euchner M (1994) Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming* 66(1):181–199. https://doi.org/10.1007/BF01581144
[298] Pohst M (1981) On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bulletin* 15(1):37–44. https://doi.org/10.1145/1089242.1089247
[299] Gama N, Nguyen PQ, Regev O (2010) Lattice enumeration using extreme pruning. *Advances in Cryptology – EUROCRYPT 2010*, ed Gilbert H (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 257–278.
[300] Aono Y, Nguyen PQ, Seito T, Shikata J (2018) Lower bounds on lattice enumeration with extreme pruning. *Advances in Cryptology – CRYPTO 2018*, eds Shacham H, Boldyreva A (Springer International Publishing, Cham), pp 608–637.
[301] Albrecht MR, Bai S, Fouque PA, Kirchner P, Stehlé D, Wen W (2020) Faster enumeration-based lattice reduction: Root Hermite factor $k^{1/(2k)}$ in time $k^{k/8+o(k)}$, Cryptology ePrint Archive, Report 2020/707. https://ia.cr/2020/707.
[302] Albrecht MR, Bai S, Li J, Rowell J (2021) Lattice reduction with approximate enumeration oracles. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 732–759.
[303] Ajtai M, Kumar R, Sivakumar D (2001) A sieve algorithm for the shortest lattice vector problem. *Proceedings of the Thirty-Third Annual ACM Symposium on Theory*

of Computing STOC '01 (Association for Computing Machinery, New York, NY, USA), p 601–610. https://doi.org/10.1145/380752.380857

[304] Micciancio D, Voulgaris P (2010) Faster exponential time algorithms for the shortest vector problem. *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms* SODA '10 (Society for Industrial and Applied Mathematics, USA), p 1468–1480.

[305] Albrecht MR, Ducas L, Herold G, Kirshanova E, Postlethwaite EW, Stevens M (2019) The general sieve kernel and new records in lattice reduction. *Advances in Cryptology – EUROCRYPT 2019*, eds Ishai Y, Rijmen V (Springer International Publishing, Cham), pp 717–746.

[306] Nguyen PQ, Vidick T (2008) Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology* 2(2):181–207. https://doi.org/doi:10.1515/JMC.2008.009

[307] Laarhoven T (2015) Sieving for shortest vectors in lattices using angular locality-sensitive hashing. *Advances in Cryptology – CRYPTO 2015*, eds Gennaro R, Robshaw M (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 3–22.

[308] Anja Becker AJ Nicolas Gama (2015) Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search, Cryptology ePrint Archive, Report 2015/522. https://ia.cr/2015/522.

[309] Laarhoven T, de Weger B (2015) Faster sieving for shortest lattice vectors using spherical locality-sensitive hashing. *Progress in Cryptology – LATINCRYPT 2015*, eds Lauter K, Rodríguez-Henríquez F (Springer International Publishing, Cham), pp 101–118.

[310] Becker A, Ducas L, Gama N, Laarhoven T (2016) New directions in nearest neighbor searching with applications to lattice sieving. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms* SODA '16 (Society for Industrial and Applied Mathematics, USA), p 10–24.

[311] Ducas L (2018) Shortest vector from lattice sieving: A few dimensions for free. *Advances in Cryptology – EUROCRYPT 2018*, eds Nielsen JB, Rijmen V (Springer International Publishing, Cham), pp 125–145.

[312] Laarhoven T, Mariano A (2018) Progressive lattice sieving. *Post-Quantum Cryptography*, eds Lange T, Steinwandt R (Springer International Publishing, Cham), pp 292–311.

[313] Doulgerakis E, Laarhoven T, de Weger B (2020) Sieve, enumerate, slice, and lift:. *Progress in Cryptology - AFRICACRYPT 2020*, eds Nitaj A, Youssef A (Springer International Publishing, Cham), pp 301–320.

[314] Zhao Z, Ding J (2022) Several improvements on BKZ algorithm, Cryptology ePrint Archive, Report 2022/239. https://ia.cr/2022/239.

[315] Kirshanova E, Laarhoven T (2021) Lower bounds on lattice sieving and information set decoding. *Advances in Cryptology – CRYPTO 2021*, eds Malkin T, Peikert C (Springer International Publishing, Cham), pp 791–820.

[316] Hanrot G, Pujol X, Stehlé D (2011) Terminating BKZ, Cryptology ePrint Archive,

Report 2011/198. https://ia.cr/2011/198.
[317] Chen Y, Nguyen PQ (2011) BKZ 2.0: Better lattice security estimates. *Advances in Cryptology – ASIACRYPT 2011*, eds Lee DH, Wang X (Springer Berlin Heidelberg, Berlin, Heidelberg), pp 1–20.
[318] Yu Y, Ducas L (2018) Second order statistical behavior of LLL and BKZ. *Selected Areas in Cryptography – SAC 2017*, eds Adams C, Camenisch J (Springer International Publishing, Cham), pp 3–22.
[319] Li J, Nguyen PQ (2020) A complete analysis of the BKZ lattice reduction algorithm, Cryptology ePrint Archive, Report 2020/1237. https://ia.cr/2020/1237.
[320] Postlethwaite EW, Virdia F (2021) On the success probability of solving unique SVP via BKZ. *Public-Key Cryptography – PKC 2021*, ed Garay JA (Springer International Publishing, Cham), pp 68–98.
[321] Albrecht MR, Curtis BR, Deo A, Davidson A, Player R, Postlethwaite EW, Virdia F, Wunderer T (2018) Estimate all the {LWE, NTRU} schemes! *Security and Cryptography for Networks*, eds Catalano D, De Prisco R (Springer International Publishing, Cham), pp 351–367.
[322] Cramer R, Ducas L, Wesolowski B (2017) Short Stickelberger class relations and application to ideal-SVP. *Advances in Cryptology – EUROCRYPT 2017*, eds Coron JS, Nielsen JB (Springer International Publishing, Cham), pp 324–348.
[323] Ducas L, Plançon M, Wesolowski B (2019) On the shortness of vectors to be found by the ideal-SVP quantum algorithm. *Advances in Cryptology – CRYPTO 2019*, eds Boldyreva A, Micciancio D (Springer International Publishing, Cham), pp 322–351.
[324] Bernstein D (2021) S-unit attacks. SIAM Conference on Applied Algebraic Geometry (plenary talk, Aug. 20, 2021).
[325] Bernstein DJ, Lange T (2021) Non-randomness of S-unit lattices, Cryptology ePrint Archive, Report 2021/1428. https://ia.cr/2021/1428.
[326] Bernard O, Lesavourey A, Nguyen TH, Roux-Langlois A (2021) Log-S-unit lattices using explicit Stickelberger generators to solve approx ideal-SVP, Cryptology ePrint Archive, Report 2021/1384. https://ia.cr/2021/1384.
[327] Bernard O, Roux-Langlois A (2020) Twisted-PHS: Using the product formula to solve approx-SVP in ideal lattices. *Advances in Cryptology – ASIACRYPT 2020*, eds Moriai S, Wang H (Springer International Publishing, Cham), pp 349–380.
[328] Pellet-Mary A, Hanrot G, Stehlé D (2019) Approx-SVP in ideal lattices with preprocessing. *Advances in Cryptology – EUROCRYPT 2019*, eds Ishai Y, Rijmen V (Springer International Publishing, Cham), pp 685–716.

A. Acronyms

Acronyms

AES	Advanced Encryption Standard

BKZ	Block Korkine-Zolotarev algorithm
CCA	Chosen Ciphertext Attack
CPA	Chosen Plaintext Attack
DNSSEC	Domain Name System Security Extensions
EUF-CMA	Existential Unforgeability under Chosen-Message Attack
FIPS	Federal Information Processing Standard
FPGA	Field Programmable Gate Array
gapSVP	Gap Shortest Vector Problem
HFE	Hidden Field Equations
IKE	Internet Key Exchange
IND-CCA	Indistinguishability under Chosen-Ciphertext Attack
IND-CCA2	Indistinguishability under Adaptive Chosen-Ciphertext Attack
IND-CPA	Indistinguishability under Chosen-Plaintext Attack
IPsec	Internet Protocol Security
KEM	Key-Encapsulation Mechanism
KiB	Kibi Byte, Measuring Unit 2^{10} Bytes = 1024 Bytes
LWE	Learning With Errors
LWR	Learning With Rounding
MLWE	Module Learning With Errors
MLWR	Module Learning With Rounding
MSIS	Module Short Integer Solution
NIST	National Institute of Standards and Technology
NISTIR	NIST Interagency or Internal Report
NTT	Number Theoretic Transform
OW-CPA	One-way under Chosen-Plaintext Attack

PKE	Public-Key Encryption
PQC	Post-Quantum Cryptography
QC-MDPC	Quasi-Cyclic Moderate Density Parity Check
QCCF	Quasi-cyclic Codeword Finding
QCSD	Quasi-cyclic Syndrome Decoding
QROM	Quantum-accessible Random Oracle Model
RAM	Random Access Memory or Random Access Machine
RLWR	Ring Learning With Rounding
ROM	Random Oracle Model
SHA	Secure Hash Algorithm
SHAKE	Secure Hash Algorithm KECCAK
SIDH	Supersingular Isogeny Diffie-Hellman
SIS	Short Integer Solution
SIVP	Shortest Independent Vector Problem
SP	Special Publication
SSH	Secure Shell
SUF-CMA	Strong Existential Unforgeability under Chosen-Message Attack
SVP	Shortest Vector Problem
TLS	Transport Layer Security
UOV	Unbalanced Oil and Vinegar cryptosystem

B. Cost Models

The RAM model. The most common cost model is the *Random Access Machine (RAM)* model. In this model, the cost of an attack is determined by counting operations that act on a fixed number of bits, including reading or writing to memory. The cost of memory access is assumed not to depend on the size of the memory, even when the memory is read or written in a random access fashion (i.e., a fashion where the memory address is not predictable). In the context of the NIST PQC Standardization Process, the version of the RAM model, where the operations being counted are "bit operations" that act on no more than 2 bits at a time and where each one-bit memory read or write is counted as one bit-operation, is sometimes referred to as the *gate count* model. This approach simplifies the cost analysis of a particular attack, as the metric does not require analyzing how memory is arranged in a physical computing system and how the distance between memory access points affects real-world costs like energy consumption and latency.

Since it treats the cost of memory access to a large memory as no more expensive than memory access to a small memory, a cost estimate in the RAM model will generally underestimate the cost of attacks that require random access to a large memory. Parameters that appear to meet their targeted security level when analyzed in the RAM model should therefore be considered safe barring new cryptanalysis. However, since it is likely that the cost of randomly accessing a memory will increase with its size in any physically realizable memory architecture, it may be possible to argue that more aggressive (and presumably better performing) parameter sets can meet their targeted security levels, even when RAM model analysis suggests that the best attack on the parameter set is cheaper than the attack (brute force key search or collision search) used to define the minimum attack cost for the security strength category.

Local models. In a *local model*, the distance between memory access points is considered in the cost analysis. Moving information from an initial point to a destination point requires some amount of energy and time. This cost could in fact be quite large for attacks where a significant fraction of the computations involve randomly accessing a large memory.

2D nearest neighbor models are local models that assume that memory is arranged in a two-dimensional fashion. One of the first studies on efficient layouts of gates to reduce costs by finding tradeoffs between area (of a chip, memory board, etc.) and time was published in 1981 [288]. This model is commonly referred to as the *time* $\times$ *area* model. Another type of 2D-nearest neighbor model, referred to by [289] as the "Expensive Memory Model," attempts to estimate the energy cost of accessing memory under the assumption that the cost of each random memory access is proportional to the distance that a bit must travel to or from the location where it is read or written. In this type of model, the cost of reading or writing a bit in a memory of size n would be equivalent to $\mathcal{O}(\sqrt{n})$ bit operations.

An estimate for the value of the hidden constant can be found in [60, Section 6.6]. It should be noted that this estimate is based on the density of Dynamic Random Access Memory (DRAM) and the per-distance energy cost of moving data via electric signals

through on-die wires and so may not be accurate where other technologies can be used. For large memories, perhaps several petabytes, it would likely make sense to transmit data via fiber optics, which incur some additional costs at their endpoints but consume significantly less energy per bandwidth per unit distance. For even larger memories, it would likely make sense to use a memory technology that is denser and cheaper to manufacture but slower and more expensive to read/write than DRAM. Examples of such memory technologies are hard disk and flash memory.

3D nearest neighbor models are local models that consider memory boards connected in a three-dimensional arrangement [290]. Such an arrangement increases the number of "near neighbors" to any particular bit of memory relative to the 2D model, reducing the cost to $\mathcal{O}(\sqrt[3]{n})$. It has been noted that heat cannot easily or quickly dissipate in a stacked memory board structure, so possible time delays to allow for cooling could potentially cancel out the cost reduction from the 2D to 3D model, depending on how often the memory needs to be accessed and whether the memory technology generates heat when it is idle.

Other cost models *Fixed Budget* models measure an attack by limiting the attacker to a fixed budget, and estimating the time required to execute the attack given the budget. Van Orschoot and Weiner included budget-based examples in their analyses of parallel collision search [291, 292] and in 2020, Longa et al. used a budget-based cost model to consider attacks on SIKE [227].

This approach arguably gives a realistic way of determining risk, when an attack can be executed within the considered budget on a timescale of no more than a few years. However, very long time estimates are likely not meaningful, because an attacker may do better by waiting for technology to improve before beginning to implement the attack. Very long time estimates may also be a sign that a larger budget should be considered when evaluating the attack. Moreover, the small budget considered in long-time-scale attacks is often illusory, in particular when evaluated costs do not include power consumption and hardware replacement costs, which surely would dominate over any initial investment in hardware for any attack lasting more than a few years. If the considered budget is restricted to the point where a very long time scale is required, the net effect will be to overestimate the cost of attacks that require a lot of memory, whether that memory is accessed in a random access or a local fashion.

The quantum circuit model. The most common model used for giving concrete quantum resource estimates is the quantum circuit model. In this model, a computation is described by a series of transformations (typically unitary gates) acting on some number of qubits. Each gate acts on at most two qubits at a time with no locality restriction. This is similar to the classical circuit model, where Boolean logic gates act on classical bits. While circuit models are non-local models in their most basic form, they differ significantly from RAM models in that an operation equivalent to a serial random access to a memory of size M would generally require M gates. As demonstrated by [293] this cost can be amortized if multiple processes are accessing the memory in parallel. As with the RAM model, a local

version of the circuit model may be considered, where qubits are assumed to be arranged in a two-dimensional or three-dimensional grid, and gates can only be performed between nearby qubits.

The resource costs of quantum algorithms are often assessed at the *logical* level (i.e., under the assumption that qubits and gates are essentially perfect). Alternatively, one can choose to assess resource costs at the *physical* level, i.e., considering the costs of constructing near-perfect qubits and gates from their real, imperfect analogues. An intermediate option is to assess costs at the logical level, but to count Clifford gates (which are typically cheap in quantum fault-tolerance schemes) differently from T gates (which are typically much more expensive.)

The NIST PQC call for proposals [9] highlighted a variant of the quantum circuit model where the adversary is limited to performing no more than MAXDEPTH gates in series. This is particularly relevant when making comparisons to quantum attacks on AES and SHA, which are known to not parallelize well.

The Quantum RAM model. The *quantum RAM model* [294] generalizes the classical RAM model to quantum computation and is used fairly often in giving asymptotic costs for quantum attacks. A logarithmic cost for RAM queries was proposed by [294] based on an idealized "Bucket-Brigade" architecture. However, it was argued by [295] that for large quantum computations, such as those needed in cryptanalysis, a Bucket-Brigade memory would require active error correction, yielding a similar cost for RAM access as predicted by the quantum circuit model. More recent analysis [296] has suggested that the Bucket Brigade architecture has an advantage over other architectures for emulating quantum RAM even where error correction is required, although it does not contradict the claim of [295] that a quantum memory that needs to be accessed a large number of times will require a number of active gates comparable to the size of the memory.

C. On the Concrete Intractability of Finding Short Lattice Vectors

The standard method for finding short lattice vectors, the BKZ (Block Korkine-Zolotarev) algorithm [297], was developed by Schnorr and Euchner in 1991. The BKZ algorithm solves the γ-approximate SVP problem[16] for lattices with large dimension, d, by iteratively calling an "oracle" for solving the SVP problem in sub-lattices of smaller dimension, β – gradually improving the "quality" of the lattice basis by finding vectors that are shorter and more nearly orthogonal to each other.

There are two types of lattice reduction algorithms that may be used to implement the SVP "oracle," enumeration and sieving. Enumeration algorithms (see, for example, [298–302]) require small amounts of memory but have run times that are super-exponential in β. Sieving algorithms (see, for example [303–305]) have run times that are exponential in β,

[16]The γ-approximate SVP problem involves finding a vector that is at most γ times longer than the shortest vector. This is a search problem and is closely related to the decision problem GapSVP.

but also require an exponential amount of memory. While enumeration algorithms outperform sieving algorithms for smaller dimensions, sieving performs better as the dimension increases. At least in the case of classical implementations of these algorithms, sieving performs better at dimensions that are used for cryptography [300]. While it is possible that quantum implementations of enumeration algorithms may affect whether it is possible for certain lattice parameter sets to meet category 2 or 4, the known quantum speedups are small enough that parameter sets with enough classical security to meet categories 1, 3, and 5 should also meet these targets when quantum attacks are considered. The performance of sieving algorithms has been improving [306–314], however recent results [315] indicate that improvements in locally sensitive hash techniques, which have resulted in the largest decreases in asymptotic complexity for sieving thus far, cannot be improved further.

BKZ and the sieving algorithms used to implement the SVP "oracle" are heuristic algorithms, and so estimating the cost of running BKZ can be tricky for various reasons. This is particularly the case since the behavior of these algorithms for dimensions of sizes relevant to cryptography is different from theoretical upper and lower bounds, and is also different from their behavior for small problem instances that are computationally tractable. Estimating the concrete hardness of sieving algorithms also requires accounting for the large amount of memory required (see Appendix B).

One commonly-used approach for estimating the cost of BKZ involves determining the *core SVP hardness* [156, Section 6.1] of the problem. While solving BKZ requires a polynomial number of calls to the "oracle" [316], determining the exact number of calls required can be difficult. The core SVP method avoids this complication by only estimating the cost of a single call to the SVP "oracle" in dimension β, after estimating the value of β that is required to find a solution that is useful for cryptanalysis. While comparing the core SVP hardness of various cryptosystems can be useful in comparing their relative security levels (see Tables 10 and 11), this approach necessarily underestimates the cost of running the BKZ algorithm.

Over the past decade, work has progressed in understanding the behavior of the BKZ algorithm on lattices of dimensions that are used in cryptosystems [317–320], and tools are available to aid in determining the concrete security of lattice-based cryptosystems [321].[17] As a result, understanding of the concrete security of lattice-based cryptosystems has greatly improved over the past several years.

Many lattice-based cryptosystems are based on problems that have algebraic structure, such as Ring-LWE or Module-LWE. These problems are connected to variants of the shortest vector problem that involve lattices with algebraic structure, such as ideal and module lattices, and the ideal and module SVP problems. The BKZ algorithm does not exploit the structure that is present in ideal or module lattices. However, for the purposes of practical cryptanalysis, there is no available evidence to suggest that algorithms that outperform BKZ on such lattices exist.

Nonetheless, from a theoretical perspective, it is important to note that there is a quan-

[17] The third round submission for KYBER provides an example of an extensive analysis of the concrete security of a lattice-based cryptosystem. [14, Section 5]

tum algorithm that runs in polynomial time and solves γ-approximate ideal SVP with an approximation ratio γ that is mildly subexponential [322]. This approximation ratio is asymptotically better than the approximation ratio achieved by BKZ with any fixed block size; but it appears to be worse than BKZ when the lattice dimension and block size are in the typical range for attacks on practical lattice-based cryptosystems [323]. Hence, the existence of this quantum algorithm does not appear to impact the practical security of lattice-based cryptosystems. In addition, the techniques used in this algorithm rely heavily on the multiplicative structure of ideal lattices, and do not seem to be directly applicable to module lattices of rank 2 or more. Hence these techniques are not known to directly impact the hardness assumptions of any of the round 3 candidates (e.g., Ring LWE, NTRU, Module LWE), even asymptotically.[18]

One area of recent interest is S-unit attacks [324, 325] and the "Twisted-PHS" algorithm [326–328], which can be viewed as generalizations of the quantum algorithm for γ-approximate ideal SVP. These algorithms use a computationally-intensive pre-processing step to improve the quality of the solution that is found. Some evidence suggests that these algorithms may achieve mild improvements in the approximation ratio γ, but there is little evidence that this will change the asymptotic scaling of γ from subexponential to polynomial (as a function of the dimension n). Hence these improvements are of theoretical interest, but seem unlikely to lead to practical attacks on lattice-based cryptosystems.

[18] In particular, the primal and dual attacks on the Ring LWE problem (see Section 3.2.3) require solving approximate SVP on a module lattice of rank 2 or 3, rather than an ideal lattice.

D. Figures and Tables

Candidate	Claimed Security	Public key	Private key	Ciphertext
Classic McEliece348864	Level 1	261 120	6 492	128
Classic McEliece460896	Level 3	524 160	13 608	188
Classic McEliece6688128	Level 5	104 992	13 932	240
Classic McEliece6960119	Level 5	1 047 319	13 948	226
Classic McEliece8192128	Level 5	1 357 824	14 120	240
KYBER512	Level 1	800	1 632	768
KYBER768	Level 3	1 184	2 400	1 088
KYBER1024	Level 5	1 568	3 168	1 568
NTRU-HPS2048677	Level 1	930	1 234	930
NTRU-HRSS701	Level 1	1 138	1 450	1 138
NTRU-HPS4096821	Level 3	1 230	1 590	1 230
NTRU-HPS40961229	Level 5	1 842	2 366	1 842
NTRU-HRSS1373	Level 5	2 401	2 983	2 401
Light Saber	Level 1	672	832	736
Saber	Level 3	992	1 248	1 088
Fire Saber	Level 5	1 312	1 664	1 472

Table 6. Key and ciphertext sizes (in bytes) for the KEM finalists

Candidate	Claimed Security	Public key	Private key	Ciphertext
FrodoKEM-640	Level 1	9 616	19 888	9 720
FrodoKEM-976	Level 3	15 632	31 296	15 744
FrodoKEM-1344	Level 5	21 520	43 088	21 632
BIKE	Level 1	1 540	280	1 572
	Level 3	3 082	418	3 114
	Level 5	5 122	580	5 154
HQC-128	Level 1	2 249	40	4 481
HQC-192	Level 3	4 522	40	9 026
HQC-256	Level 5	7 245	40	14 469
SIKEp434	Level 1	330	374	346
SIKEp503	Level 2	378	434	402
SIKEp610	Level 3	462	524	486
SIKEp751	Level 5	564	644	596
(NTRU Prime)				
sntrup653	Level 1	994	15 158	897
sntrup761	Level 2	1 158	1 763	1 039
sntrup857	Level 2/3	1 322	1 999	1 184
sntrup953	Level 3/4	1 505	2 254	1 349
sntrup1013	Level 4	1 623	2 417	1 455
sntrup1277	Level 5	2 067	3 059	1 847
ntrulpr653	Level 1	897	1 125	1 025
ntrulpr761	Level 2	1 039	1 294	1 167
ntrulpr857	Level 2/3	1 184	1 463	1 312
ntrulpr953	Level 3/4	1 349	1 652	1 477
ntrulpr1013	Level 4	1 455	1 773	1 583
ntrulpr1277	Level 5	1 847	2 231	1 975

Table 7. Key and ciphertext sizes (in bytes) for the KEM alternates. Some parameter sets for NTRU Prime claim two different security levels, depending on their interpretation of NIST's security requirements.

Candidate	Claimed Security	Public key	Private key	Signature
Dilithium	Level 2	1 312	2 528	2 420
	Level 3	1 952	4 000	3 293
	Level 5	2 592	4 864	4 595
FALCON-512	Level 1	897	7 553	666
FALCON-1024	Level 5	1 793	13 953	1 280
Rainbow I	Level 1&2	161 600	103 616	66
Rainbow III	Level 3&4	882 080	626 016	164
Rainbow V	Level 5	1 930 600	1 408 704	212

Table 8. Key and signature sizes (in bytes) for the signature finalists. Some Rainbow parameter sets each claim two security levels.

Candidate	Claimed Security	Public key	Private key	Signature
G*e*MSS128	Level 1	352 168	16	33
G*e*MSS192	Level 3	1 237 934	24	52
G*e*MSS256	Level 5	3 040 659	32	72
Picnic-L1-full	Level 1	34	17	30 809
Picnic3-L1	Level 1	34	17	12 359
Picnic-L3-full	Level 3	48	24	68 493
Picnic3-L3	Level 3	48	24	27 173
Picnic-L5-full	Level 5	64	32	121 616
Picnic3-L5	Level 5	64	32	46 282
SPHINCS$^+$-128s	Level 1	32	64	7856
SPHINCS$^+$-128f	Level 1	32	64	17 088
SPHINCS$^+$-192s	Level 3	48	96	16 224
SPHINCS$^+$-192f	Level 3	48	96	35 664
SPHINCS$^+$-256s	Level 5	64	128	29 792
SPHINCS$^+$-256f	Level 5	64	128	49 856

Table 9. Key and signature sizes (in bytes) for the signature alternates. Some parameter sets for Picnic have variable signature sizes. The Picnic signature sizes given in the table are the empirical averages of 100 samples for each parameter set.

Candidate	Claimed Security	core SVP Estimate	Gate Count	Memory
KYBER512	Level 1	C:118 bits Q:107 bits	2^{151}	2^{94}
KYBER768	Level 3	C:183 bits Q:166 bits	2^{215}	2^{139}
KYBER1024	Level 5	C:256 bits Q:232 bits	2^{287}	2^{190}
NTRU hps2048677	Level 1	C:144 bits	2^{176}	2^{111}
NTRU hrss701	Level 1	C:134 bits	2^{168}	2^{105}
NTRU hps4096821	Level 3	C:178 bits	2^{209}	2^{134}
NTRU hps40961229	Level 5	C:274 bits		
NTRU hrss1373	Level 5	C:283 bits		
Light Saber	Level 1	C:118 bits Q:107 bits		
Saber	Level 3	C:189 bits Q:172 bits		
Fire Saber	Level 5	C:260 bits Q:236 bits		

Table 10. Claimed security metrics for the lattice KEM finalists (source: submission documents). The C represents classical, while Q is for quantum.

Candidate	Claimed Security	core SVP Estimate	Gate Count	Memory
Dilithium	Level 2	C:123 bits Q:112 bits	2^{159}	2^{98}
Dilithium	Level 3	C:182 bits Q:165 bits	2^{217}	2^{139}
Dilithium	Level 5	C:252 bits Q:229 bits	2^{285}	2^{187}
FALCON-512	Level 1	C:120 bits Q:108 bits		
FALCON-1024	Level 5	C:273 bits Q:248 bits		

Table 11. Claimed security metrics for the lattice signature finalists (source: submission documents). The C represents classical, while Q is for quantum.

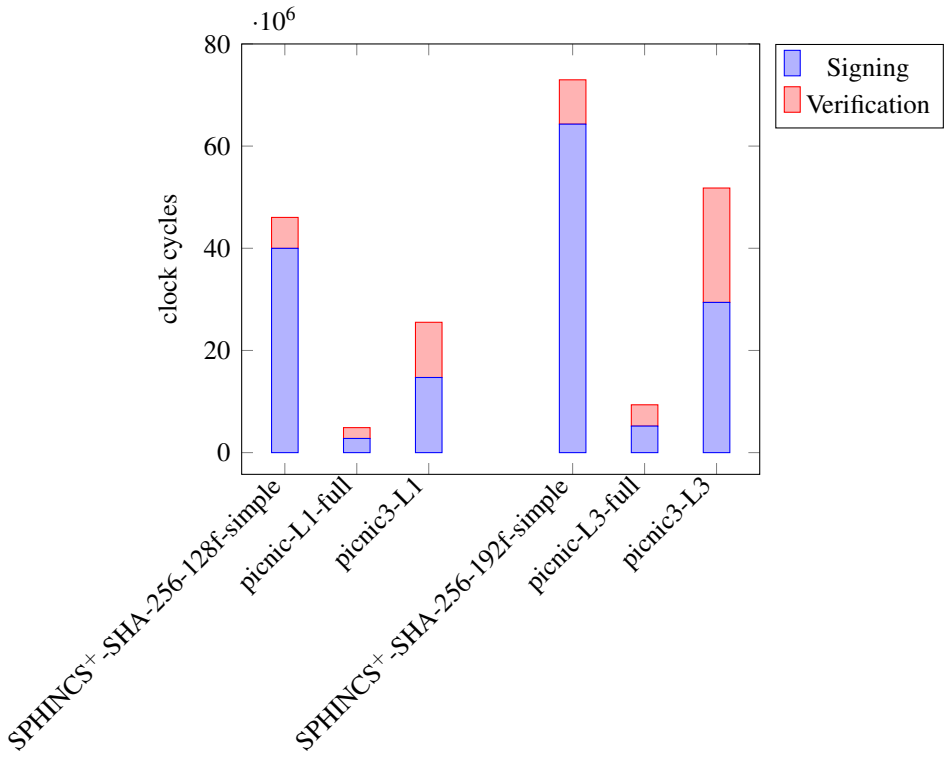

Figure 11. Picnic and SPHINCS$^+$ Benchmarks on x86-64 processor (using average signature sizes)

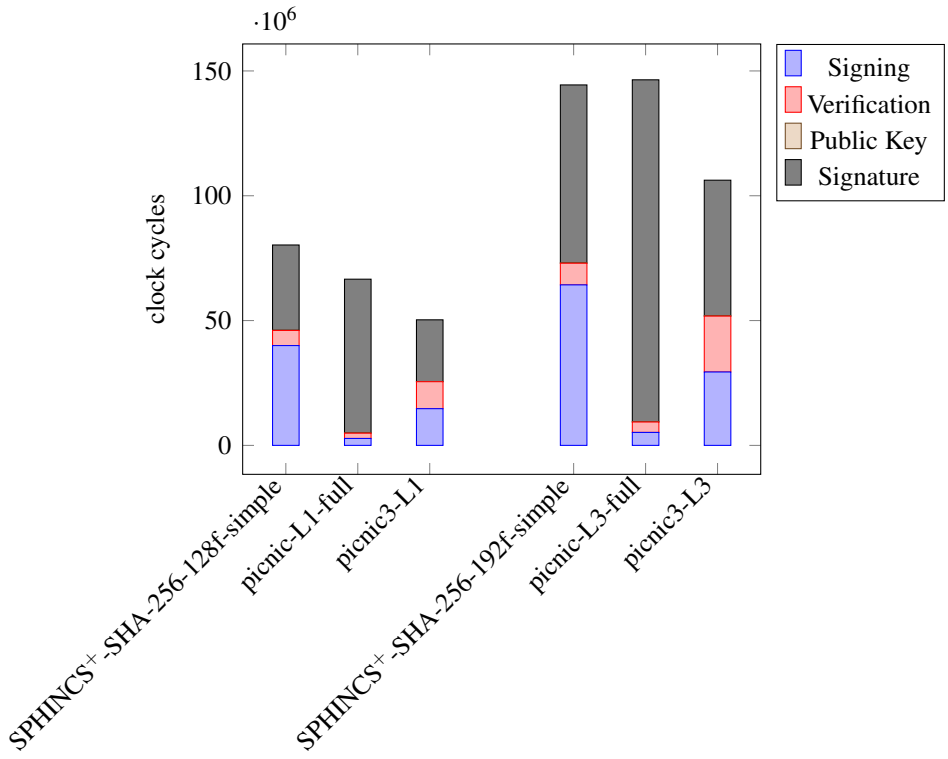

Figure 12. Picnic and SPHINCS$^+$ Benchmarks on x86-64 processor (using average signature sizes) with 2000 cycles/byte transmission costs

E. Change Log

The errata update of 09-26-2022 incorporated the following changes:

- In Section 3.2.3, we changed the paragraph

  > Miccancio [149] introduced a ring-based analogue of Ajtai's SIS problem in 2002. A ring-based analogue of LWE (and an associated public-key encryption scheme) was introduced by Lyubashevsky, Peikert, and Regev [152] in 2010. Further, an algebraically-structured (and in particular, module-based) formulation of SIS/LWE-type problems – which can be syntactically viewed as interpolating between the original integer-based presentation and the later polynomial-ring-based presentations – was first introduced by Brakerski, Gentry, and Vaikuntanathan [153] in 2011 under the name *General Learning With Errors*.

  to

  > Miccancio [149] introduced a ring-based analogue of Ajtai's SIS problem in 2002. A search variant of ring-based LWE (and an associated public-key encryption scheme, relying on the Goldreich-Levin hardcore function

> [150]) was introduced by Stehle, Steinfeld, Tanaka, and Xagawa [151] in 2009. A decisional variant of Ring-LWE with associated public-key encryption scheme (and an associated search-to-decision reduction) was introduced by Lyubashevsky, Peikert, and Regev [152] in 2010. Further, an algebraically-structured (and in particular, module-based) formulation of SIS/LWE-type problems – which can be syntactically viewed as interpolating between the original integer-based presentation and the later polynomial-ring-based presentations – was first introduced by Brakerski, Gentry, and Vaikuntanathan [153] in 2011 under the name *General Learning With Errors*.

The change was made because NIST inadvertently omitted an important reference when discussing the history of lattice-based cryptography.

- References [150] and [151] were added as a result of the amended text. All subsequent references were re-numbered.