Cryptographic Standards in a Post-Quantum Era

Dustin Moody and Angela Robinson

July 11, 2022

1 Introduction

In 1994, Peter Shor discovered a *quantum* algorithm that could efficiently find the prime factors of large integers [1]. Mathematicians have long been interested in factoring algorithms and have developed a variety of techniques to factoring. The problem has been of renewed interest the past several decades because the widely used RSA cryptosystem relies upon the presumed intractability of factoring. The best known classical algorithm, the general number field sieve, takes sub-exponential time in the size of the integer (i.e., the number of bits in the binary representation of the number being factored). The parameters used in RSA for modern security levels use integers so large [1] that even with exceptional computing power the general number field sieve is too inefficient. What makes Shor's algorithm so notable is that it would run in polynomial time on a quantum computer.

Quantum computers are machines that harness the properties of quantum physics to store data and perform calculations. Researchers and engineers around the world have been steadily making progress at building larger and larger quantum computers. While they will not outperform classical computers universally, there are certain applications where they may provide colossal speedups in areas such as computational chemistry, artificial intelligence, machine learning, financial modelling, and drug design (to name just a few). Currently, quantum computers have not yet advanced to the state where they are outperforming today's computers for these applications, but they may in the next few decades.

While the applications listed above would yield positive benefits to society, Shor's algorithm would be more disruptive. In our connected world, information is protected through the use of cryptography. Every day we use the internet, mobile phones, social networks and cloud computing to communicate securely and make financial transactions. Behind the scenes, the protocols that run our digital infrastructure depend crucially on a few cryptographic primitives: public key encryption, digital signatures and key exchange. Together, the functionality

[1] Most current implementations of RSA typically use integers that are 2048 bits or 3072 bits long.

these algorithms provide is known as public-key cryptography. Specific public-key cryptosystems include such well-known algorithms as RSA, Diffie-Hellman, and elliptic curve cryptography (ECC). The security of each depends on the hardness of certain mathematical problems. As mentioned previously, RSA relies on the integer factorization problem being difficult, while Diffie-Hellman and ECC both rely on the hardness of a certain number theoretic problem known as the discrete logarithm problem. Remarkably, with a large enough quantum computer, Shor's algorithm would be able to solve each of these problems efficiently, putting in peril the cryptographic security we rely upon.

The impact of quantum computers on cryptography extends further. Public-key cryptography is often used to create a shared secret key between two parties who have not previously communicated. After this shared secret key is known to both parties, they can use faster symmetric-key cryptographic algorithms to encrypt information with block ciphers like AES. Grover's algorithm [2] is a generic quantum algorithm that yields a quadratic speedup on unstructured search, meaning an attempt to do a brute force exhaustive key search that would normally take $O(N)$ steps could be done in just $O(\sqrt{N})$ steps. While the consequences are not as drastic as the effect Shor's algorithm will have on public-key cryptography, Grover's algorithm will likely require changes to the symmetric-key cryptosystems we use today. Fortunately these changes are much more manageable, for example, doubling the length of the secret key used for AES should provide sufficient protection. For public-key cryptography, the cryptosystems we use today will have to be replaced with new ones.

Existing quantum computers are not yet large enough to implement Shor's (or Grover's) algorithm so that they threaten the security of the cryptographic algorithms currently used. It is not known for certain when such a quantum computer could be built, although experts speculate that it may be possible within the next two decades [3]. While one might wonder why this is a problem now, there is a potential threat to users today – even though a large-scale quantum computer does not yet exist. Suppose an adversary managed to gain access to some sensitive encrypted data. While they cannot break the cryptography protecting the data today, they could simply wait for the time when a big enough quantum computer is available, and then use it to break the encryption and gain access to the data. If the time until a quantum computer is shorter than the time desired to keep the data secret, there is a threat already.

2 Post-Quantum Cryptography

Cryptosystems believed to be resistant to attacks by quantum computers are broadly referred to as quantum-safe or *post-quantum* cryptography (PQC). These schemes are classified according to the underlying mathematical problem upon which the security is based and the corresponding mathematical objects involved. Recall that the security of RSA is based on the difficulty of the integer factorization problem. Thus the mathematical objects involved in RSA are integers (and rings of integers modulo n). PQC schemes are largely based on one

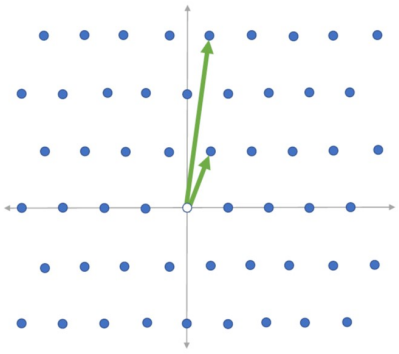

Figure 1: Skew basis for lattice in $\mathbb{R}^2$

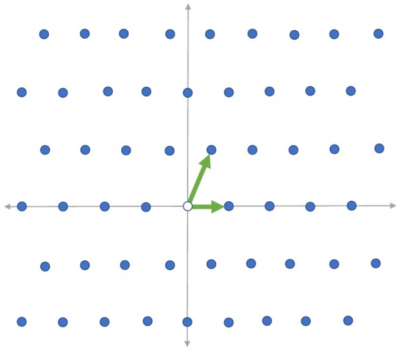

Figure 2: Second basis for same lattice in $\mathbb{R}^2$

of the following mathematical objects: lattices, error-correcting codes, systems of multivariate polynomials, isogenies between elliptic curves, or cryptographic hash functions. In this section we briefly describe these main families of PQC and the corresponding computational problems upon which security relies. Several details are omitted for simplicity, but the interested reader is encouraged to see references for rigorous definitions and descriptions.

Lattices. A lattice is a set of evenly-spaced points in some space $\mathcal{S}$. A lattice $\mathcal{L}$ is generated by a finite number of vectors $\{b_0, b_1, \ldots, b_n\}$ so that any lattice point can be represented as an integer linear combination of of the b_i. The collection of n vectors is known as the *basis* of the lattice, where n is the dimension of the lattice. Figure 1 shows an example of a lattice in $\mathbb{R}^2$ than can be fully represented by the two bold vectors. Figure 2 shows the same lattice represented by another pair of bold vectors. This 2-dimensional lattice can be represented by infinitely many bases.

One of the computationally difficult problems in lattice theory is known as

the shortest vector problem (SVP). As the name suggests, given a lattice $\mathcal{L}$ one must find (one of) the shortest nonzero vector(s) in $\mathcal{L}$. At a high level, lattice bases with more orthogonal vectors enable more efficient solving of the SVP, while bases containing longer, more skew vectors make solving SVP more difficult. For lattices with a high enough dimension, SVP is believed to be secure against quantum computers. As a result several post-quantum cryptosystems have been based on the SVP and variants.

Public-key cryptosystems based on lattices typically involve making a "bad" basis of a lattice public, while incorporating a short vector of the lattice into a secret value. Ajtai presented a lattice-based cryptosystem in 1996 and independently, Hoffstein, Pipher, and Silverman's NTRU encryption scheme was presented in 1998 [4]. The field of lattice-based research and algorithmic design has expanded to meet a broad range of cryptographic needs – zero knowledge proofs, homomorphic encryption, digital signatures, etc. Algorithms based on *unstructured* lattices typically have a direct security reduction to a well understood worst-case hardness problem while *structured*-lattice-based schemes often feature significant efficiency gains and security assumptions that are not as well understood. In general, lattice-based signatures and KEMs offer balanced performance profiles with reasonably sized parameters and very efficient run times.

Error-correcting codes. Error-correcting codes were designed to identify and remove environmental noise from communication transmissions. Let $\mathcal{C}$ be an error-correcting code and $\mathcal{D}$ an efficient decoder for $\mathcal{C}$. The sender of a message m who wishes to use the error-correcting capabilities of $\mathcal{C}$ must first map m into $\mathcal{C}$ or *encode* m into a codeword c. The codeword is then sent across noisy channels to the recipient. The recipient receives a noisy codeword $c + e$ and uses the efficient decoder $\mathcal{D}$ to remove the introduced errors and recover the codeword c. For more information on error-correcting codes and coding theory, see [5].

In 1978 Robert McEliece showed how intentionally added noise to a message could be used to create a public-key encryption scheme [6]. It turns out that any particular code $\mathcal{C}$ can be described in several equivalent ways, and some descriptions enable more efficient decoding than others. This is exploited in the McEliece cryptosystem by making the public key a description of the code that does not lead to efficient decoding, while the private key is a description of the code that enables efficient decoding. In this way, only the private key holder will be able to decrypt and recover an underlying secret message. This is a somewhat loose description of the general decoding problem which is believed to be secure against quantum computers (up to certain choice of codes and parameters).

The McEliece cryptosystem has survived decades of security analysis but has not reached widespread adoption due to exceptionally large key sizes. In particular, the McEliece public key is approximately 1 MB. Cryptographers have worked towards preserving the security guarantees of McEliece encryption while also reducing key sizes.

Systems of multivariate quadratic equations Before we describe systems of multivariate equations, recall the premise of solving a system of linear equations from a standard algebra course. Consider:

$$\begin{aligned} 4x_1 - x_2 &= 1 \\ -3x_1 + 2x_2 &= 8, \end{aligned}$$

which is a system of two linear equations in two variables x_1, x_2. The coefficients of this system $\{4, -1, -3, 2\}$ and the solution $(2, 7)$ are all real numbers, or elements of the (infinite) field $\mathbb{R}$. A matrix representation of the same system of equations is given by

$$\begin{bmatrix} 4 & -1 \\ -3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \end{bmatrix}. \tag{1}$$

A general system of multivariate quadratic equations involves m equations in n variables with coefficients and solutions (if any) in some field . Such a system includes equations with polynomials of degree at most 2 by including product terms of the form $x_i x_j, i \neq j, i, j \in \{1, \ldots n\}$.

Solving systems of multivariate quadratic (MQ) equations over a finite field $\mathbb{F}$ is quite difficult. One may then (loosely speaking) construct a cryptosystem based on this problem by making public the matrix of coefficients and the right-hand side of equation (1) and incorporating the solution vector into the shared secret. MQ-based cryptosystems typically feature large public keys and small signatures. Many researchers have attempted to reduce the public key sizes of MQ-based cryptosystems, with limited success.

Isogenies between elliptic curves. Informally, elliptic curves can be thought of as the set of points (x, y) satisfying the equation $y^2 = x^3 + Ax + B$, for constants A, B. While not obvious, it has long been known how to define an addition law for elliptic curve points so that the set of points forms an abelian group. See Figure 3 for a geometric interpretation of the group law.

Using elliptic curves for cryptography was first introduced around 1985, and their usage has become widespread since then. As mentioned in the introduction, elliptic curve cryptosystems currently in wide use are vulnerable to Shor's algorithm on a quantum computer. Interestingly, cryptographers have devised a new way to use elliptic curves, which is believed to be resistant to attacks from quantum computers.

The new approach relies on maps between (supersingular) elliptic curves which respect their group structure, known as *isogenies*. The basic idea behind the hard problem their security relies on is to explicitly find an isogeny, given two curves for which it is known that there is an isogeny between them. Isogeny-based cryptography is a newer approach, but is seen as very promising since the key sizes of the resulting cryptosystems are very small. One of the disadvantages is that implementations of isogeny-based cryptosystems are typically an order of magnitude slower than other PQC solutions.

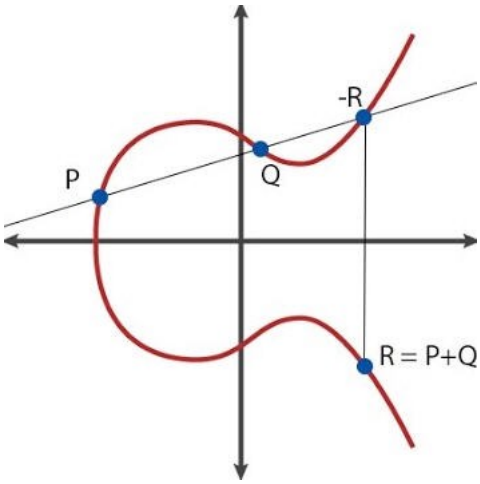

Figure 3: The addition law for points on an elliptic curve

Cryptographic hash functions One-way functions - functions that are easy to compute but computationally difficult to invert - are the foundation of public-key cryptography. Examples of functions believed to be one-way are seen in the RSA, Diffie-Hellman, and ECC, where the difficulty of inversion directly correlates to the security of the algorithm.

We say h is a one-way hash function if h maps arbitrary-length strings (documents, messages, etc.) to fixed-length strings of length n and the following two properties hold: (1) It is computationally infeasible, given some output $y = h(m)$, to find an input that maps to y under h and (2) Given a string x, it is computationally infeasible to find a different string y so that $h(x) = h(y)$. These properties are formally referred to as preimage resistance and second-preimage resistance, respectfully. The naive approach to finding the preimage or second-preimage is through a brute-force search requiring approximately 2^n elementary operations. However, the quantum Grover algorithm and other quantum collision-finding algorithms only require approximately $\sqrt{2^n} = 2^{\frac{n}{2}}$ (or less) elementary operations. As such, cryptographic hash functions are not believed to be "broken" by quantum computers, but it is recommended that the cryptographic hash function output length n be twice the security level to protect against Grover's algorithm.

Whitfield Diffie and Martin Hellman first proposed the use of one-way hash functions in cryptography in 1976, and referenced Leslie Lamport's partial hash-based signature scheme solution. Shortly after, in 1979, Lamport published a full hash-based signature scheme where the signing key can only be used once. The security of the signature scheme relies only on the hash function being one-way and on the length n of the hash values (outputs). Cryptographic hash functions have since been used as important components of zero-knowledge proofs, Merkle trees, key derivation functions, as well as several digital signature schemes.

3 The NIST PQC Standardization Project

The mission of the National Institute of Standards and Technology is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life. The NIST Cryptographic Technology Group has developed cryptographic standards for digital signatures (FIPS 186), hash functions (FIPS 180 and FIPS 202), symmetric encryption (FIPS 197), and public-key encryption (SP 800-56B), among others. All public-key cryptosystems currently standardized by NIST, though efficient and widely deployed, are known to be vulnerable to quantum attacks as they are based on variations of the integer factorization and discrete logarithm problems. NIST initiated a process to update its public-key standards to schemes which will resist the quantum computing threat.

The NIST PQC Standardization Process began in 2016 with a call for proposals for post-quantum digital signatures and post-quantum public-key encryption or key encapsulation mechanisms (PKE/KEMs) [7]. Submission packages were required to provide detailed algorithmic specifications, security analyses against both classical and quantum attacks, reference and optimized C code implementations, known-answer tests, and statements on intellectual property. NIST defined five security levels to correspond to both classical bit security as well as security against quantum attacks. Submitters were encouraged to provide parameter sets to target a range of security levels.

NIST received 82 total submissions and 69 of these proposals fit both the minimum acceptance criteria and submission requirements. A majority of the 69 candidates were based on lattices or error-correcting codes. The first NIST PQC Standardization Conference was held in 2018, and each submission team was invited to present their algorithm. NIST studied the security analyses provided by submitters and also considered cryptanalysis conducted by the PQC community. The security of schemes and computational efficiency of hardware and software and storage requirements were the primary considerations during the first round. Benchmarking and real-world experiments conducted by the cryptographic community, as well as discussion on the pqc-forum[2], were invaluable to NIST during candidate evaluation.

In 2019 NIST announced that 26 schemes were advancing to a second round of evaluation. NIST published a report [9] naming the selected schemes, along with the rationale for their selection. As in the first round, lattice- and code-based schemes again accounted for the majority of the second-round candidates (see Tables 3 and 3). NIST held a second workshop, where detailed research results and submission teams' updates were both included. Just as happened in the first round, during the second round several schemes were broken, or had their security weakened as a result of published attacks.

After another year of analysis, NIST announced in 2020 that 7 finalist and 8 alternate algorithms would advance to the third round [10]. The finalists

[2]NIST has set up an online pqc-forum mailing list [8]. The mailing list is used to discuss the standardization and adoption of secure, interoperable and efficient post-quantum algorithms.

KEMs			
Type	Round 1	Round 2	Round 3
Lattice-based	21	9	5
Code-based	17	7	3
Multivariate	2	0	0
Other	5	1	1
Total	45	17	9

Table 1: NIST PQC KEM candidates across three rounds

Signatures			
Type	Round 1	Round 2	Round 3
Lattice-based	5	3	2
Code-based	2	0	0
Multivariate	7	4	2
Hash/symmetric	3	2	2
Other	2	0	0
Total	19	9	6

Table 2: NIST PQC signature candidates across three rounds

were those algorithms that NIST considered to be the most promising for the majority of use cases and the most likely to be ready for standardization at the end of the third round. The alternate candidates were regarded as potential candidates for future standardization, but would likely need another round of evaluation. Included among the set of finalists were 4 KEM algorithms, and 3 signature schemes. Of these seven finalists, five were based on structured lattices.

4 The Path Forward

In July 2022, six years after the PQC standardization process was announced, NIST made public the first algorithms they would be standardizing [11]. For public-key encryption (or key-establishment), NIST selected CRYSTALS-Kyber, one of the third-round finalists. For digital signatures, NIST will standardize CRYSTALS-Dilithium as the primary algorithm it will recommend. In addition, the signature algorithms Falcon and SPHINCS+ will also be standardized.

Both Kyber and Dilithium are based on structured lattices, and offer strong security and excellent performance. The signature algorithm Falcon was selected for standardization because there may be use cases where Dilithium's signature sizes are too large. As both Dilithium and Falcon are lattice-based, NIST also will standardize SPHINCS+ in order to not rely solely on the security of lattices for signatures.

For encryption, NIST will continue to evaluate four candidates from the third

round: BIKE, Classic McEliece, HQC, and SIKE. The first three are based on error-correcting codes, while SIKE is based on isogenies of elliptic curves. This fourth round of evaluation will continue for about two years, after which NIST intends to select at least one to standardize to diversify its encryption portfolio.

While the fourth round is ongoing, NIST will begin drafting the first post-quantum cryptography standards to replace those which are vulnerable to attacks from a quantum computer. The completed standards will provide concrete parameter sets and specify how to implement each algorithm. It will also describe the security offered by the different parameter sets, as well as provide guidance for safe usage of the algorithms. NIST will post the drafts for public comment, and it is expected that the standards will be published by 2024.

At the conclusion of the third round, SPHINCS+ was the most promising non-lattice signature. While NIST is standardizing SPHINCS+ in order to have a non-lattice signature algorithm, the performance of SPHINCS+ will likely not be good enough for many applications. The signature sizes for SPHINCS+ are many times larger than Dilithium, and both signing and verifying are at least an order of magnitude slower (signing is particularly painful). As a result, NIST has announced that it will issue a new public call for digital signature algorithms in the near future. These signature submissions will be evaluated in a new undertaking, similar in nature to the PQC standardization effort that is ending. It is expected that this newer signature process will be much smaller in scope, with the main objective being to find a better performing non-lattice signature scheme that could be used by most applications. The submitted signature schemes will need to be thoroughly analyzed, which will take several years before standardization could occur.

It should be noted that standardization efforts in this area will continue for some time, even though NIST is beginning to draft the first PQC standards. 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 more guidance on the transition.

The transition to PQC algorithms will undoubtedly have many complexities, and there will be challenges for some use cases. There has been some interest from industry in considering hybrid mechanisms, which combine the use of both a post-quantum algorithm with an already existing and standardized classical algorithm. The idea being that to break the security of the hybrid scheme, an attacker would need to break both algorithms. While this would obviously entail worse performance, it may be a good way to get experience with using the newer post-quantum algorithms.

Several other organizations and research groups are also working on post-quantum standards and recommendations. For example, the Institute of Electrical and Electronics Engineers (IEEE) developed some of the earliest lattice-based standards. The Internet Engineering Task Force (IETF) has standardized hash-based signatures and is exploring hybrid approaches to PQC. The European Telecommunications Standards Institute (ETSI) has developed recommendations and guidelines around PQC preparation and transition, in addition to quantum key distribution (QKD) standards, and the International Organization

for Standardization (ISO) is also doing standardization work.

Even though the NIST PQC standardization process is nearing completion, much work will still continue. The field of post-quantum cryptography is a young and very active research area, and undoubtedly new ideas will be found in the coming years. With the newly standardized algorithms, users can begin to protect their information from the threat of future quantum computers. There have been many challenges that have had to be overcome, and there will be more which are yet unforeseen. NIST is grateful for the hard work and cooperation from so many that have enabled us to get to where we are today. For more information on NIST's post-quantum cryptography project, see `www.nist.gov/pqcrypto`.

References

[1] Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th annual symposium on foundations of computer science* (Ieee), pp 124–134.

[2] Grover LK (1996) A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp 212–219.

[3] Mosca M, Piani M (2022) Quantum threat timeline report 2021. *Website: https://globalriskinstituteorg/publications/2021-quantum-threat-timeline-report/* .

[4] 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.

[5] Van Lint JH (2012) *Introduction to coding theory*. Vol. 86 (Springer Science & Business Media), .

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

[7] 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

[8] (2022) NIST pqc-forum mailing list. Available at `https://groups.google.com/a/list.nist.gov/g/pqc-forum`.

[9] 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

[10] 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

[11] Alagic G, Apon D, Cooper D, Dang Q, Dang T, Kelsey J, Lichtinger J, Liu YK, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D (2022) Status report on the third round of the NIST post-quantum cryptography standardization process (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8413. https://doi.org/10.6028/NIST.IR.8413