Post Quantum Cryptography(PQC) - An overview

(Invited Paper)

Manoj Kumar and Pratap Pattnaik
IBM Thomas J. Watson Research Center
{manoj1,pratap}@us.ibm.com

***Abstract*—We discuss the Post Quantum Cryptography algorithms for key establishment under consideration by NIST for standardization. Three of these, Crystals-Kyber, Classic McEliece and Supersingular Isogeny based Key Encapsulation (SIKE), are representatives of the three classes of hard problems underlying the security of almost all 69 candidate algorithms accepted by NIST for consideration in round 1 of evaluation. For each algorithm, we briefly describe the hard problem underlying the algorithm's cryptographic strength, the algebraic structure i.e., the groups or finite fields, underlying the computations, the basic computations performed in these algorithms, the algorithm itself, and the performance considerations for efficient implementation of the basic algorithm on conventional many-core processors. For Crystals-Kyber and SIKE, we will discuss the potential solutions to improve their performance on many-core processors.**

I. Introduction

Most public-key cryptographic algorithms in use today, such as RSA, Diffie-Hellman etc., rely on the conjecture that it is "hard" to solve any of the following three problems in a reasonable time: a. factoring a large integer, b. order-finding and c. finding discrete logarithm. If any of these problems can be solved in reasonable complexity, i.e. polynomial time, then the solution can be used to break the encryption and compute the secret keys. All known classical computer algorithms require at least exponential $(O(n\,2^n))$ time, where n is the integer being factored or the argument to the discrete log. Furthermore, even symmetric key algorithms require public-key algorithms to agree on a shared secret, the symmetric encryption key.

It is known that quantum computers can solve any of the above three problems with complexity $O(n^2)$, i.e. in polynomial time. Thus, with the availability of quantum computers, the current cryptographic methods can be broken easily. The key behind the efficiency of quantum computers is that they have a natural way of doing the Fourier transform on vectors of length $N = 2^n$ in $O(n^2)$ steps, much less than the $O(n\,2^n)$ steps required on a classical computer. Using this result, Shor [1] developed an algorithm to factor an integer in $O(n^2)$ steps. Others have extended this technique to solve the other hard problems of order-finding and finding discrete logarithms, also within the same complexity [2].

A. When will adoption of PQC happen?

The Accenture report [3] cites academic research consensus that by 2028 quantum computer will be capable of implementing Shor's algorithm at the scale needed to break current cryptographic algorithms. It also states that, in the authors' opinion, the inflection may happen by 2026. Researchers continue to make significant progress in increasing qbit lifetime [4]. Recent announcement by NIST of the third round finalists in their PQC standardization process [5], along with the revised time-line, suggests that draft standards are likely to be available in 2022.

Business and organizations must ensure that their sensitive data encrypted using traditional cryptographic approaches, vulnerable to quantum computing based attacks, stays indecipherable for many years after the traditional approaches are compromised by quantum computing. This in turn requires them to adopt a new class of algorithms resilient against attacks by quantum computers, collectively known as post quantum cryptography (PQC), many years ahead of the anticipated access to quantum computing by their adversaries. Google had tested NewHope, one of the early round PQC candidates, in combination with Elliptic Curve Diffie-Hellman on Chrome Canary.

B. PQC algorithms under consideration for standardization by NIST

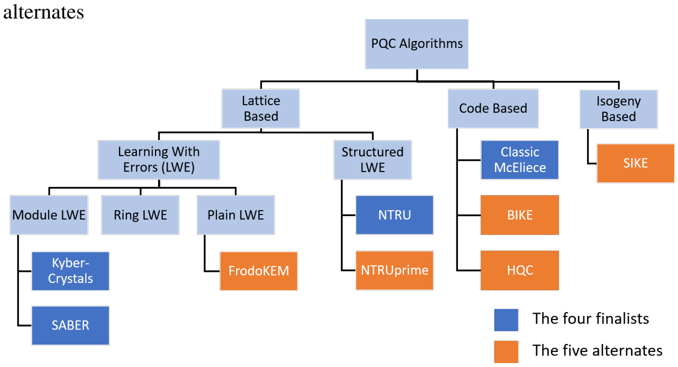
NIST standardization of Post-Quantum Cryptography started in 2017 with 69 candidate algorithms for key establishment mechanisms (KEM) and digital signatures. In January 2019, based on evaluation/analysis of their security, performance, key length and other such characteristics, 26 algorithms advanced to the second round for more analysis. In July 2020, 15 moved to the third round, seven as finalists and eight as alternate. The 9 KEM algorithms advancing to the third round, shown in Figure 1, fall in three categories, lattice based, code based, and isogeny based. Furthermore, they are grouped as four finalists and five alternates. From the KEM finalists, NIST currently intends to declare one KEM algorithm as the standard. The key outcomes of the evaluations, in terms of relative advantages and disadvantages of the second round algorithms, are summarized in [6]. The alternate KEM algorithms: 1) serve as a back-up in case security vulnerability is exposed in the finalists during the third round evaluation; 2) can be standardized in a future round four for targeted use cases such as ultra fool-proof security, or very short key-lengths, etc.; or 3) simply need to give more time to the cryptography community to prove their mettle.

C. Outline of the paper

In the next section we will provide an overview of the hard lattice problems and their connection to cryptography

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

Fig. 1. The Post Quantum Cryptography Key-Establishment Mechanism (KEM) algorithms advancing to the third round, the four finalists and three alternates



constructs, the short integer solution (SIS) and Learning With Error (LWE). In section III we will explain the variants of LWE, the plain-LWE, Ring-LWE, and the Module-LWE, and explain the performance concerns driving the evolution and the security concerns arising from this evolution. We will also give an example of each from the NIST submissions. Kyber is one of them and is a NIST KEM finalist. In section IV we will present the basic computational structure of Classic McEliece, also a NIST KEM finalist. In section V we explain elliptic curves, isogenies over them and describe the SIKE (Supersingular Isogeny based Key Exchange) protocol based on it. Finally in section VI we discuss the comparative advantage of the KEM algorithms. For the alternatives, we explain their plausible attractiveness for specific use cases.

Our focus in this paper is on the computational constructs in the PQC KEM algorithms, a sketch of the theory that gives rise to the computational constructs, and the computational performance aspects. Due to space/time limitations, we specifically have omitted three important topics: 1) the PQC digital signature algorithms; 2) The history of successful attacks on PQC KEM algorithms and the consequent evolution; and 3) reduction from conjecture or established hard problems to the cryptographic constructs used.

II. Background for Lattice Based PQC Algorithms

In this section we first describe hard lattice problems and their useful variants, the approximate version of the problem and the search/decision version. Then we introduce the hard problems used in modern lattice-based cryptography, namely the Short Integer Solution (SIS) problem and the Learning With Errors (LWE) problem. For LWE, we explain its three variants, plain-LWE, Ring-LWE and Module-LWE [7]. In the cryptography literature, the hardness or cryptographic robustness of SIS or LWE is proven by reducing them to the hard lattice problems.

A. Hard Lattice Problems

Ajtai, in his seminal papers [8], [9] in 1996, described three hard problems for random instances of n dimensional lattices $\mathcal{L}(\mathcal{B}) \subset \mathbb{R}^n$, defined as integer weighted sums of basis vectors $\mathcal{B} = [\bar{b}_0, \bar{b}_1, \cdot\cdot\cdot, \bar{b}_{n-1}] \in \mathbb{R}^n$.

$$\mathcal{L}(\mathcal{B}) = \sum_{i=0}^{n-1} x_i \bar{b}_i : x_i \in \mathbb{Z} \tag{1}$$

These problems have a desirable property that if one can solve a random instance of these problems, then one can also solve the worst-case instance. Hence they are also called *worst-case* problems. The three hard problems are:

- Finding the length of the shortest non-zero vector, approximately up to a polynomial (in n) factor.
- Finding the shortest non-zero vector v, where v is unique in the sense that all vectors shorter than $n^c||v||$ are parallel to v for a sufficiently large absolute constant c. (Shortest Vector Problem (**SVP**)).
- Finding a basis $[\bar{b}_0, \bar{b}_1, \cdot\cdot\cdot, \bar{b}_{n-1}]$ of the lattice of smallest possible length up to a polynomial factor. Length is defined as $\max_i ||\bar{b}_i||$. (Shortest Independent Vectors Problem (**SIVP**)).

Minkowski's theorem states that the length of the shortest vector in the above lattice has an upper bound of $\sqrt{n}(det\ [\bar{b}_0, \bar{b}_1, \cdot\cdot\cdot, \bar{b}_{n-1}]\ )^{\frac{1}{n}}$.

The above hard lattice problems, i.e., *worst-case* lattice problems, have defied any polynomial time solution for over a century and half, and are thus conjectured to have no polynomial-time solution. These problems also have variants that are equally hard [10]. The first is the exact vs. approximate variant. For example, the approximate variant of SVP states that finding a vector in a lattice that is no larger than $\gamma(n)$ times the shortest vector, a problem denoted as $\mathbf{SVP}_\gamma$, is as hard as the **SVP** problem. The second variant is the search vs. decision variant. For example, the decision variant of SVP states that determining whether the shortest vector is either less than 1, or greater than $\gamma(n)$, a problem denoted as $\mathbf{GapSVP}_\gamma$, is as hard as the **SVP** problem.

B. Average-case lattice problems in modern cryptography

For performance and algorithmic reasons, the above worst-case lattice problems are not directly usable in modern lattice-based cryptography. Most modern cryptographic schemes are based on hard problems different from the hard lattice problems, and also referred to as *average-case* problems. Two of the most frequently used average-case problems are Short Integer Solution (**SIS**), and Learning With Errors (**LWE**). Each average-case problem is related to some worst-case problem, and the relationship has a very important attribute - *if one can solve any instance of the average-case problem, then one can solve all instances of the corresponding worst-case problem.* This simplifies the proofs for cryptographic robustness of the lattice based cryptography algorithms.

Ajtai in [8] also established the foundation of lattice-based cryptography by mapping the hard lattice problems on $\mathcal{L}(\mathcal{B})$ to hard lattice problems on lattices $\Lambda^T(A, q)$ defined below. He showed that if a polynomial time algorithm exists for finding a

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

short vector in a q-ary lattice $\Lambda^T(A, q)$, defined as a collection of points $h \in \mathbb{Z}^m$ satisfying

$$\Lambda^T(A, q) \triangleq \sum_{i=0}^{m} h_i \bar{a}_i = \bar{0}^n (mod\, q) : a_i \in \mathbb{Z}_q^n \qquad (2)$$

then an algorithm exists to solve the three hard problems defined above for $\mathcal{L}(\mathcal{B})$. This is the Short Integer Solutions (**SIS**) Problem, as hard as SIVP on $\Lambda^T(A, q)$. The cryptographic constructs such as key generation, encryption, and decryption work on the dual of the above q-ary lattice is defined as

$$\Lambda(A, q) \triangleq \bar{y} \in \mathbb{Z}^m : y = A^T \bar{s} \quad \text{for some} \quad \bar{s} \in \mathbb{Z}^n \qquad (3)$$

(Note: The more recent publications [11]–[14] switch the notation used by Ajtai [8] for Λ and the q-ary lattice its dual Λ^T. We are using the notation of recent publications rather than that used in [8].)

Another average-case hard problems underlying lattice-based PQC methods is the Learning with Errors (LWE) problem [7]. This problem asserts that given a lattice of points in $\mathbb{Z}_q^n$ generated by a basis A of uniformly sampled n-dimensional vectors (n approximately in the range of 500 to 1000), a secret key s and an error vector e, both short vectors in $\mathbb{Z}_q^n$ sampled from a narrow distribution, it is hard to distinguish between a vector $t = As + e$ and a random value t' [10]. Learning With Errors (LWE) relies on the decisional and approximate variant of SVP, GapSVP$_\gamma$.

III. Lattice Based Crypto Systems

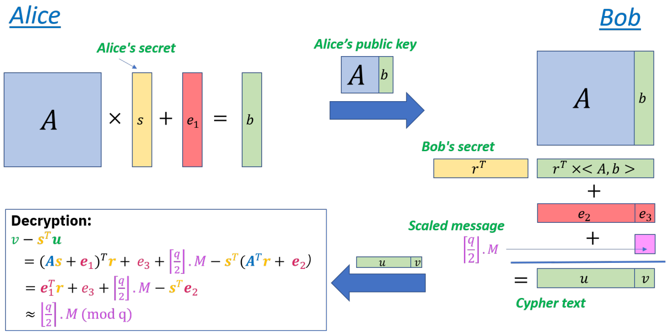
In this section we describe the plain-LWE algorithm, the first to appear on the LWE scene, and its evolution to Ring-LWE and then Module-LWE. Three of the four KEM finalist in round three of NIST standardization process are lattice-based algorithms, namely Kyber, SABER, and NTRU. The first two use Module-LWE. One of the four KEM alternates, FrodoKEM, uses plain-LWE. We give a brief description of FrodoKEM, Kyber, and SABER and NTRU, along with NewHope , a Ring-LWE algorithm. The section concludes with a brief discussion on the computational behavior of the Ring and Module LWEs.

A. Plain-LWE based cryptosystems and the Frodo algorithm

The LWE crypto systems are comprised of the following steps (or similar ones), described below and illustrated in Figure 2. A $m \times n$ matrix A is sampled from $\mathbb{Z}_q$

- **Key generation**: A private key $\bar{s} \in \mathbb{Z}_q^n$ of Alice the receiver is sampled element-wise from a narrow distribution χ, and the public key is constructed by Alice as $\langle A, \bar{b} = (A\bar{s} + \bar{e}_1) \rangle$. Here, $\bar{e}_1 \in \mathbb{Z}_q^m$ is also sampled element-wise from χ, its addition prevents an eavesdropper from recovering $\bar{s}$ by Gaussian elimination.
- **Encryption**: Bob, the receiver, chooses his private key $\bar{r} \in \mathbb{Z}_q^m$, also sampled element-wise from χ, and creates the cyphertext $\langle \bar{u} = A^T \bar{r} + \bar{e}_2, v = (\bar{b}^T \bar{r} + e_3 + M\frac{q}{2}) \rangle$, where M is a one bit message to be encrypted. Once again $\bar{e}_2$ and e_3 are added to prevent eavesdropper from recovering $\bar{r}$.

Fig. 2. Key exchange protocol based on learning with errors (LWE)


Alice
Bob
Alice's secret
A
×
s
+
e_1
=
b
Alice's public key
A b
A
b
Bob's secret
r^T
$r^T \times < A, b >$
+
e_2
e_3
Scaled message
$\left[\frac{q}{2}\right] . M$
+
=
u
v
Cypher text
u v
Decryption:
$v - s^T u$
$= (As + e_1)^T r + e_3 + \left[\frac{q}{2}\right] . M - s^T (A^T r + e_2)$
$= e_1^T r + e_3 + \left[\frac{q}{2}\right] . M - s^T e_2$
$\approx \left[\frac{q}{2}\right] . M$ (mod q)

- **Decryption**: Alice computes $v - \bar{s}^T \bar{u}$ and rounds off the results to recover M, as illustrated in Figure 2.

The narrow distribution χ is a discrete Gaussian or some variant/approximation for it, or often for computational efficiency reasons a centered binomial distribution. The variance of the distribution is chosen to be large enough to ensure that the samples $\bar{e}_1, \bar{e}_2, e_3, \bar{s}, \bar{r}$ are almost uniformly distributed over $\mathbb{Z}_q^m$. At the same time the variance has to be small enough to ensure that the right hand side terms in the following decryption equation not involving M are much smaller than M, allowing M to be decoded reliably my rounding it off to 0 or q.

$$\begin{aligned} \bar{v} - \bar{s}^T \bar{u} = \bar{e}_1 \bar{r} + e_3 + \lceil \frac{q}{2} \rfloor M + \bar{s}^T \bar{e}_2 \\ \approxeq M \qquad (4) \end{aligned}$$

Recovering the secret key $\bar{s}$ from $(A^T \bar{s} + \bar{e})$, the average case LWE problem, is as hard as solving the worst case approximate decisional Shortest Vector Problem (GapSVP$_\gamma$).

In LWE, the vector $\bar{b}$ sent from Alice to Bob, and $\bar{u}$ sent from Bob to Alice, allow both to develop a close approximation of the shared secret $\bar{r}^T \cdot A \cdot \bar{s}$. However, an eves dropper cannot develop that approximation from the knowledge of $A, \bar{b}, \bar{u}$, and v alone. The distribution of $\bar{b}$ (or $\bar{u}$) given A is completely random [8]. The term v is sum of the approximation of shared secret $\bar{r}^T \cdot A \cdot \bar{s}$ computed by Bob, the error term e_3 and the message $\lfloor q/2 \rfloor M$. Hence, Allice can recover the message by subtracting her approximation of the shard secret $\bar{r}^T \cdot A \cdot \bar{s}$ from $\bar{v}$.

Frodo is an alternate candidate in the third round of NIST evaluation [11]. In the Frodo proposal $A \in \mathbb{Z}_q{}^{n \times n}$. The proposed values of are $n \in \{352, 592, 752, 864\}$, and values of q corresponding to the n are $2^{11}, 2^{12}, 2^{15}$, and 2^{15}. For each value of of n, the $\bar{s}, \bar{e_1}, \bar{r}$ and $\bar{e_2}$ are $n \times \bar{n}$ matrices, and thus $\bar{e_3}$ is $\bar{n} \times \bar{n}$ matrix. $\bar{n}$ is chosen to be 8. The number of message bits extracted off $\bar{v}$, for the four values of n are $1, 2, 4,$ and 4. In Frodo, the shared secret is not encoded explicitly as perturbation, the additive $\lfloor q/2 \rfloor M$ term, in the ring element sent from Bob to Alice as shown in Figure 2. The *rounding* function $\lfloor \cdot \rceil$ and *cross-rounding* function $\langle \cdot \rangle$, are used to implicitly recover the shard secret from the ring element $(\bar{u}, \langle \bar{v} \rangle_{2^B})$, sent from Bob to Alice. This implicit recovery is

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

illustrated in Figure 4, with the code shown in green color, and described in [15].

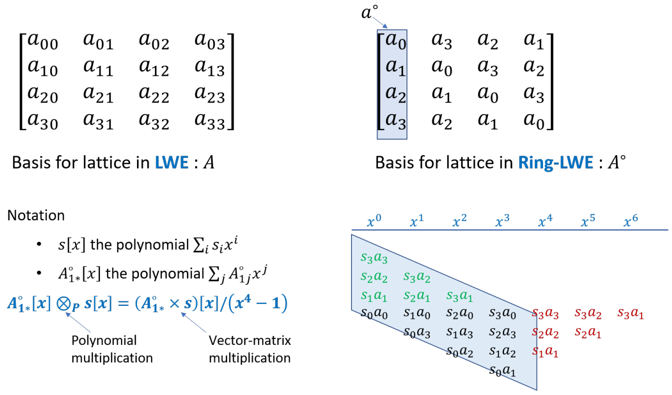
The main drawback of Frodo is the size of matrix A, $\mathcal{O}(n^2)$ or about $500K - 750K$ entries for the *Recommended* and *Paranoid* levels of security. This size will factor in either computing those elements on each use, or storing and transmitting those elements. The matrix is used twice in vector-matrix multiplications, to extract 1,2,4,4 bits at a time corresponding to the four values of n, of the shared key K. The key size is rather large, particularly for the purpose of edge devices, and the computations required could also be large for low power devices.

B. Ring-LWEs and the NewHope protocol

The key size in Ring-LWE is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ by using a circulant matrix A° as the basis for a lattice, an ideal of which is then used for the cryptographic operations. As shown in Figure 3, in a circulant matrix every basis vector (column) is a rotation of the first basis vector a°. The points of this lattice can be represented as polynomials over $\mathbb{Z}_q^m[x]/(x^m - 1)$. As illustrated in lower half of Figure 3, matrix-vector multiplications such as $A^\circ \times s$ become polynomial multiplications $A_{1*}{}^\circ[x] \times s[x] \bmod (x^m - 1)$. In the lower part of the figure, the product terms in red are for powers of x greater than 3, which are reduced mod $(x^4 - 1)$ to the green terms of powers of x less than 4.

The ideal of interest for crypto applications are polynomials of order n in $R_q \triangleq \mathbb{Z}_q^n[x]/(x^n + 1)$. In addition to the reduction in key size, with proper choice of the irreducible polynomial defining the ideal, such as $x^n + 1$, polynomial multiplications or equivalently matrix-vector multiplications can be carried out more efficiently. In R_q polynomial multiplications can be carried out in $\mathcal{O}(n\, log\, n)$ time by using number theoretic transforms (NTT).

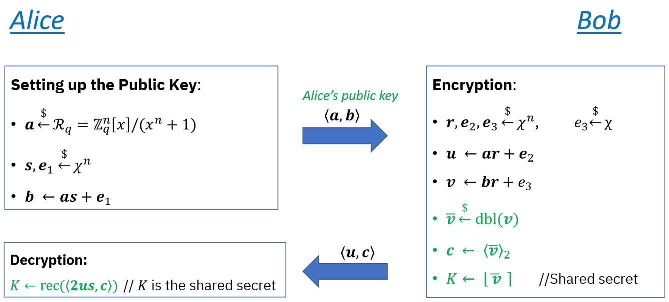
Fig. 3. Equivalence of polynomial multiplications and matrix-vector multiplications in $\mathbb{Z}_q^m[x]/(x^m - 1)$


$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$
Basis for lattice in LWE : A
a°
$\begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}$
Basis for lattice in Ring-LWE : A°
Notation
• $s[x]$ the polynomial $\sum_i s_i x^i$
• $A^\circ_{1*}[x]$ the polynomial $\sum_j A^\circ_{1j} x^j$
$A^\circ_{1*}[x] \otimes_P s[x] = (A^\circ_{1*} \times s)[x]/(x^4 - 1)$
Polynomial multiplication
Vector-matrix multiplication
x^0 x^1 x^2 x^3 x^4 x^5 x^6
s_3a_3
s_2a_2 s_3a_2
s_1a_1 s_2a_1 s_3a_1
s_0a_0 s_1a_0 s_2a_0 s_3a_0 s_3a_3 s_3a_2 s_3a_1
s_0a_3 s_1a_3 s_2a_3 s_2a_2 s_2a_1
s_0a_2 s_1a_2 s_1a_1
s_0a_1

NewHope is a Ring-LWE lattice-based cryptosystem proposed for standardization by NIST [13]. It did not qualify for the third round in NIST evaluations, but we present it here none the less as the intermediate step in the evolution from Plain-LWE to Module-LWE. The key exchange operations of NewHope are shown in Figure 4. Comparing it to the key exchange operations of plain-LWE shown in Figure 2, two differences are observed:

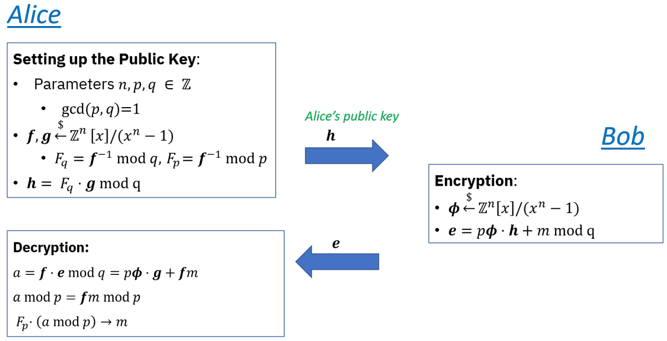
1) Both vector-matrix multiplications, As and $r^T A$, have been replaced by polynomial multiplications $a^\circ s$ and $a^\circ r$
2) The extraction of shared secret from $\langle u, v\rangle$ is different. The two approaches of extracting shared secrets are independently fungible.

Fig. 4. Ring-LWE based key exchange mechanism in NewHope


Alice
Bob
Setting up the Public Key:
• $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}_q = \mathbb{Z}_q^n[x]/(x^n + 1)$
• $\boldsymbol{s}, \boldsymbol{e}_1 \overset{\$}{\leftarrow} \chi^n$
• $\boldsymbol{b} \leftarrow \boldsymbol{as} + \boldsymbol{e}_1$
Alice's public key
$\langle \boldsymbol{a}, \boldsymbol{b}\rangle$
Encryption:
• $\boldsymbol{r}, \boldsymbol{e}_2, \boldsymbol{e}_3 \overset{\$}{\leftarrow} \chi^n,$ $e_3 \overset{\$}{\leftarrow} \chi$
• $\boldsymbol{u} \leftarrow \boldsymbol{ar} + \boldsymbol{e}_2$
• $\boldsymbol{v} \leftarrow \boldsymbol{br} + e_3$
• $\overline{\boldsymbol{v}} \overset{\$}{\leftarrow} \mathrm{dbl}(\boldsymbol{v})$
• $\boldsymbol{c} \leftarrow \langle \overline{\boldsymbol{v}} \rangle_2$
• $K \leftarrow \lfloor \overline{\boldsymbol{v}} \rceil$ //Shared secret
$\langle \boldsymbol{u}, \boldsymbol{c}\rangle$
Decryption:
$K \leftarrow \mathrm{rec}(\langle \boldsymbol{2us}, \boldsymbol{c}\rangle)$ // K is the shared secret

NTRU predates the development of lattice-based cryptography, particularly the LWE methods, however it uses polynomial rings $\mathbb{Z}_q{}^n[x]$, and its cryptographic strengths can be derived from Ring-LWEs [14]. The NTRU protocol is shown in Figure 5. The interesting and unique aspect of the algorithm is the use of multiplicative term p in the cipher text that is used to remove the obfuscation factor $p\,\phi{\cdot}h$ by modules p operation during decoding. All other algorithms have the recipient of cipher text compute and subtract the obfuscation factor.

Fig. 5. Key exchange mechanism in NTRU


Alice
Setting up the Public Key:
• Parameters $n, p, q \in \mathbb{Z}$
• $\gcd(p,q)$=1
• $\boldsymbol{f}, \boldsymbol{g} \overset{\$}{\leftarrow} \mathbb{Z}^n[x]/(x^n - 1)$
• $F_q = \boldsymbol{f}^{-1} \bmod q$, $F_p = \boldsymbol{f}^{-1} \bmod p$
• $\boldsymbol{h} = F_q \cdot \boldsymbol{g} \bmod \mathrm{q}$
Alice's public key
$\boldsymbol{h}$
Bob
Encryption:
• $\boldsymbol{\phi} \overset{\$}{\leftarrow} \mathbb{Z}^n[x]/(x^n - 1)$
• $\boldsymbol{e} = p\boldsymbol{\phi} \cdot \boldsymbol{h} + m \bmod \mathrm{q}$
$\boldsymbol{e}$
Decryption:
$a = \boldsymbol{f} \cdot \boldsymbol{e} \bmod q = p\boldsymbol{\phi} \cdot \boldsymbol{g} + \boldsymbol{f}m$
$a \bmod p = \boldsymbol{f}m \bmod p$
$F_p \cdot (a \bmod p) \rightarrow m$

The circulant matrix A° defined by a° generates lattices with a very rigid structure. These lattices are called cyclic lattices. Their n basis vectors are obtained by repeatedly rotating the coefficients of a° by one position, i.e., by taking coefficients of the polynomials $ax^i \bmod (x^n - 1)$. There are concerns about the vulnerability of Ring-LWEs to attacks designed to take advantage of the structure in lattice implied by a polynomial.

C. Module-LWEs and the Kyber protocol

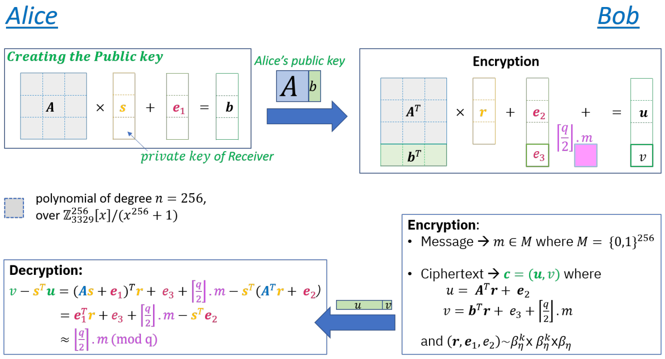
Module-LWE offers a trade-off between the security concerns and higher performance/efficiency of Ring-LWE, and

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

the converse attributes of plain-LWE [12]. The polynomials $\bar{s}, \bar{e_1}, \bar{r}$ and $\bar{e_2}$ in $\mathcal{R}_q$ in the Ring-LWE are replaced by a vector of k polynomials in $\mathcal{R}_q^k$. Figure 6 shows the key exchange algorithm for **Kyber.** Mathematically, the dimension of the A matrix is now $k \times k$, and its elements are degree n polynomials over $\mathbb{Z}_q$, i.e., the structure of A is $[\mathbb{Z}_q[x]/(x^n+1)]^{k \times k}$. Similarly, the vectors $\bar{s}, \bar{e_1}, \bar{r}$ and $\bar{e_2}$ are k-element arrays, each element being a q-ary n-degree polynomial, i.e., their structure is $[\mathbb{Z}_q[x]/(x^n+1)]^k$. As in FrodoKEM, $\bar{s}, \bar{e_1}, \bar{r}$ and $\bar{e_2}$ are sampled from a narrow distribution χ. For Kyber $k \in \{2, 3, 4\}$, $n = 256$, and $q = 3329$.

SABER, like Kyber is Module-LWE, often referred to as Module-LWR because the shared secret is determined by rounding [16]. As in NewHope, Alice receives the $\langle u, c \rangle$ pair from Bob, and uses the inner product of her shared secret s and u to develop a consensus set of bits K. The modulus q in SABER is a power of 2, and hence SABER can not deploy NTT and has to use other algorithms to speedup multiplications, however, the power of 2 moduls make the modular reductions easier.

Fig. 6. Module-LWE based key exchange in Kyber


Alice
Bob
Creating the Public key
$A \times s + e_1 = b$
private key of Receiver
Alice's public key
A b
Encryption
$A^T \times r + e_2 + = u$
b^T e_3 $\left\lceil\frac{q}{2}\right\rfloor.m$ v
polynomial of degree $n = 256$, over $\mathbb{Z}_{3329}^{256}[x]/(x^{256}+1)$
Encryption:
Message → $m \in M$ where $M = \{0,1\}^{256}$
Ciphertext → $c = (u, v)$ where
$u = A^T r + e_2$
$v = b^T r + e_3 + \left\lceil\frac{q}{2}\right\rfloor.m$
and $(r, e_1, e_2) \sim \beta_\eta^k \text{x}\, \beta_\eta^k \text{x} \beta_\eta$
u v
Decryption:
$v - s^T u = (As + e_1)^T r + e_3 + \left\lceil\frac{q}{2}\right\rfloor.m - s^T(A^T r + e_2)$
$= e_1^T r + e_3 + \left\lceil\frac{q}{2}\right\rfloor.m - s^T e_2$
$\approx \left\lceil\frac{q}{2}\right\rfloor.m \text{ (mod q)}$

D. Computational characteristics of Ring and Module LWE

The key-generation, encryption and decryption routines in Ring and Module LWE spend almost half their cycles in Number Theoretic transforms (NTT), and another approximately 15% of their cycles in SHA3/SHAKE256 to generate A, the basis for the lattice $[\mathbb{Z}_{3329}^{256}[x]/(x^{256}+1)]^{k \times k}$, for $k \in \{2, 3, 4\}$ [17]. In contrast, since FrodoKEM does not take advantage of NTT, and its larger lattice dimension requires more random numbers, almost all its cycles are spend in SHA3/SHAKE256.

The q in $\mathbb{Z}_q^n[x]/(x^n+1)$ or $\mathcal{R}_q$ in lattice based cryptographic is mostly a short integer (16 bits). The main performance challenges are to: 1) use the vector capabilities of modern many-core processors effectively to carry out the number theoretic transforms (or their inverse) needed for efficient polynomial multiplications; and 2) to generate the random numbers in $\mathbb{Z}_q$, or even smaller domains for error vectors and secrets efficiently. With the use of vector facility to perform NTT and inverse NTT, and optimization of SHA3 to the extent permissible, the time taken by the SHA3 component becomes almost 50% [17], making it a strong candidate for hardware acceleration.

IV. Code based PQC algorithms and McEliece Cryptosystem

In this section we will first lay down the basic principles of code based cryptosystem McEliece, described in [18]. The *private key* of the receiver is a set of three matrices $\{S, G, P\}$. S is an invertible $k \times k$ matrix, referred to as the scrambler matrix. G is a $[n, k, t]$ binary linear block code over $GF(2)$, capable of correcting upto t errors with the $n - k$ parity bits. (In [18], it is the Goppa code corresponding to irreducible polynomial of degree t' over $GF(2^m)$ and $t = m \cdot t'$). P is a $n \times n$ permutation matrix. The public key used by the sender is $G' = S \cdot G \cdot P$.

To encode a k bit vector M, the sender generates a n symbol error vector e of weight t, and outputs the cipher text r given by:

$$r = M \cdot G' + e$$

To decode the message r, the receiver first post multiplies it with P^{-1}, then decodes it using G to eliminate the error term and post multiplies the result with S^{-1} to recover M

$$r' = r \times P^{-1} = m \cdot S \cdot G + e'$$
$$m = \textbf{Decode}_G\,(r') \times S^{-1} \tag{5}$$

The KEM finalist classic McEliece [19] has features beyond the simple principles of Code Based Cryptography laid out in [18]. The additional features pertain to how the matrices G and G' are computed and represented in a compact manner. In particular, the G matrix is derived from a Vendermonde matrix. As shown in Figure 1, two of the alternates, BIKE and HQC, are also code based cryptosystems.

The ten parameter sets recommended in the NIST submission have parameters in the ranges $3488 \leq n \leq 8192$, $64 \leq t \leq 128$, $12 \leq m \leq 13$. The parameters have a strong positive correlation. One can readily observe that the key sizes are very large, and unlike lattice-based methods where the matrix A can be computed from an published seed, in McEliece the G' matrix has to be transmitted. Furthermore, the encryption and decryption operations involving G' and G respectively become expensive too.

V. PQC algorithm based on Elliptic Curve Isogeny (SIKE)

The section begins with the definition of algebraic structure for Elliptic Curves, and a class of elliptic curves called Montgomery curves. Then we describe the Diffie-Hellman, Elliptic Curve Diffie-Hellman (ECDH), and SIKE protocols, emphasizing the differences in the object manipulated or operation carried out at each step of the protocol. The next subsection defines the 'Isogeny object'. The section concludes with a description of the computational aspect of the point addition and point doubling on Elliptic curves.

In this paper we restrict our attention to Montgomery curves, the class used in SIKE protocol. The reader wishing to know more about other classes of elliptic curves, their use in other cryptographic protocols, and the mapping between the two curves of different classes, is refereed to two excellent texts [20], [21].

A. The algebraic structure of Elliptic Curves

An elliptic curve is an Abelian Group with a finite set of points (elements) and a group operation, referred to as point addition and denoted by $\boxplus$.

The points, or the set of elements in the group is

1) the identity element (point) of the group, or the point at ∞, (some times also denote it as $\mathcal{O}$)
2) the collection of all the points $P \in \mathcal{F}_q \times \mathcal{F}_q$ satisfying the equation

$$By^2 = x^3 + Ax^2 + x \tag{6}$$

The elliptic curve thus defined by equation 6, with $A, B \in \mathcal{F}_q$ is denoted as $EC_{AB}[\mathcal{F}_q]$. We will use the symbols P, Q, R etc., also represented as $(x_P, y_P), (x_Q, y_Q), \cdots$, to denote the elements of the group.

As a short hand we define a mapping *n-map* over points on an elliptic curve as the addition of the point to itself n times, i.e.,

$$\textit{n-map}[P] \equiv [n] \cdot P \equiv P \boxplus P \boxplus \cdots \boxplus P \tag{7}$$

The $\boxplus$ operator for two points P and Q on the elliptic curve is specified by rational functions, i.e. ratio of two polynomials over $(x_P, y_P), (x_Q, y_Q)$, thus satisfying the closure property. This axiomatic definition has the following geometric analog, which is helpful in deriving the rational function. For $P \neq Q$, and both $P, Q \neq \mathcal{O}$, let $\bar{R} = (x_{\bar{R}}, y_{\bar{R}})$ be the intersection of the chord through points P and Q with the curve of equation 6 in the Euclidean plane, and R be the reflection of $\bar{R}$ about the x-axis, modulo p. Then, $R = P \boxplus Q$ is given by

$$R = (x_{\bar{R}} \bmod p, -y_{\bar{R}} \bmod p) \tag{8}$$

Following the chord calculations we get the following rational functions for point addition of $P \boxplus Q$,

$$\begin{aligned} x_{P+Q} &= Bm^2 - (x_P + x_Q) - A \\ y_{P+Q} &= (2x_P + x_Q + A)m - Bm^3 - y_P \\ &= m(x_P - x_{P+Q}) - y_P \\ m &= \begin{cases} \frac{y_P - y_Q}{x_P - x_Q} \; for \; P \neq Q \; and P \neq (-Q) \\ \frac{3x_P^2 + 2Ax_P + 1}{2By_P} \; for \; P = Q \end{cases} \end{aligned} \tag{9}$$

The corner cases of $P = Q$ or $P = \mathcal{O}$ or $P = \mathcal{O}$, can be handled by a similar geometric approach [20], [21].

B. Elliptic curve Diffie-Hellman and isogeny based key encapsulation

Both Elliptic Curve Diffie-Hellman (ECDH) and isogeny based key encapsulation (SIKE) protocols can be explained in terms of the Diffie-Hellman key exchange as shown in Figure 7. The black labels in Figure 7A illustrate the Diffie-Hellman protocol, summarized below.

1) Both Alice and Bob start with an agreed upon (specified in standards) field $\mathcal{F}(p)$ and a base element g, and compute $(g^A \bmod p)$ and $(g^B \bmod p)$ respectively.
2) A and B are Alice and Bob's secrets, but $(g^A \bmod p)$ and $(g^B \bmod p)$ are exchanged over open channel, visible to adversary Eve.
3) Alice and Bob generate the same shared secret $(g^{AB} \bmod p)$ by raising the information they receive from each other, $(g^B \bmod p)$ and $(g^A \bmod p)$, to the power of secret keys A and B respectively.
4) The cryptographic strength of the protocol is rooted in Eve's inability to deduce A or B from $(g^A \bmod p)$ or $(g^B \bmod p)$, **the discrete log problem**.

The green labels in Figure 7A illustrate the Elliptic Curve Diffie-Hellman (**ECDH**) protocol. The protocol is similar to Diffie-Hellman, the differences being:

1) The starting information is an elliptic curve E and a starting point s on it.
2) The $g^A \bmod p$ and $g^B \bmod p$ operations are replaced by the *n-map* operation $[A] \cdot s$ and $[B] \cdot s$ on elliptic curve E.
3) The cryptographic strength of the protocol is rooted in Eve's inability to deduce A or B from $[A] \cdot s$ or $[B] \cdot s$ respectively.

Fig. 7. The Post Quantum Cryptography Key-Establishment Mechanism (KEM) algorithms advancing to the third round, the four finalists and three alternates

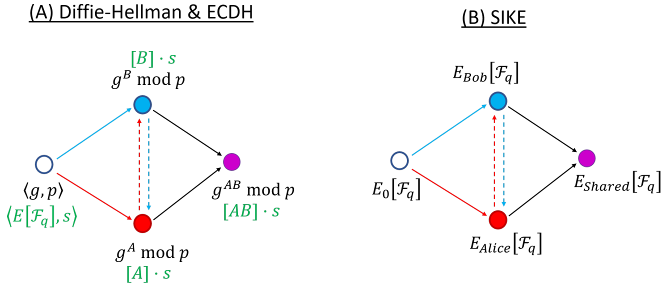


Figure 7B illustrates the Supersingular Isogeny based Key Encapsulation (**SIKE**) protocol. The protocol enables Alice and Bob to derive isomorphic elliptic curves $E_{SharedA}$ and $E_{SharedB}$ and the *j-invariant* of these two curve is their shared secret. The formula or *j-invariant* for a Montgomery elliptic curve is

$$j(E_a b \equiv by^2 = x^3 + ax^2 + x) = \frac{256(a^2-3)^3}{a^2-4} \tag{10}$$

The $j-invariants$ of two elliptic curves are same if and only if the two elliptic curves are isomorphic. The differences of SIKE from ECDH are

1) The information exchanged between Alice and Bob, and visible to Eve, are elliptic curves E_{Alice} and E_{Bob}. They are isogenies ϕ_{Alice} and ϕ_{Bob} of the starting

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

public curve E_0, dependent on secrets of Alice and Bob, s_A and s_B, respectively. Isogenies are defined in the next section, along with their method of computation in the SIKE proposal. The *j-invariant* of the elliptic curve depends only on the coefficient A of the elliptic curve of equation 6. A can can be retrieved from any three distinct points P, Q and R on the curve by the following formula. The elliptic curves in SIKE are specified/exchanged by defining the three $x-$coordinates.

$$a = \frac{1-(1-x_Px_Q-x_Px_R-x_Qx_R)^2}{4x_Px_Qx_R} - x_P - x_Q - x_R$$

2) Using E_{Bob} and E_{Alice}, and their secrets s_{Alice} and s_{Bob}, Alice and Bob derive the isogenies $E_{SharedA}$ and $E_{SharedB}$, two other curves which are isomorphic to each other, though unlikely to be the same. They compute the *j-invariant* of the shared curves to create the shared secret.
3) The cryptographic strength of the protocol is rooted in Eve's inability to deduce the secrets s_{Alice} or s_{Bob}, or the isogenies ϕ_{Alice} and ϕ_{Bob}, necessary to create the shared secret, from E_{Alice} and E_{Bob}.

C. Isogenies on/over Elliptic Curves

Isogeneies are non-constant, rational function maps from one Elliptic Curve, $EC_1[\mathcal{F}]$ to another Elliptical Curve $EC_2[\mathcal{F}]$ (the respective infinities of both curves included), while preserving the group structure (hence they are homorphism). If ϕ is an Isogeny, then

$$\begin{aligned} \phi : EC_1(\mathcal{F}) &\mapsto EC_2(\mathcal{F}) \\ \infty_{EC_1} &\mapsto \phi(\infty_{EC_1}) = \infty_{EC_2} \\ \forall R = P \boxplus Q \in EC_1 &\mapsto \phi(P) \boxplus \phi(Q) = \phi(R) \in EC_2 \end{aligned} \tag{11}$$

Note that the $\boxplus$ operators for EC_1 and EC_2 are different rational functions.

To compute their isogenies S_{Alice} and S_{Bob} from the common starting curve S_0, and their secrets s_A and s_B, Alice and Bob perform these two steps:

1) Using public parameters P_A, Q_A, P_B, and Q_B, Alice and Bob create their secret generators, $S_A = P_A + [s_A]Q_A$ and $S_B = P_B + [s_B]Q_B$ respectively, which are points on E_0.
2) Using published parameter e_2, Alice computes a sequence of e_2 2-isogenies to arrive at E_{Alice}. Bob similarly computes a sequence of e_3 3-isogenies to arrive at E_{Bob}. The formulae for generating the e_2 and e_3 are well known. E_{Alice} and E_{Bob} are isogenies of E_0.
3) Note that p in $\mathcal{F}_{p^2}$ is related to e_2 and e_3 as

$$p = e^2\, e^3 - 1 \tag{12}$$

D. Computational aspects of elliptic curve point operations

Note that the coordinates of a point P on the elliptic curve are elements of $\mathcal{F}_{P^2}$, i.e., $x_P, y_P \in \mathcal{F}_{P^2}$. The operations on the points on the elliptic curve, for example the operation $P_A \boxplus Q_A$, are defined/implemented in terms of $\mathcal{F}_{p^2}$ operations of add, multiply and inverses. The $\mathcal{F}_{p^2}$ operations are in turn defined/implemented as operations on $\mathcal{F}_p$. The implementation of $\mathcal{F}_{p^2}$ operations in terms of $\mathcal{F}_p$ operations is analogous to implementation of operations in $\mathbb{C}$ in terms of operations in $\mathbb{R}$. Finally, operations in $\mathcal{F}_p$ are arithmetic on few hundred bit wide integers implemented in 64 or 32 bit arithmetic instructions in software, often using algorithmic approaches to reduce the operation count. This recursive simplification of arithmetic on points on elliptic curves is handy in understanding the software implementations.

The proposed SIKE standard defines four public parameter sets, each with a different value of $\langle e_2, e_3 \rangle$, and hence a different p. (The parameter sets have additional public information such as the starting curve E_0 and various points on the curve needed by the protocol.) Table I below lists the value e_2 and e_3, and the number of bits in the resulting p. The larger values of p correspond to stronger security.

TABLE I
THE PRIMES p IN THE FOUR PARAMETER SETS. $p = 2^{e_2}3^{e_3} - 1$

Parameter Set	e_2	e_3	**bits in** p
SIKEp434	216	137	434
SIKEp503	250	159	503
SIKEp610	305	192	610
SIKEp751	372	239	751

E. Background literature on Isogeny based cryptography

Foundations of Elliptic Curve based cryptography were put down by Victor Miller in his seminal paper [22]. That became the stepping stone for both current elliptic curve based schemes like ECDH/ECDSA (Elliptic Curve Diffie Hellman and Elliptic Curve Digital Signature Algorithm), and hence also for the Post Quantum variant in isogeny based key exchange.

Costello has an elegant introduction to supersingular elliptic curves isogeny based cryptography protocols [23]. He explains the protocol laid out in section V-B using curves over $\mathcal{F}_{p^2}$ with $p = 2^4 3^3 - 1 = 431$. The key concept to take from this paper is the isogeny graph in which each node is the collection of elliptic curves in $\mathcal{F}_{p^2}$ with the same j-invariant. The edges in the graph represent isogenies of degree 2 or 3. In other words, the graph has two types of edges, isogeny 2 edges and isogeny 3 edges.

The above toy example is used to illustrate computation of Alice's (or Bob's) isogeny ϕ_A (or ϕ_B) as a composition of a sequence smaller isogenies. In Alice's case e_2 two isogenies are composed to arrive at ϕ_A. In Bob's case e_3 three isogenies are composed to arrive at ϕ_B. The composition of isogenies corresponds to traversing the edges of the isogeny graph. The example illustrates the mapping of a common starting elliptic curve E_0 to elliptic curves E_A and E_B respectively, as a composition of e_2 2-isogenies and e_3 3-isogenies respectively. E_A and E_B are then mapped by Bob and Alice respectively, using their secret keys, to E_{AB} and E_{BA} which will be

978-1-7281-9219-2/20/$31.00 ©2020 IEEE

isomorphic, and hence have identical *j-invariant*, their shared secret.

Costello and Smith elaborate on the mapping of points on an elliptic curve E over $\mathcal{F}_{P^2}$ to $\mathbb{P}^1$ [24], building upon Montgomery [25]. They further present the point addition, point doubling and ladder algorithms in terms of operations in $\mathcal{F}_p$. Algorithms 1-4 in [24] are codified, as is, in the Optimized implementation submitted to NIST.

Implementation specific details, particularly the details relevant to the performance of the SIKE Reference and Optimized implementations can be found in [26] and [27].

VI. Comparative advantages of various PQC protocols families

The main attributes of a cryptography protocol considered in selecting one from a pool are the cryptographic robustness, computational costs, and key length. Cryptographic robustness is the ability to withstand cryptanalytic attacks, which is either proved by reducing the algorithm to a known (and typically conjectured) hard problem, or demonstrated by a long history of no successful attacks. In Figure 8 we highlight these attributes for the protocols discussed in this paper. The key points are:

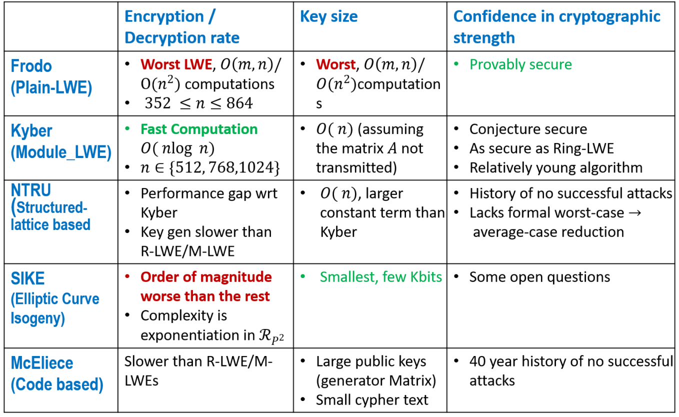
- Kyber, and its companion Module-LWE finalist SABER, have attractive key-length and computational costs, but compared to some other protocols, they have not yet withstood the multiple decades of cryptanalytic attacks to establish their robustness. Hence, they may be more suitable for high volume client-server transactions that do not have strong forward secrecy requirements, particularly when the sessions established with the share secret are short lived. An example would be securing credit card transactions.
- Conversely, McEliece has larger keys and slower computational speeds, but has resisted crypto-attacks for four decades. Similarly, among the LWE algorithms, FrodoKEM has least amount of structure and hence believed to be less vulnerable to attacks than its Module-LWE counterparts Kyber and SABER. Applications with strong forward secrecy requirements, for example in health-care, or financial industry, or data backup applications where the overhead of establishing a shared secret is amortized over longer messages being encrypted, could favor Mceliece or FrodoKEM over Module-LWEs.
- SIKE has the smallest public key sizes and temporary storage requirements, and hence may be more suitable for low volume key exchanges with edge devices with limited storage.

At the end of round 3 NIST intends to standardize one of the four finalists, the blue entries in Figure 1. Beyond that in round 4 it also plans to standardize algorithms for specific use cases from the alternates.

VII. Summary

Most of the PQC techniques have been around for two decades or so. However, the performance of the lattice-based algorithms has improved significantly by advances like learning with errors (LWE), and employing polynomial rings. This, coupled with the advances in quantum computing and its consequent threat to the workhorses of cryptography, RSA, Diffie-Hellman, elliptic curve Diffie-Hellman, etc., PQC has become important.

Fig. 8. Comparative advantages of various PQC protocols



	Encryption / Decryption rate	Key size	Confidence in cryptographic strength
Frodo (Plain-LWE)	• **Worst LWE**, $O(m,n)$/ $O(n^2)$ computations • $352 \le n \le 864$	• **Worst**, $O(m,n)$/ $O(n^2)$computation s	• Provably secure
Kyber (Module_LWE)	• **Fast Computation** $O(n\log n)$ • $n \in \{512, 768,1024\}$	• $O(n)$ (assuming the matrix A not transmitted)	• Conjecture secure • As secure as Ring-LWE • Relatively young algorithm
NTRU (Structured-lattice based	• Performance gap wrt Kyber • Key gen slower than R-LWE/M-LWE	• $O(n)$, larger constant term than Kyber	• History of no successful attacks • Lacks formal worst-case → average-case reduction
SIKE (Elliptic Curve Isogeny)	• **Order of magnitude worse than the rest** • Complexity is exponentiation in $\mathcal{R}_{p2}$	• Smallest, few Kbits	• Some open questions
McEliece (Code based)	Slower than R-LWE/M-LWEs	• Large public keys (generator Matrix) • Small cypher text	• 40 year history of no successful attacks

There is vast amount of literature on the cryptographic aspects of PQC algorithms. In this paper, first of its kind to the best of our knowledge, we have focused on the algorithmic aspects PQC, compared the similarity in computational characteristics of of lattice-based PQC algorithms, and analyzed the efficacy and limitations of modern many-core processors in executing them.

Acknowledgments

The authors would like to thank Camille Bossut, K. Ekanadham, Sandhya Koteshwara, Camille Bossut and Rajat Rao.

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303 – 332, 1999.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[3] (2018) Cryptography in a postquantum world. url-https://www.accenture.com/_acnmedia/PDF87/Accenture-809668QuantumCryptographyWhitepaperv05.pdf. (Last accessed Aug 1, 2020).

[4] K. C. Miao, J. P. Blanton, C. P. Anderson, A. Bourassa, A. L. Crook, G. Wolfowicz, H. Abe, T. Ohshima, and D. D. Awschalom, "Universal coherence protection in a solid-state spin qubit," *Science*, 2020. [Online]. Available: https://science.sciencemag.org/content/early/2020/08/12/science.abc5186

[5] (July 22, 2020) PQC Standardization Process. url=https://csrc.nist.gov/News/2020/pqcthirdroundcandidate-announcement. (Last accessed Aug 1, 2020).

[6] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone. (July 22, 2020) Status report on the second round of the nist post-quantum cryptography standardization process. url=https://csrc.nist.gov/publications/detail/nistir/8309/final. (Last accessed Aug 1, 2020).

[7] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '05. New York, NY, USA: ACM, 2005, pp. 84–93. [Online]. Available: http://doi.acm.org/10.1145/1060590.1060603

[8] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996, pp. 99–108.

[9] ——, "Generating hard instances of lattice problems," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 3, no. 7, 1996. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/1996/TR96-007/index.html

[10] C. Peikert, "A decade of lattice cryptography," *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 4, pp. 283–424, 2016. [Online]. Available: http://dx.doi.org/10.1561/0400000074

[11] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! practical, quantum-secure key exchange from LWE," *IACR Cryptology ePrint Archive*, vol. 2016, p. 659, 2016.

[12] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," in *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, April 2018, pp. 353–367.

[13] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum Key Exchange—A New Hope," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 327–343. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[14] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory*, J. P. Buhler, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 267–288.

[15] C. Peikert, "Lattice Cryptography for the Internet," in *Post-Quantum Cryptography. PQCrypto 2014. Lecture Notes in Computer Science)*, M. M., Ed., vol. 8772. Springer, 2014.

[16] J.-P. D'Anvers, A. Karmakar, S. Sinha Roy, and F. Vercauteren, "SABER: Module-LWR based key exchange, cpa-secure encryption and CCA-Secure KEM," in *Progress in Cryptology – AFRICACRYPT 2018*, A. Joux, A. Nitaj, and T. Rachidi, Eds. Cham: Springer International Publishing, 2018, pp. 282–305.

[17] S. Koteshwara, M. Kumar, and P. Pattnaik, "Performance optimization of lattice post-quantum cryptographic algorithms on many-core processors," in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2020.

[18] R. McEliece. (1978) A public–key cryptosystem based on algebraic coding theory. url"https://tmo.jpl.nasa.gov/progress_report2/42 – 44/44N.PDF". (Last accessed Aug 24, 2020).

[19] D. Bernstein, T. Chou, T. Lange, I. Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, and W. Wang. (2019) Classic McEliece: conservative code-based cryptography. urlhttps://classic.mceliece.org/nist/mceliece-20190331.pdf. (Last accessed Aug 24, 2020).

[20] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. 2004 Springer-Verlag, 2004.

[21] L. C. Washington, *Elliptic Curves Number Theory and Cryptography*. Chapman and Hall/CRC, 2008.

[22] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology — CRYPTO '85 Proceedings*, H. C. Williams, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426.

[23] C. Costello, "Supersingular isogeny key exchange for beginners," in *Selected Areas in Cryptography – SAC 2019*, K. G. Paterson and D. Stebila, Eds. Cham: Springer International Publishing, 2020, pp. 21–50.

[24] C. Costello and B. Smith, "Montgomery curves and their arithmetic," *J Cryptogr*, vol. 8, pp. 227–240, 2018. [Online]. Available: https://doi.org/10.1007/s13389-017-0157-6

[25] P. L. Montgomery, "Speeding the pollard and elliptic curve methods of factorization," *J Cryptogr*, vol. 48, pp. 243–264, 1987. [Online]. Available: https://doi.org/10.1090/S0025-5718-1987-0866113-7

[26] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Post-Quantum Cryptography. PQCrypto 2011. Lecture Notes in Computer Science, vol 7071*, Y. BY., Ed. Springer Berlin Heidelberg, 2011.

[27] L. D. Feo, D. Jao, and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," *Journal of Mathematical Cryptology*, vol. 8, no. 3, pp. 209 – 247, 2014. [Online]. Available: https://www.degruyter.com/view/journals/jmc/8/3/article-p209.xml

978-1-7281-9219-2/20/$31.00 ©2020 IEEE