Investigation on NIST post-quantum lattice-based encryption schemes

Giacomo Mazzola[1] , **Kaitai Liang**[1] , **Huanhuan Chen**[1]
[1]TU Delft

Abstract

In the last decade, development in quantum computing has threatened the security of current public-key cryptography. For this reason, the American National Institute of Standards and Technology (NIST) has organized a competition-like process to standardize new quantum-resistant public-key encryption and digital signature schemes.
This research project aims to give a general overview of post-quantum lattice-based cryptography and analyze and compare the submitted lattice-based public-key encryption schemes over performance, security, distinguishing features and potential vulnerabilities.

1 Introduction

In the last decade, research in quantum computers has developed significantly, to the point that big companies, such as Google and IBM, claimed their quantum supremacy. If a big enough quantum computer is ever built, it could seriously threaten the security of many of today's public-key cryptosystems. Thanks to Shor's algorithm [1], it would solve mathematical problems, such as integer factorization, discrete logarithms and elliptic curves, in polynomial time. Therefore, encryption, key establishment, and digital signature schemes that rely on these computational problems will be severely affected.

In response to this problem, the American National Institute of Standards and Technology (NIST) has organized a competition-like process to standardize new encryption and digital signature schemes, which are robust against both classical and quantum attacks. The first round of the competition started in 2017 when 69 candidates, including public-key encryption (PKE) and digital signature (DS) schemes, were submitted. These schemes were analyzed thoroughly by both NIST and public peer reviewers, which assessed theoretical and empirical security, performance, and risk factors. At the time of writing, seven finalist schemes and eight alternates made it to the third and final round. The candidates and reports for each round can be found here [2] [3] [4].

The cryptographic schemes submitted to the competition can be divided into five main categories with respect to the mathematical problem they are based on. As can be seen in [5], these categories are: (1) lattice-based, (2) code-based, (3) multivariate (4) hash-based, (5) Supersingular Isogeny-based. This research project aims to present to the reader a general overview of the state-of-the-art post-quantum lattice-based encryption and key encapsulation systems. In addition to a detailed background into this subject, this paper also offers an analysis and comparison of the lattice-based encryption schemes admitted to the second round of the NIST competition. Particular focus will be paid to (1) theoretical security against both classical and quantum attacks, (2) theoretical and practical level of cost, especially runtime and bandwidth performance, (3) computation/distinguishing features, (4) potential vulnerabilities and shortages (5) overall complexity of the schemes and their implementations.

Note that round 3 of the NIST competition is happening at the time of writing. Therefore, the analysis is based on the submissions in round 2. In the discussion section, a paragraph is dedicated to the current state of the competition.

The paper is structured as follows. Background knowledge is given in section 2. In section 3, the method is explained. A summary of each analyzed scheme is given in section 4. Theoretical and practical performance results are shown and briefly discussed in section 5. In section 6, the security of the schemes is analysed. An ethical discussion is presented in section 7. The results of the research project are discussed in section 8. Finally, section 9 contains the conclusions and future work.

2 Background

Lattice-based cryptography is based on lattices, which are algebraic abstract structures that consist of an ordered set of points with a periodic structure in an n-dimension space. They can be defined by a set of n vectors $B = (b_1, b_2, ..., b_n)$, denominated as the basis of the lattice (see Figure 1). Some computationally hard problems can be defined on lattices:

$\boldsymbol{SVP}$**:** given a set of basis B of an n-dimensional lattice L, find the shortest non-zero vector in the lattice. An approximated version also exists: SVP_γ: given an

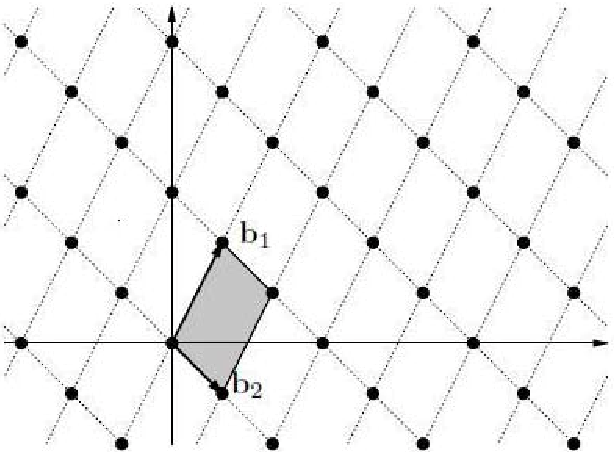

Figure 1: A 2-dimensional lattice generated by the basis $B = (b_1, b_2)$. (image taken from [6])

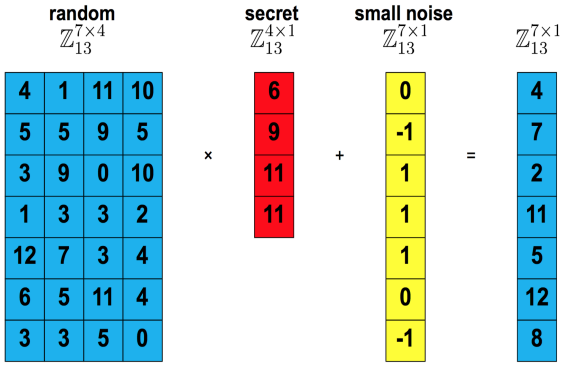

Figure 2: Image taken from [8]

approximation factor $\gamma >= 1$, find a solution v such that $||v|| <= \gamma * \lambda_1(L)$, where $||v||$ is the euclidean length of the vector v, and $\lambda_1(L) = min_{v \in L \setminus 0}||x||$ is the minimal distance of the lattice L, which is the length of the shortest non-zero vector in L .

CVP **:** given a set of basis B of an n-dimensional lattice L and a target point x, find the lattice point that is closest to the target. An approximated version also exists: CVP_γ: given an approximation factor $\gamma >= 1$, find a solution v such that $||v - x|| <= \gamma * dist(x, L)$.

Most of the submitted schemes are based on the Regev's Learning with Errors (LWE) problem [7] or its variants: Module Learning with Errors (MLWE) and Ring Learning with Errors (RLWE). Their security can be proven by finding a theoretical reduction to a computationally hard lattice problem.

Informally, LWE can be seen as, given a sequence of approximate random linear equations on s in $\mathbb{Z}_q^n$, find s. A more appropriate definition of the problem is given below.

LWE **:** given a dimension n, a modulus q and an error distribution χ over $\mathbb{Z}$, construct m samples of the form ($\boldsymbol{a}$, b = $\langle \boldsymbol{a}, \boldsymbol{s} \rangle$ + e mod q) where $\boldsymbol{s} \in \mathbb{Z}_q^n$, $\boldsymbol{a} \in \mathbb{Z}_q^n$ chosen uniformly at random and an integer error e $\in \mathbb{Z}$ sampled using χ (See Image 2 for a visual representation of a LWE instance). We can define two versions of the LWE problem: *search*, which asks to recover the secret $\boldsymbol{s}$ and *decision*, which asks to recognize samples taken from the LWE distribution from uniformly sampled ones.

One downside of cryptographic schemes based on LWE is that they require quite large keys, usually in the order of n^2, and therefore, their runtime performance is also affected. One way to reduce the size of the keys is by assuming that the lattice has a specific structure. This can be achieved by interchanging the group $\mathbb{Z}_q^n$ with a ring, such as the quotient polynomial ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$, where $\mathbb{Z}_q[x]$ is the set of all polynomials that have coefficients in Z_q. One can then sample instances in (R_q^k, R_q) where k is the rank of the lattice. If $k = 1$, then we call the problem Ring-LWE, if $k >= 2$ we call it Module-LWE. Thanks to the additional structure, MLWE and RLWE feature better performance than LWE, but they are theoretically more vulnerable to attacks that could exploit the extra structure. Even though no attack that exploits the additional structure has been found so far, the cryptographic community believes that these schemes need to be further studied. MLWE can be considered as a trade-off between the security of LWE and the high performance of RLWE [9]. MLWE also features incredible flexibility; in fact, to increase security, it is enough to change the rank k without affecting the underlying ring R_q or any arithmetical operation defined on it.

Another variant of LWE is Learning with rounding (LWR). It differs from LWE in the way the equations are approximated. While in LWE a small error is added, in LWR they get rounded to a smaller modulo. While maintaining the same level of security, LWR features two advantages with respect to LWE: it is more compact since, by rounding, the size of the keys and the ciphertext decreases, and it is more simple and efficient since it does not need to sample errors from a distribution. The Module and Ring variants can also be applied to the LWR assumption, respectively MLWR and RLWR.

An alternative to LWE/R and its variants is NTRU (Nth degree-truncated polynomial ring units). NTRU was first proposed in 1998 by Hoffstein, Pipher and Silverman [10] as a public-key cryptosystem, with the name of NTRUEncrypt (see appendix A for a description of the scheme). It is based on factorization of polynomials over the ring $R = Z_q[x]/\phi_1 * \phi_n$ where $\phi_1 = (x - 1)$ and $\phi_n = (x^n - 1)/(x - 1) = x^{n-1} + x^{n-2} + ... + 1$.

NTRU, unlike LWE/R, lacks an average to worse case reduction; however, thanks to its long history of cryptoanalysis, the cryptographic community lays trust in it.

According to Peikert [11], lattice-based cryptography is the most promising alternative to nowadays public-key cryptography for the following reasons: (1) it is efficient, bandwidth requirements are fairly small and runtime performance is even better than classical schemes (e.g., Diffie-Hellman). (2) So far, it has been proven resistant against quantum attacks. (3) Its security (except for NTRU) is based on formal average-case to worst-case reductions, which guarantee that it is computationally hard on average instances. (4) It can be used to implement different schemes, such that encryption and digital signatures, and it also provides solutions for the 'holy grail' problems, such as fully homomorphic encryption.[1]

3 Method

During the first stage of the research, for approximately 1-2 weeks, a thorough literature study has been conducted to acquire a basic understanding of the mathematical concepts behind lattice-based encryption. Research has also been done on the state of the art post-quantum cryptography. Useful information can be found in NIST round reports [2][3][4] and in [5]. Furthermore, in this survey [**lattice˙survey˙peikert**], Peikert gives an overview of the lattice-based encryption development in the last decade.

Subsequently, the submitted schemes were analyzed. This was done firstly by reading the reference papers submitted to the NIST competition and secondly by reading reviews and comments published by the peer review community. Particular attention was paid to the performance, security and distinguishing features of the schemes. Finally, After the theoretical analysis, a practical analysis has been conducted. In fact, the codebases of the schemes were cloned and run all on the same machine to obtain data for a fair comparison.

4 Summary of Analysed Schemes

This section presents an analysis for each of the considered schemes. For every scheme is given an insight into the original problem they are based on and how they differ from it, in addition to performance, security and general features. seven schemes out of nine are based on LWE/R or their variants; therefore, it is worth dedicating a paragraph to explain the common features.

4.1 LWE/R based schemes:

All LWE/R based schemes admitted to the second round of the competition are key encapsulation mechanisms (KEMs). They are all based on an IND-CPA PKE scheme which is generally part of the submission, and they all achieve IND-CCA security by applying slight modifications of the Fujisaki-Okamoto (FO) transformation [12] (see appendix B for a detailed explanation of IND-CPA and IND-CCA). The main idea behind this transform is to check if the ciphertext is valid by re-encrypting it after decryption and accept it only if it is valid. This procedure obviously affects the performance of the schemes, but it gives them higher security. In order for the FO to produce a correct IND-CCA KEM, the IND-CPA PKE needs to have a negligible failure rate, approximately lower than 2^{128} [13].

For every scheme, unless otherwise specified, three parameter sets are proposed. These parameter sets reach level 1, level 3, and level 5 security specified by NIST.[2]

The schemes differ mainly on the choice of the underlying ring, the dimensions of the moduli, the error correcting codes, and the distribution used to sample errors and secrets.

4.1.1 FrodoKEM

FrodoKEM [14] is a suite of KEMs based on the LWE problem. It differs from the original scheme proposed by Regev [7] on multiple points, such as: $\boldsymbol{A}$ is a square $N \times N$ matrix instead of a long rectangular one, and it is pseudorandomly generated from a small seed. These modifications allow FrodoKEM to use smaller keys and ciphertexts. New sets of parameters are also suggested. The submission proposes three sets of parameters that differ on the security level: FrodoKEM-640 (level 1), FrodoKEM-976 (level 3), and FodoKEM-1344 (level 5). The three versions differ mainly on the ring and the modulo q, which is a power of 2 such that $q <= 2^{16}$, and the standard deviation of the Rounded Gaussian distribution. Moreover, for every security level, two implementations are provided: AES, which uses AES128 to generate the matrix $\boldsymbol{A}$, and SHAKE, which uses SHAKE128. Generally, SHAKE offers better performance, whereas AES is particularly suitable on platforms that provide hardware for AES acceleration (such as AES-NI on Intel platforms).

The general design idea behind FrodoKEM is: "a conservative approach that errs comfortably on the side of security and simplicity over performance and (premature) optimization." [14]. Even though the creators claim that the performance of FrodoKEM is acceptable for most of nowadays usages, it is significantly less efficient than the schemes based on algebraically structured versions of LWE. Specifically, the bandwidth usage and the runtime are one order of magnitude bigger than the other schemes.

The performance loss is a direct disadvantage of choosing LWE. However, there are some significant advantages as well. First of all, the scheme is really simple (e.g., there is no need to implement the complicated NTT[3] transform since the main operations are simple matrix-vector products). Secondly, it is theoretically more secure than the other schemes

[1] Fully homomorphic encryption is a form of encryption that allows a user to make computations on encrypted data, without first decrypting it, eventually having the same result that would be produced if the operations were computed on decrypted data.

[2] Level 1, level 3 and level 5 are defined by matching or exceeding the brute-force security of respectively AES-128, AES-192 and AES-256.

[3] Number Theoretic Transform: it is the Discrete Fourier Transform over a ring, it allows to store elements of the ring in a compact way and to multiply them efficiently.

because it has less structure that an attacker could exploit. Even though no attacks that exploit the extra structure have been found so far, the more robust theoretical security is the reason why FrodoKEM was selected as an alternate scheme. To cite NIST words: "FrodoKEM could also serve as a conservative backup in the case of new cryptanalytic results targeting structured lattices being discovered in the third round." [3].

4.1.2 NewHope

NewHope [15] is a family of KEMs that rely on the RLWE assumption. It is based on the polynomial ring $R = Z_q[x]/(x^n + 1)$, which is a power of 2 cyclotomic ring. The submission includes only two parameter sets: NewHope512 (level 1) and NewHope1024 (level 5), because, in order to implement the NTT efficiently, the size of the Ring n has to be a power of 2, and the following relation between n and the modulo q must hold: $q \equiv 1 \mod 2n$. Therefore, it is impossible to achieve any intermediate security level.

NewHope is based on the scheme described by Lyubashevsky, Peikert and Regev [16]. The modulo q is the same for both implementations and is chosen to be as little as possible since the security level grows with the noise-to-modulo ratio; the smallest modulo that satisfies all the constraints is $q = 12289$. The secret s and the error e are both sampled from the same centered binomial distribution, which in practice is easier to implement than a Gaussian distribution and is less vulnerable against timing attacks.

NewHope achieves excellent performance both for bandwidth and runtime. However, it is fairly complex due to the implementation of the NTT and the reconciliation mechanism. Furthermore, unlike MLWE/R schemes, RLWE based schemes do not feature flexibility in changing the security level; in fact, to increase security, the underlying ring must be altered. For these reasons, NIST developed a slight preference over MLWE schemes and decided not to admit NewHope to the third and final round.

4.1.3 Kyber

Kyber [17] is a suite of KEMs based on the MLWE problem. In the submission, three parameter sets are specified: Kyber512 (level 1), Kyber768 (level 3) and Kyber1024 (level 5). In addition, for every parameter set, an optimized version "90s" is provided. The parameter $n = 256$ is the same for all the variants because they need to encapsulate keys with 256 bits of entropy (encrypt a plaintext 256 bits long). Also, the modulo $q = 3329$ is the same for all the variants. It was chosen to be as small as possible, satisfying the condition $n|(q-1)$ to implement the NTT efficiently and have negligible failure probability. The scalar k, used to fix the lattice dimension to a multiple of n, is the only parameter that can be changed and is, therefore, the parameter responsible for changing the security level.

Kyber is based on the module R_q^k where R is the same ring used by NewHope. Kyber also uses the same centered binomial distribution to sample secrets and errors.

Kyber can be seen as a trade-off between efficient RLWE schemes and secure LWE ones; the parameter sets proposed in the submission witness less structure than RLWE and similar performance, being more scalable. In fact, being based on MLWE assumption, in order to vary the hardness, and therefore the security, it is enough to vary the dimension of the module without changing the underlying ring structure.

Kyber was selected to be a finalist and it is considered to be a likely candidate for standardization.

4.1.4 Saber

Saber [18] is a suite of KEMs based on the MLWR problem. It differs from MLWE by applying a second smaller modulo reduction instead of adding a small random error. For this reason, MLWR schemes halve the randomness required and reduce bandwidth usage. The Module structure gives it the same flexibility as MLWE schemes, such as Kyber. The submission contains three different parameter sets: LightSaber (level 1), Saber (level 3), FireSaber (level 5).

Saber has excellent performance both for bandwidth and runtime. The MLWR structure allows it to reduce bandwidth and, thanks to the choice of using power of 2 moduli, modular reductions can be computed efficiently by using bit-masking operations, but it makes it impossible to implement the NTT. However, this is not a severe problem because the scheme only requires multiplications between random elements from R_q and small elements from the same ring. Moreover, since the small elements have bounded coefficients in absolute value, it is possible to implement multiplication efficiently by simple circular shifts and additions. The overall performance is comparable to the ones of other structured LWE schemes. Thanks to its high performance, simplicity and flexibility, Saber was admitted to the final round.

4.1.5 LAC

LAC [19] is a family of cryptographic algorithms based on the RLWE assumption. The submission provides three sets of parameters for the KEM scheme: LAC-128 (level 1), LAC-192 (level 3), and LAC-256 (level 5).

The principle design idea is to keep the keys and ciphertext sizes as small as possible. Therefore, to reduce the bandwidth requirements, the byte modulus $q = 251$ was chosen. As a consequence, it is not possible to implement the NTT to compute operations between polynomials efficiently; however, Intel Advanced Vector Extensions2 (AVX2) can be used to improve the efficiency of the computations by parallelizing multiple multiplication operations in one instruction cycle. A consequence of using a small modulo is that the error rate increases. Therefore, LAC uses an error correction code, such as BCH, to guarantee correctness.
LAC uses a similar cyclotomic ring as NewHope, and it also samples secrets and errors from a narrow centered binomial distribution.

During the first and the second round, some security is-

sues were discovered. Despite the designers modified LAC to resist attacks that could exploit these issues, NIST believes that LAC still needs to be studied further before it can be considered for standardization. Therefore, it was not admitted to the final round.

4.1.6 Three Bears

Three Bears [20] is a family of KEMs based on the Kyber MLWE implementation with a major difference: the polynomial ring is replaced with the integers modulo a generalized Mersenne number[4], making it Integer Module Learning With Errors (I-MLWE). The submission contains three parameter sets that achieve different levels of security: BabyBear (level 2), MamaBear (level 3) and PapaBear (level 5).

The creators' goal is to explore some less studied variants of the LWE problem that might have a good potential, such as I-MLWE. The design choice to store the private key as a seed is motivated by the fact that the key generation is very fast and, therefore, it is worth saving on space. As a consequence, the private key is only 40 bytes long.

As with the others RLWE and MLWE schemes, Three Bears is characterized by excellent bandwidth and runtime performance. However, both the I-MLWE problem and Three Bears have not received much attention and have not been thoroughly studied by the community; therefore, NIST decided not to admit it to the final round despite its qualities.

4.1.7 Round5

Round5 [21] is a family of KEMs based on (Generalized) LWR problem. It is a merger of the round one submissions Round2 [22] and HILA5 [23]. The most particular feature of Round5 is the unified design that allows it to be instantiated as both a LWR or a RLWR problem depending on the input parameters. This features allows Round5 to be flexible and particularly suitable for more applications: instantiated as a LWR, it suites applications that don't have strict performance requirements but need high security, while instantiated as a RLWR problem, it suites for applications with opposite requirements. Round5 uses $\phi = x^n + ... + x + 1$, with $n+1$ a prime, as a reduction polynomial because it allows to choose n from a wide range to achieve different security levels.

Round5 is characterized by a great bandwidth and runtime performance. The rounding leads to faster execution time since no errors has to be sampled and reduces the bandwith requirements. The moduli p and q are powers of 2, therefore rounding is computed efficiently by simple bit-masking operations. It derives XEf error correcting code from HILA5 to reduce the failure probability.

The submission is composed by 18 parameter sets: six ring parameter sets without error correction, six ring parameter sets with error correction, and six non-ring parameter sets. Each parameter set contains IND-CPA and IND-CCA version of security levels 1, 3, and 5. In this paper are analyzed the level 5 IND-CCA variants for the ring version with error correcting code (R5ND_5CCA_5d) and the non ring version (R5N1_5CCA_0d).

Even though Round5 achieves great levels of performance, it is not enough to compensate its complexity. Therefore, NIST decided to focus its attention to other structured-lattice schemes and not to admit Round5 to the final round.

4.2 NTRU

Only two KEMs based on the NTRU assumption have been submitted to the NIST competition: NTRU [24] and NTRUPrime [25]. The idea behind NTRU assumption is the hardness of factoring polynomials, in a truncated polynomial ring, into a quotient of two small coefficients polynomials.

Both KEMs are perfectly correct, which means that decryption is guaranteed to always be correct, and they both feature fast encryption and decryption routines since the main operations are efficient polynomial multiplications. They both achieve IND-CCA security by applying a version of the FO transform.

Many versions of NTRU have been designed in the past decades. Appendix C shows the genealogical tree of such schemes.

4.2.1 NTRU

NTRU [24] is a family of KEMs based on the NTRU assumption. The NTRU round 2 submission is based on the first round's NTRUEncrpypt and NTRU-HRSS-KEM submissions [2]. Four parameter sets for NTRU are proposed: ntruhrss701, ntruhps2048509, ntruhps2048677 and ntruhps4096821. The creators provide two different models to assess the theoretical security level: a non-local model, which is similar to the ways other schemes assess security, and a local model, a more aggressive one. NTRU lacks a level 5 scheme in the non-local model. NIST declared that these models need to be further studied in order to find a consensus.

NTRU features fast encapsulation and decapsulation routines, but the key generation is significantly slower than the RLWE and MLWE schemes because it requires polynomial division; for this reason, it is harder to achieve forward security in ephemeral encryption systems.

Even though it lacks a formal worst-case-to-average-case reduction, NTRU has been extensively studied during its long history, and other organizations have even standardized some of its versions. For this reason, NIST developed a strong preference for NTRU and admitted it to the final round.

4.2.2 NTRUPrime

NTRU Prime is a family of KEMs composed by "Streamlined NTRU Prime" and "NTRU LPRime". The idea behind NTRU Prime is to reduce significantly the attack surface in exchange to a slight loss in performance. The main difference with respect to the other submissions is

[4]Marsenne number: a prime number that is one less than a power of two ($2^n - 1$).

the choice of the ring. NTRU Prime in fact substitutes the commonly used ring $R = Z_q[x]/(x^n \pm 1)$ with the field $F = Z_q[x]/(x^n - x - 1)$, where n is prime. This change reduces the ring homomorphism available to an attacker.

The main differences between the two KEMs are the algebraic structure to compute the public key and the way they add noise to the ciphertext. Streamlined NTRU is denominated as a "quotient rounded NTRU' scheme. It follows the classic NTRU key generation procedure with a small change: instead of computing the public key as $h = 3 * g/f$ computes $h = g/3 * f$. This new design choice brings two advantages: slightly better performance obtained by skipping computing h^{-1} in decryption routine and less space for storing the key pair. This scheme is called "rounded" because it adds noise to the ciphertext by computing $h * r$ and rounding every coefficient to the nearest multiple of 3. The "rounded NTRU" has two advantages over "noisy NTRU": first of all, it simplifies protection against chosen-ciphertext attacks because the message m is directly determined by r; secondly, thanks to the fact that the ciphertext is rounded, it requires smaller bandwidth. NTRU LPRime is denominated a "product noisy NTRU' scheme. It is called "product" because the public key is generate as $h = d + a * G$, where a and d are secret small polynomials and G is a public element in R_q. As for the noise, it follows the original design of choosing m at random.

NTRU Prime proposes three sets of parameters, respectively in the security classes level 2, level 3 and level 4. NIST advanced NTRU Prime to the final round as an alternate finalist because of the different choice for the ring, which could be an alternative to the cyclotomic ring used by most of the other schemes. However, NIST asked to NTRU Prime submitters to provide a security level 5 parameter set.

5 Performance Analysis

In this section, the theoretical and practical performance results will be reported. For every scheme, different security levels and optimized versions are provided. To make the comparison as fair as possible, in this section we analyze the NIST level 5 non-optimized version for each scheme.

5.1 Bandwidth and Memory Comparison

Table 1 shows the claimed sizes (in bytes) of the public key (pk), private key (sk) and ciphertext (ct) for the analyzed schemes.

scheme	pk	sk	ct
NewHope	1824	3680	2208
FrodoKem	21 520	43 088	21 632
Kyber	1568	3168	1568
Three Bears	1584	40 (seed)	1697
NTRU	1230	1592	1230
LAC	1056	2080	1424
Saber	1312	1664 (384)	1472
Streamlined NTRU Prime	1184	1462	1312
NTRU LPrime	1322	1999	1184
R5ND_5CCA _5d	978	-	1285
R5N1_5CCA _0d	14 636	-	14 708

Table 1: Claimed bandwidth and memory performance (bytes)

5.2 Run-time Comparison

Table 2 shows the claimed run-time efficiency measured in CPU cycles. The machines the schemes were tested on are also reported.

Table 3 shows the experimental run-time data obtained by running the schemes on a Intel Core i7-8750H 2.2 GHz with hyper threading and turbo boost on. It can be noticed that the schemes generally achieved a better runtime performance on this machine then on the test-benches used by the submitters. A possible explanation could be that hyper-threading and turbo-boost were enabled on this machine while they were disabled on the other test-benches. The reason why the schemes were tested using these configurations is that eventually, when one or more schemes will be standardized, they will run on 'ordinary' machines with this features enabled.

Moreover, it can be also noticed that the relative differences in performance do not change; RLWE and MLWE achieve high runtime performance, LWE scheme is the slowest one, and NTRU features competitive performance for encoding and decoding but significantly slower key generation than structured LWE schemes.

It is worth noticing that data for Three Bears and Streamlined NTRU Prime is missing. This is due to an error during compilation that prevented them from running. Since this is a theoretical research project, and gathering experimental data is an addition, it was decided not to spend more time trying to fix these errors, but to focus on more in depth theoretical analysis on the security of the schemes. Finally, runtime results for NTRU LPrime are exaggeratedly big. This suggests a possible mistake made when executing the experiment. However, due to time constraints, this situation was not further investigated.

5.3 Failure Rate

Table 4 shows the claimed decryption failure rates of the analyzed schemes.

From this table it can be noticed that NTRU based schemes are the only correct KEMs; however, the failure rate of the other schemes is so low that they can be assumed to be correct.

scheme	machine	key gen	enc	dec
NewHope	Intel core i7-4770k 3.5 GHz	244 944	377 092	437 056
FrodoKem	Intel core i7-6700 3.4 GHz	30 301 000	32 611 000	32 387 000
Kyber	Intel core i7-4770k 3.5 GHz	331 418	396 928	451 096
Three Bears	Intel core i3-6100U 2.3GHz	118 000	145 000	211 000
NTRU	Intel core i7-4770k 3.5 GHz	31 835 958	1 856 936	4 920 436
LAC	Intel core-i7-4770S 3.1GHz	377 123	643 024	916 835
Saber	Intel Core I5-7200U 2.5 GHz	131 000	159 000	165 000
Stremlined NTRU Prime	Intel Xeon E3-1275 v3 3.5 GHz	940 852	44 788	93 676
NTRU LPRIME	Intel Xeon E3-1275 v3 3.5 GHz	44 948	81 144	113 708
R5ND_5 CCA_5d	Intel Core i7 2.6GHz	101 000	152 000	207 000
R5N1_5 CCA_0d	Intel Core i7 2.6GHz	29 048 000	26 589 000	26 844 000

Table 2: Claimed runtime performance (cpu cycles)

scheme	key gen	enc	dec
NewHope	138 420	195 328	227 560
FrodoKem	2 813 367	3 587 118	3 414 095
Kyber	225 804	256 810	279 999
NTRU	3 431 973	341 056	168 663
LAC	101 422	171 105	286 049
Saber	150 279	174 258	189 999
NTRU LPrime	6 373 771	12 708 470	19 060 046

Table 3: Experimental runtime performance (cpu cycles)

6 Security Analysis

To analyze the security level of the schemes, we only consider attacks on unstructured lattices since no attacks that can exploit the additional structure of the ideal or NTRU lattices have been found so far. Many strategies to assess the security of the encryption schemes are known. However, we can rule out BKW (Blum-Kalai-Wasserman) and linearization attacks due to the limited number of provided samples by the schemes, leaving out with two BKZ (block Korkine-Zolotarev) attacks: primal and dual attack. Primal attack is a widely used strategy to assess security for the search-LWE problem because its estimation is pretty conservative and it only requires polynomial LWE samples [**primal˙attack**].

6.1 Primal Attack

Informally, one can say that the primal attack consists of constructing a unique-SVP instance from the LWE problem and solving it using BKZ. The BKZ algorithm reduces the lattice basis by using an oracle in a smaller dimension b. The two widely used techniques are (1) enumeration, that runs in super exponential time but requires limited memory and therefore, is efficient in low dimensions; (2) sieving, which is exponential both in time and in space and is more efficient than enumeration in higher dimensions. Although it is known that the number of calls to the oracle is polynomial, it is really hard to estimate it precisely. Therefore, the primal attack aims to evaluate the core-SVP hardness, which consists of just one call to the oracle in the smaller dimension b, clearly underestimating the actual security of the scheme [26]. A more formal definition of the primal attack is given below.

Primal attack **:** Given a concrete LWE instance (A, b), construct the lattice as $\Lambda = \{x \in \mathbb{Z}^{m+n+1} : (A|I_m| - b)x = 0 \mod q\}$ of dimension $d = m + n + 1$, volume q^m, and with a unique-SVP solution v = (s | e | 1) of norm $\lambda \approx \sigma\sqrt{n+m}$. Then solve using a lattice reduction algorithm, such as BKZ, with appropriate block size b.

In Table 5 are reported the estimated classical and quantum costs, given in $\log_2$ of CPU operations, for primal attacks on the level 5 security version of the schemes.

6.2 Dual Attack

Dual Attack is another strategy to assess the security of a scheme; however, it works only on LWE based schemes and not on NTRU based ones.

Dual Attack **:** The Dual Attack aims to solve the Decisional-LWE by reducing it to an instance of the Short Integer Solution Problem (SIS). The SIS problem consists of finding short vectors in the lattice $L = \{x \in Z_q^m | x^t * A \equiv 0 \mod q\}$, where the rows of $\boldsymbol{A}$ are the LWE samples.

[5]Values computed respectively in the non-local and local model.

[6]Quantum values for the primal attack are not reported in NTRU specification paper.

scheme	failure rate
NewHope	2^{-213}
FrodoKem	2^{-252}
Kyber	2^{-228}
Three Bears	2^{-256}
NTRU	correct
LAC	2^{-138}
Saber	2^{-165}
NTRU Prime	correct
R5ND_5CCA_5d	2^{-239}
R5N1_5CCA_0d	2^{-151}

Table 4: Claimed failure decryption rate

In Table 6 are reported the dual attack security estimates, given in $\log_2$ of CPU operations, for the level 5 schemes for which a dual attack analysis can be applied.

7 Responsible Research

In this section, we motivate the ethical needs of this project and we propose an ethical discussion about the reproducibility of the experiments conducted within this research.

7.1 Ethical Discussion about the Project

Cryptography is a branch of mathematics and computer science that aims to protect people's privacy by encrypting data and guaranteeing confidentiality (only who has access to some data can decipher it) and integrity (only who is allowed to change data can change it). It consists of encryption and digital signature schemes. These schemes are used every day in applications such as online banking, messaging systems, medical systems...

However as explained in the introduction, quantum computers will threaten some of the most widely used public key encryption and digital signature systems, placing at risk people's privacy.

The goal of this project is to widen the study on new quantum-resistant cryptosystems that will allow people to protect their privacy.

7.2 Reproducibility of Experiments

This research project is mainly theoretical. This means that most of the information and results reported in this paper have been gathered by studying the existing literature.

The research was conducted in the most possible unbiased way, trying not to be influenced by the opinions that already exist on the analyzed schemes. The data is reported in the most transparent way possible.

scheme	classical	quantum
NewHope	259	235
FrodoKem	281.6	256.3
Kyber	256	232
Three Bears	354	321
NTRU	179/253[5]	- [6]
LAC	323	293
Saber	283	257
Streamlined NTRU Prime	153-368	139-208
NTRU LPrime	140-210	153-364
5ND_5CCA_5d	256	233
R5N1_5CCA_0d	257	234

Table 5: Claimed primal attack cost

scheme	classical	quantum
NewHope	257	233
FrodoKem	279.8	254.7
Kyber	256	232
LAC	320	290
Saber	338	308
5ND_5CCA_5d	259	235
R5N1_5CCA_0d	257	234

Table 6: Claimed dual attack cost

As for the practical experiments, an ordinary Intel x64 laptop was decided to use as a common unbiased test-bench to asses the practical runtime performance of the schemes. To reproduce this experiment, one has to clone the source codes that can be found here [3] and run the test scripts already provided. As the schemes were developed on different operating systems, using different libraries and language versions, the most difficult and time consuming part of the experiment will be to make the source codes compile and run without mistakes.

8 Discussion

In this section, we discuss the results reported in the sections 5 and 6 and present a general overview of the lattice-based cryptography up to date with NIST round 2.

8.1 Bandwidth and Memory Performance

As can be seen in table 1, the data supports the theoretical claims presented in the background section. LWE based systems suffer from high bandwidth and memory usage, as can be noticed by looking at FrodoKEM's data. Generally, LWR schemes require less bandwidth than LWE ones because of the compactness of the keys and ciphertexts achieved by rounding to a smaller modulo. In fact, Saber requires less bandwidth and memory than Kyber. It can also be noticed that schemes based on structured lattices feature high and comparable performance.

Private keys are not a deciding factor in bandwidth requirements because they are never shared, but they account for memory usage. ThreeBears and Saber give up a little bit of performance to generate private keys from a small seed, which makes them more suitable for systems characterized by limited memory, such as embedded systems.

8.2 Runtime Performance

The data gathered in Table 2 and in Table 3 supports the runtime claims given in the background. It is clear that schemes based on structured lattices have better runtime performance than those based on unstructured lattices. Schemes based on MLWE/R and RLWE/R achieve comparable results, while NTRU has similar performance for encapsulation and decapsulation but has a very slow key generation due to polynomial division required in the process.

However, the analyzed schemes have different features and can be optimized for different platforms and applications. As an example, NTRU seems to feature high performance on platforms which support AVX2 operations. As a consequence, there is no best scheme overall.

8.3 Security

As mentioned before, schemes based on variants of Module or Ring LWE/R have theoretical lower security than ones based on plain LWE because of their extra structure that attackers could exploit. Therefore, FrodoKEM, which is the only scheme based on LWE assumption, could be used in sensitive applications that need high security and do not have strict performance constraints.

Schemes based on MLWE/R and RLWE/R witness more structure. Even though their performance is similar, the cryptographic community developed a preference towards MLWE/R schemes because of their less structure and therefore higher theoretical security.

ThreeBears is the only submitted scheme based on I-MLWE. The goal of this scheme is to explore new possibilities for lattice-based cryptography. However, even though the security seems similar to the other schemes, NIST decided not to consider it for standardization because it believes the scheme has not being studied enough.

NTRU seems to feature similar security with respect to LWE/R schemes. A significant disadvantage is that it lacks a formal average-case to worst-case reduction. However, since it has been subjected to cryptoanalysis for more than two decades, and some versions have been standardized by other organizations, it might be the most conservative choice.

8.4 Current status of round 3

The submissions of round 3 are slightly different from the ones of round 2. The creators tried to improve them taking into consideration the feedback received from the cryptographic community and NIST itself. As a result, the finalist schemes might have sightly different parameter sets which increase security or performance, more thorough theoretical proofs and more cryptoanalitic study as a support. However, overall the schemes did no go through significant changes.

Since round 3 is happening at the time of writing, and will still go on for almost a year, not a lot of comments and reviews about the schemes have been published yet.

9 Conclusions and Future Work

In this report we analyzed the most promising lattice-based encryption schemes submitted to the NIST post-quantum standardization competition. The lattice-based encryption schemes that were admitted to the final round are Saber [18], Kyber [17] and NTRU [24]. NIST declared that by the end of the final round (early 2022), at most only one of these three schemes will be standardized. Moreover, FrodoKEM [14] and NTRU Prime [25] are taken into consideration as an alternate finalist together with other three encryption/KEM algorithms. These algorithms will need to be studied further during a fourth round before they can be taken into consideration for standardization.

As guidelines for the future, in addition to more cryptanalytical study, NIST asked the cryptographic community to investigate more on side channel resistant implementations, performance data in internet protocols, and performance data for hardware implementations.

Even though a big enough quantum computer is not built yet, scientists believe that by 2035, they will build one that can break classical public-key cryptographic schemes. This means that an attacker could steal encrypted data today and decrypt it in the next twenty years. Therefore, it is important to deploy post-quantum encryption algorithms as soon as possible. However, they cannot be fully trusted to be resistant against both classical and quantum attacks. The best solution to this problem is to have a transition phase in which people will use hybrid encryption schemes composed by both classical and post-quantum encryption algorithms. Some experiments have already been carried out in the past. For example Google used NewHopeUsenix[7] together with ephemeral elliptic curve Diffie–Hellman (ECDH) key exchange in a hybrid TLS 1.2 ciphersuite in an experimental version of the Chrome browser. Google has then reported

[7] An old version of NewHope-CPA.

than they did not encounter any unexpected impediment and that the performance was affected by just a small margin.

References

[1] P.W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: (1995).

[2] NIST. "Post Quantum Cryptography Round 1 submissions." In: (). [Online; accessed 19-April-2021].

[3] NIST. "Post Quantum Cryptography Round 2 submissions." In: (). [Online; accessed 19-April-2021].

[4] NIST. "Post Quantum Cryptography Round 3 submissions." In: (). [Online; accessed 19-April-2021].

[5] T. Fernandez-Carames and P. Fraga-Lamas. "Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks". In: (Apr. 2020).

[6] G. Zhang and J. Qin. "Lattice-based threshold cryptography and its applications in distributed cloud computing". In: (June 2015).

[7] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: (2005).

[8] B. Buchanan. "Learning With Errors and Ring Learning With Errors". In: (July 2018). [Online; accessed 22-June-2021].

[9] M. R. Albrecht and A. Deo. "Large Modulus Ring-LWE Module-LWE". In: (Jan. 2020).

[10] J. Hoffstein, J. Pipher, and J.H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: (1998).

[11] Chris Peikert. "QCrypt 2016". In: http://2016.qcrypt.net/. Joint Center for Quantum Information and Computer Science (QuICS) at the University of Maryland. Sept. 2016.

[12] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure Integration of Asymmetric and Symmetric Encryption Schemes". In: *Advances in Cryptology — CRYPTO' 99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 537–554. ISBN: 978-3-540-48405-9.

[13] Prasanna Ravi et al. "Lattice-Based Key-Sharing Schemes: A Survey". In: *ACM Comput. Surv.* 54.1 (Jan. 2021). ISSN: 0360-0300. DOI: 10.1145/3422178. URL: https://doi.org/10.1145/3422178.

[14] E. Alkim et al. "FrodoKEM, Learning With Errors Key Encapsulation". In: (Sept. 2020). https://frodokem.org/.

[15] T. Pöppelmann et al. "NewHope, Algorithm Specifcations and Supporting Documentation". In: (Nov. 2017). https://newhopecrypto.org/.

[16] V. Lyubashevsky, C. Peikert, and O. Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Springer Berlin Heidelberg, 2010, pp. 1–23.

[17] R. Avanzi et al. "CRYSTALS-Kyber, Algorithm Specifications And Supporting Documentation". In: (Jan. 2021). https://pq-crystals.org/.

[18] J.P. D'Anvers et al. "SABER: Mod-LWR based KEM (Round 2 Submission)". In: (2018). https://www.esat.kuleuven.be/cosic/pqcrypto/saber/.

[19] X. Lu et al. "LAC, Lattice-based Cryptosystems". In: ().

[20] M. Hamburg. "Post-quantum cryptography proposal: Three Bears". In: (Mar. 2019). https://sourceforge.net/projects/threebears/.

[21] O. Garcia-Morchon et al. "Round5: KEM and PKE based on (Ring) Learning with Rounding". In: (Apr. 2020).

[22] O. Garcia-Morchon et al. "Round2: KEM and PKE based on GLWR". In: (2019).

[23] M.J. O. Saarinen. "HILA5: Key Encapsulation Mechanism (KEM) and Public Key Encryption Algorithm". In: (Nov. 2017).

[24] C. Chen et al. "NTRU, Algorithm and Supporting Documentation". In: (Mar. 2019). https://ntru.org/.

[25] D.J. Bernstein et al. "NTRU Prime: Round 2". In: (2019).

[26] E. Alkim et al. "Post-quantum key exchange - a new hope". In: (2015). https://eprint.iacr.org/2015/1092.

[27] H. AlMaget and A. AlMogren. "A Secure and Efficient ECC-Based Scheme for Edge Computing and Internet of Things". In: (2020).

A NTRU original scheme

Parameters **:**

(N, p, q): three integer parameters, $q >> p$.
L_f, L_g, L_ϕ, L_m: sets of polynomials of degree $N-1$ and integer coefficients.

Notation **:**

$$f \circledast g = h \text{ with } h_k = \sum_{i=0}^{k} f_i \cdot g_{k-i} + \sum_{i=k+1}^{N-1} f_i \cdot h_{N+k-i}$$

KeyGeneration **:**

choose polynomials $f, g \in L_g$ so that f has inverses F_q and F_p such that:
$F_q \circledast f \equiv 1 \mod q$ and $F_p \circledast f \equiv 1 \mod p$

compute public key $h = F_q \circledast g \mod q$.

Encryption **:**

select message $m \in L_m$ and random polynomial $\phi in L_\phi$.

compute ciphertext $e \equiv p \cdot \phi \circledast h + m \mod q$.

Decryption **:**

first compute $a \equiv f \circledast e \mod q$.

recover the message by computing $m = F_p \circledast a \mod p$.

B IND-CPA and IND-CCA

Ciphertext indistinguishability is the ability of a pair of ciphertexts to be indistinguishable to an attacker based on the messages they encrypt.

$\boldsymbol{IND-CPA}$ **:** an attacker feeds an encryption oracle with messages m_1 and m_2 and gets back the ciphertexts c_1 and c_2. The scheme is IND-CPA secure if the attacker has only a negligible advantage over random guessing to recognise which ciphertext represent which plaintext. The attacker is said to have a negligible advantage if they win the challenge with probability $1/2 + \epsilon$, where ϵ is a small advantage.

$\boldsymbol{IND-CCA}$ **:** in addition to the abilities owned by the attacker in the IND-CCA challenge, they now have access to the decryption oracle that they can feed with every ciphertext except with the ones the have to decrypt. The scheme is IND-CCA secure if the attacker has only negligible advantage of winning the challenge.

Image 3 shows the queries available to an attacker in IND-CPA and IND-CCA models.

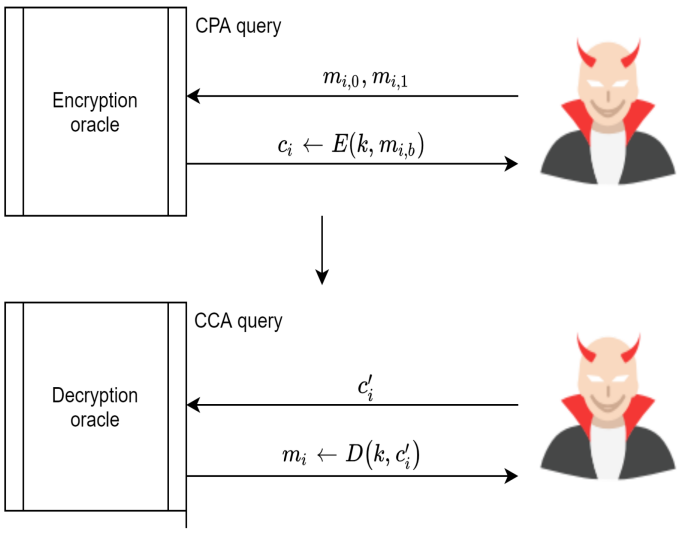

Figure 3: Image taken from [27]

C NTRU Genealogical Tree

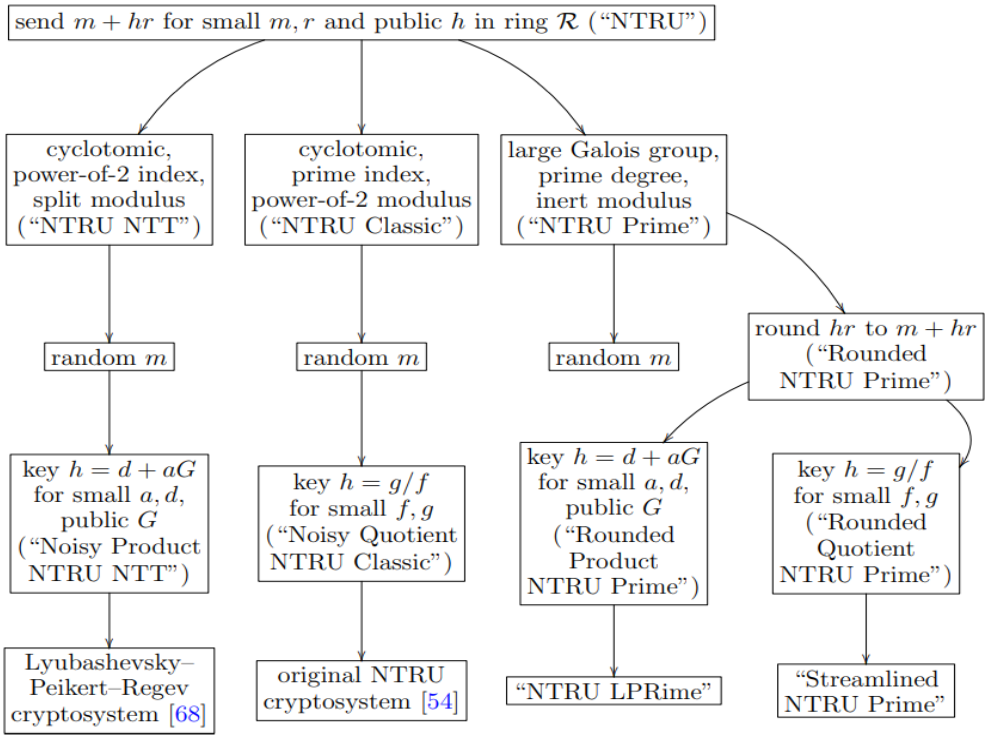

Figure 4: Image taken from [25]