An Analysis of the Performance and Randomness of Lattice-based Public Key Cryptography Algorithms

Rafael da S. Oliveira
Military Institute of Engineering Rio de Janeiro, RJ, Brazil

José A. M. Xexéo
Military Institute of Engineering Rio de Janeiro, RJ, Brazil

Renato H. Torres
Federal University of Para´ Belém, PA, Brazil

ABSTRACT
The security of public key cryptography currently used has becomea growing concern because of the advancement in the development of quantum computers. Therefore, the study of post-quantum cryp-tography becomes very relevant. Because of this scenario, the Na- tional Institute of Standards and Technology (NIST) is holding a contest to evaluate proposals for post-quantum cryptography algo- rithms for future standardization. This work performs performanceevaluations of lattice-based public key cryptography schemes thatparticipated in the third phase of the contest. Additionally, an anal-ysis of the randomness of the cryptograms generated by these al-gorithms is carried out. Based on the results found, it was possibleto view the performance of these encryption schemes and comparesome of their characteristics regarding the level of randomness.

Keywords
Post-quantum Cryptography, Lattices, Public Key Cryptography,Randomness test

1. INTRODUCTION
Most public key cryptographic systems currently used have their security guaranteed through the difficulty in solving mathemati- cal problems such as factoring integers [1] and calculating dis- crete logarithms [2] that demand exponential time for their solu- tion. This scenario may change with the advent of the quantum computer, which employs algorithms such as Shor [3], which can solve these problems in polynomial time under a quantum compu- tational model.

Post-quantum cryptographic algorithms are secure on classical computers and resistant to quantum attacks. Since 2016, the contest organized by NIST Post-Quantum Cryptography Standardization (PQCS) has been underway, which aims to evaluate and standard-ize one or more post-quantum public-key cryptography algorithms. Lattice-based cryptography is a promising post-quantum cryptog-raphy class, given its flexibility, performance, and the possibility of being used in various security problems such as public key cryptog-raphy, digital signature, and homomorphic cryptography [4]. In ad-dition, we can highlight that of the four finalist proposals in NIST for a standard in public key cryptography, three are based on the lattice.

Some works evaluate the performance of post-quantum crypto-graphic algorithms in specific hardware configurations, such as [5], which evaluates the performance of finalist schemes in the second phase of the NIST contest based on lattice for public-key cryptog-raphy and digital signature. Evaluations were performed using the ARM Cortex-M4 family of processors. In [6], an evaluation is car-ried out on public key cryptography algorithms based on the lattice, code theory, and elliptic curves on an Intel(R) CPU i7-5500 plat-form. In [7], an evaluation of the public key and digital signature algorithms finalists of the third phase of the NIST contest is carried out on a Raspberry Pi 3B+ platform.

Unlike the previously mentioned works that employ cryptographic schemes in different versions and on specific hardware, this work aims to evaluate the performance of reference versions of lattice-based public key cryptography schemes in a Linux environment from a virtual machine. The main objective of these experiments is to verify the performance of these algorithms in an implementa- tion via software simulation, considering the parameter's key sizes, CPU cycles, and execution time. Furthermore, the randomness of the cryptograms generated by these algorithms is evaluated to ver- ify the possibility of using attacks that use the cryptogram's ran-domness measure, such as distinction attacks [8]. Thus, the main contributions of this work are to evaluate candidates for the NIST standard regarding their performance in software-based applica-tions and their possible deficiencies regarding the level of random-ness of the algorithm.

This work is organized as follows: first, the basic definitions of lat-tices are presented, the main problems used in lattice-based cryp-tography, and an introduction to random number generators. Next, a summary of the lattice-based encryption schemes to be analyzed CRYSTALS-KYBER, NTRU, and SABER is made, after which a simulation scenario is defined where the algorithms are evaluated about the parameters mentioned above, and an evaluation of the observed results in addition to a comparison with the related work mentioned above. Then, the randomness of the cryptograms gener-ated by these algorithms is assessed through two random number generator tests. Finally, a conclusion on the results obtained and a suggestion for future work.

2. DEFINITIONS
2.1 Lattices
A lattice is a set of points in n-dimensional space with a periodic structure [9]. More formally, given n linearly independent vectors,we have that B is the basis of the lattice being defined as:

$B = \{b_1, b_2, b_3, ..., b_n\}, b_k \in \mathrm{R}^n$

The lattice generated by them is the set of vectors given by:

$$\mathcal{L}(B) = \mathcal{L}(b_1, b_2, b_3, ..., b_n) = \left\{ \sum_{i=1}^{n} a_i b_i \mid a_i \in \mathbb{Z}^n \right\}$$

A lattice can be generated from different bases. These bases can be considered good or bad, depending on the size and orthogonality ofthe vectors composing them. A set of small and relatively orthogonal vectors constitute a basis considered good. In contrast, large non-orthogonal vectors are usually composed of bad bases, considering they need complex combinations to produce the same lattice. Lattice-based public key systems use good bases to generate publickeys and bad bases for private keys.

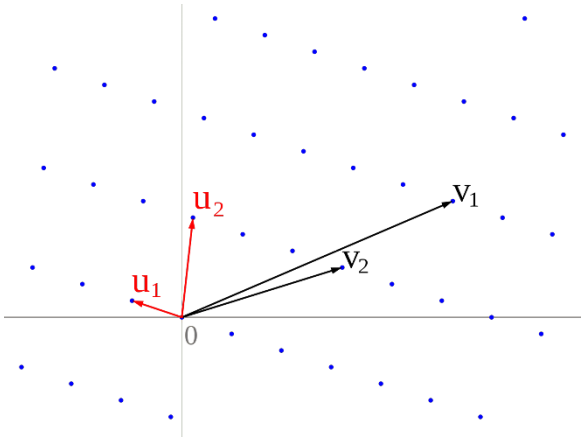

Fig. 1: A lattice in R^2 and two of its bases

The development of lattice studies in the context of cryptography gained relevance from the results obtained by Ajtai [10], who pro-posed that lattices could be used not only as a tool for cryptanalysis but also to build cryptographic primitives. The security of lattice- based algorithms is associated with difficult-to-solve computational problems, such as the Shortest Vector Problem (SVP), which in- volves finding the smallest non-zero vector in a lattice, and the Closest vector problem (CVP), which aims to find the lattice point that is closest to a given target point (which may not be a lattice point). [11].

2.2 Learning With Errors (LWE)

The LWE is a fundamental problem in lattice-based cryptography which consists of finding a secret $s \in Z_q^n$ of a sequence of approximate random linear equations in s [12]. In general, the problem can be defined given: a is a polynomial with uniformly random sampled coefficients in Z_q^n where n and q are degrees and modulusof the lattice respectively, and e a vector of small errors also ran- dom. Given the equation below, we have the pair (a, b) the public key, and s is the private key.

$$(s, a_i) + e_i = b_i \mid 1 \leq i \leq n$$

The security of the system is due to the difficulty of finding s giventhe insertion of small errors in e There are variants of LWE that can be created when the lattice and the problem of interest are definedover a ring (RLWE) or a modular group (MLWE).

2.3 Learning With Rounding(LWR)

Another fundamental problem in lattice-based cryptography is the LWR, which has a similar structure to LWE however instead ofadding a small random error to several samples $(s, a) \in Z_q^n$ to hidethe exact value of s a deterministically rounded version of (s, a), given a value p where $p < q$, we divide the elements of Z_q^n into $p$contiguous intervals of approximately q/p each element is defined by the given rounding function $\lfloor \cdot \rfloor_p : Z_q^n \rightarrow Z_p^n$.

Thus we can define the secret key as the vector s and the key publicas $(A, \lfloor A \cdot s \rfloor_p)$ [13].

The (RLWR) and (MLWR) are the ring-defined and modular vari-ants of the LWR, respectively.

2.4 Random number generators

The use of random and pseudo-random numbers appears in severalcryptographic applications. A random bit string can be interpretedas the result of flipping a “fair” unbiased coin with sides that are labeled “0” and “1”, with each flip having a probability of exactly(50%) of producing one of the two results. Also, the postings are independent of each other. The result of any previous coin postingsdoes not affect future coin postings. The “fair” unbiased currency is, therefore, the perfect random bitstream generator since the “0” and “1” values will be randomly and uniformly distributed [14].

3. POST-QUANTUM ALGORITHMS EVALUATED

3.1 NTRU

N-th degree Truncated polynomial Ring Units (NTRU) [15] is an encryption scheme that is over 20 years old, being implemented in post-quantum cryptography to achieve the security of attack in-distinguishability under adaptive chosen ciphertext (IND- CCA2).The NTRU algorithm originally used polynomial algebra, a reduc-tion module for encryption, and elementary probability theory for decryption. With the original design, it is a partially correct proba- bilistic public key encryption scheme (PPKE partially correct) but can be transformed to a deterministic correct public key encryption scheme (DPKE) using transforms since it is based on lattice [16].

3.2 CRISTALS-KYBER

Cryptographic Suite for Algebraic Lattices (CRYSTALS) encom-passes two cryptographic primitives: KYBER (public-key cryp- tography) and Dilithium (digital signatures). The KYBER algo- rithm consists of two parts: an IND-CPA secure public-key en- cryption scheme that encrypts 32-byte fixed-length messages anda Fujisaki–Okamoto (FO) transformation to ensure the IND-CCA2 [17]. CRYSTALS-KYBER use a variation of the LWE lattice prob- lem, where vectors of n dimensions are replaced by polynomials

of degree less than n composed of integer elements and modular operations. This construction is called MLWE.

3.3 SABER

SABER is a post-quantum schema whose security depends on the difficulty of solving the Learning with Rounding (LWR) problem,which is a variation of LWE. The LWR problem works differentlythan LWE, where the algorithm uses rounding the samples to cre- ate “noise” rather than adding error. SABER applies the modu-lar version of the problem (MLWR) [18]. The algorithm consists of SABER.PKE and SABER.KEM, with SABER.PKE generates the public key of the encryption algorithm. According to [19], Saber.PKE alone cannot combat chosen ciphertext (IND-CCA2) attacks, so Saber.PKE is compiled into Saber.KEM uses a post- quantum variant of the Fujisaki-Okamoto (FO) transformation) toachieve this level of security.

4. SIMULATION AND SYSTEM SETUP

The simulations were performed using a computer with the follow- ing configuration: AMD Ryzen 3 PRO 4350G with 8GB of RAM,with Virtual Box software version 6.1.34 and Kali Linux 2022.2 64bit Operating System with 2GB virtualized RAM.

To evaluate the performance of the algorithms, it used liboqs [20], an open-source C library, to analyze post-quantum cryptographic algorithms. The library implements the official version of the al- gorithms available at NIST, and this library is also used in the re- lated works [6], and [7]. In the experiments, the liboqs library fileswere compiled using the GCC 11.3.0 compiler, observing the pa- rameters: key size, number of CPU cycles, and time to perform theoperations.

The random number generator test batteries developed by NIST version 2.1-2 [16] and Dieharder version 3.31.1 [21] were used in the randomness tests.

The experiments were performed considering three security levelsspecified by NIST (I, III, and V), according to Table 1. It should be noted that the evaluated schemes have different versions with other security levels. In order to assess the algorithms more closely, the reference versions of the algorithms were used at the same securitylevels, according to Table 2.

Table 1: Security levels for evaluating candidate NIST

Level	Security Description
I	As hard to break as the AES128 (exhaustive search)
II	As hard to break as the SHA256 (collision search)
III	As hard to break as the AES192 (exhaustive search)
IV	As hard to break as the SHA384 (collision search)
V	As hard to break as the AES256 (exhaustive search)

Table 2: Versions of the algorithms considered

Level	NTRU	KYBER	SABER
I	ntruhps2048509	Kyber512	LightSaber
III	ntruhps2048677	Kyber768	Saber
V	ntruhps4096821	Kyber1024	FireSaber

5. RESULTS AND DISCUSSION

5.1 Size of keys

Initially, the key size parameter was evaluated according to Figure

2. In this regard, the NTRU encryption family obtained a slight ad-vantage over the KYBER and SABER algorithms since their cryp-tograms have smaller sizes with the same level of security com- pared to your peers. In related works employing hardware, the re- sults were like those observed in these analyses, so we can concludethat the platform used does not influence this parameter. Therefore, the size of the keys is directly related to the different versions and security levels used by the algorithm.

It should be noted that in environments with limited memory capac- ity or data transmission, this parameter is quite relevant. In addition, some systems need to store large volumes of encrypted data in spe-cific applications. Thus, the analysis of this parameter is essential in the algorithm's decision to be used.

5.2 Processing time

In this evaluation, as shown in Figure 3, it was possible to observe that algorithms from the SABER family had better performance, with the time of operations in the order of 10 milliseconds, while the algorithms of the KYBER and NTRU class performed most of the operations in the order of 50 milliseconds. It should be noted that in the key pair generation operations, the NTRU scheme pre- sented a high time compared to the other processes.

Other works carried out the evaluation of these algorithms on a hardware platform. In [5], tests are performed using the KYBER and SABER algorithms, not evaluating the NTRU. Regarding the parameter evaluated, the KYBER scheme presents a performance similar to SABER. In related work [6], the KYBER and NTRU en- cryption schemes are evaluated, SABER algorithm is not evaluatedin this work. Regarding the execution time, the KYBER scheme performs considerably superior to NTRU. In [7], tests are per- formed with the three algorithms. In this work, the algorithms of the SABER obtained results slightly superior to KYBER and con-siderably superior to NTRU. The evaluation of this parameter is essential for systems with time as a critical factor for their applica-tions.

5.3 Clock cycles

Finally, the clock cycle parameter was evaluated, as shown in Fig-ure 4. The performance of the algorithms was similar to those ob- served in the previous experiment, with the SABER algorithms ob-taining better results, followed by KYBER and NTRU. Again, thehigh number of cycles required in the key pair generation opera- tions of the NTRU scheme is highlighted. In [5], the results ob- tained using this parameter demonstrate a very close performance, with KYBER being better than SABER. The clock cycle parameter was not evaluated in the works [6] and [7].

Based on these observed results, it was possible to observe a rela- tionship between the number of clock cycles and the algorithm's execution time for these operations. These factors are of great rele-vance for the choice of the encryption scheme.

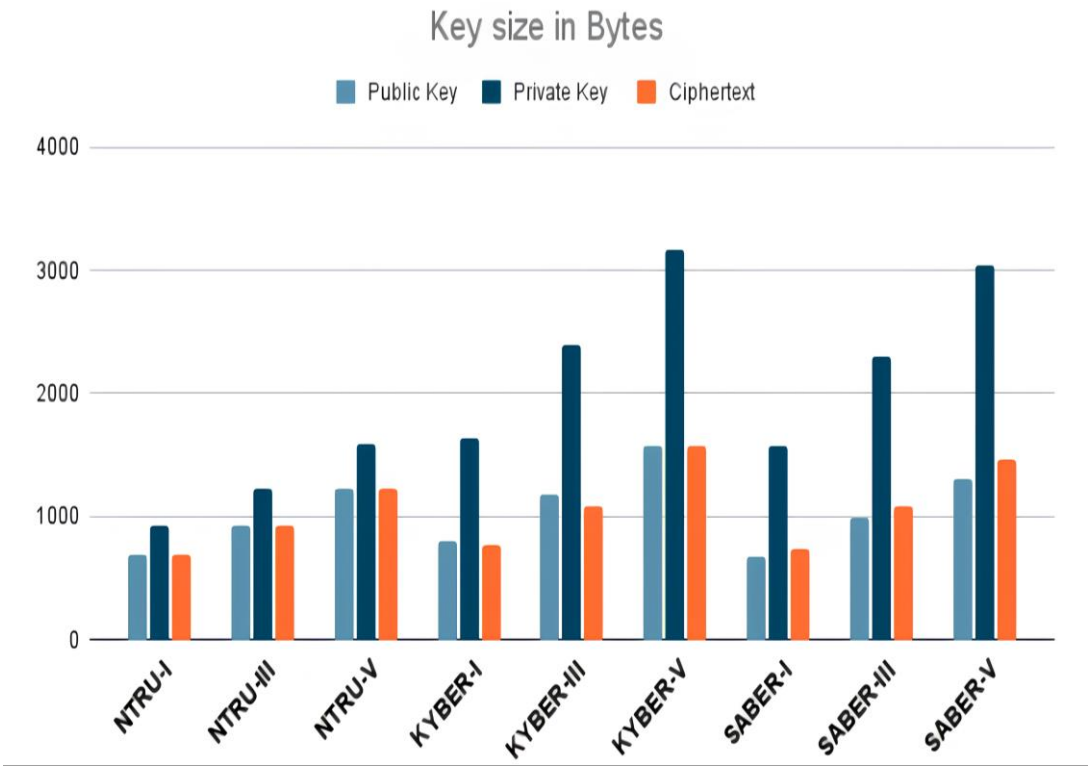

Fig. 2: Key size in bytes

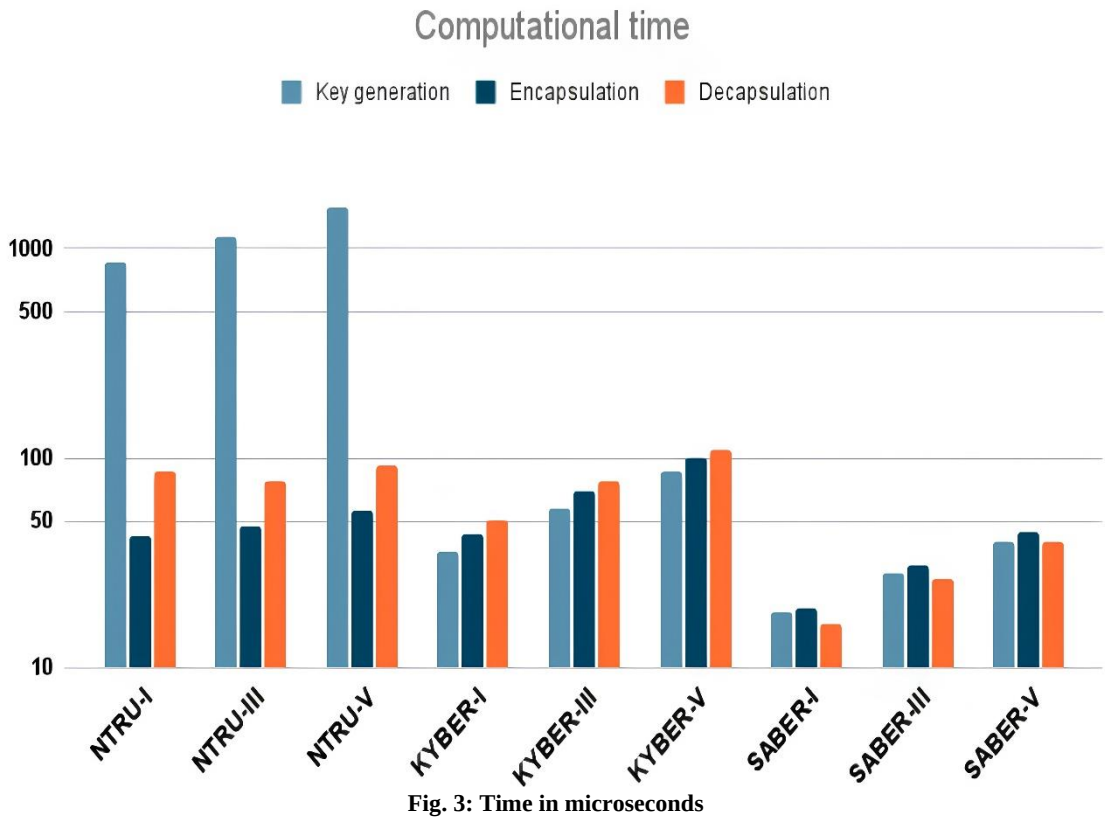

Fig. 3: Time in microseconds

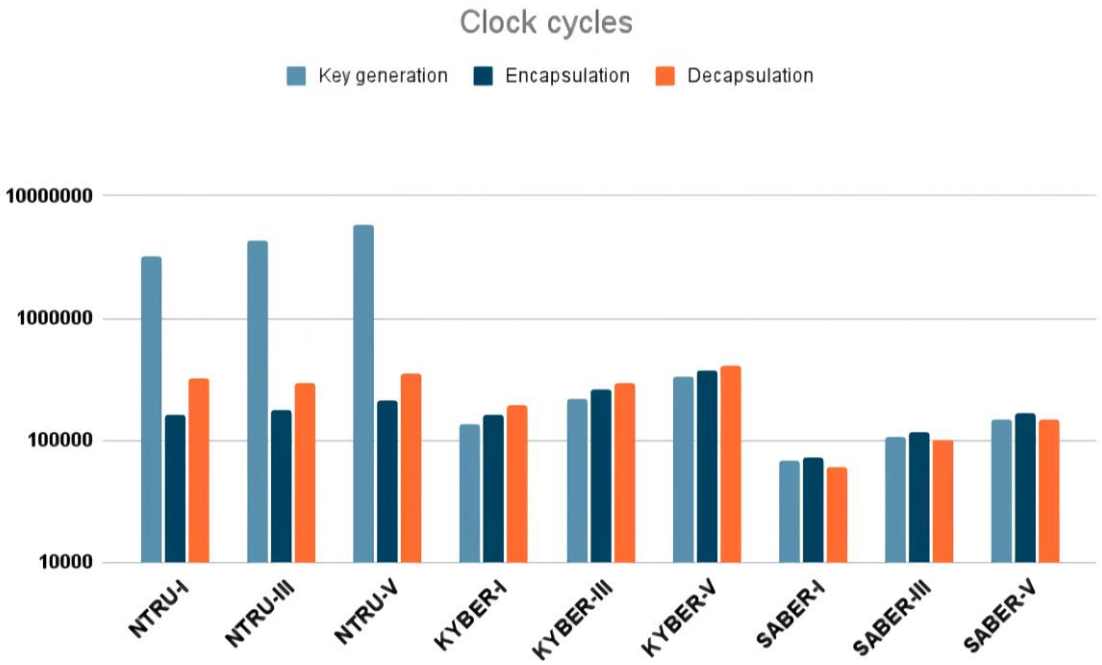

Fig. 4: Number of clock cycles for the operations

5.4 Randomness tests

To carry out the tests, generating the cryptograms was the samein the three algorithms. Binary files were generated from 20,000 samples of ciphertexts using the encapsulation of random keys. Inthe test batteries, the default settings were used, and the *p*-value ofeach bit sequence was evaluated in the tests, so the *p*-value<0.01 indicates that the sequence was not random, with a confidence levelof 99%.

5.5 NIST test battery

The battery is composed of 15 tests that evaluate different patternsof randomness, 10 bitstreams of 2,000,000 bits were used in the tests for each version of the cryptographic algorithm evaluated. Asspecified in the software documentation, if more than 2 bitstreamsfail a test, the random number generator will be considered disap- proved.

As described in Table 3, the KYBER cryptograms were deprecatedin test 11 (Approximate Entropy), which is based on the probabilitythat models in the sequence that are similar will remain similar in the subsequent incremental comparisons [16]. The other two algo-rithms tested passed all tests.

5.6 Dieharder test battery

Dieaharder is a test battery developed by Robert G. Brown from Diehard to facilitate testing (pseudo) random number generators, both software and hardware, for various purposes in research and cryptography. It is composed of 29 tests subdivided into other sub-tests, and accordingly [21], it has a higher level of demand in testsfor generating random numbers.

Table 3. : KYBER faults in NIST test battery

Statistical test	KYBER-I	KYBER-III	KYBER-V
Frequency	0	1	0
Block Frequency	0	0	0
Cumulative Sums	0	1	0
Runs	0	0	0
Longest Run	0	0	1
Rank	0	0	0
FFT	0	0	0
Non-Overlapping Template	1	1	1
Overlapping Template	2	0	0
Universal	0	0	0
Approximate Entropy	5	9	10
Random Excursions	0	0	0
Random Excursions Variant	1	1	1
Serial	0	0	0
Linear Complexity	0	0	0

The algorithms were submitted to 114 tests, being evaluated ac- cording to p-value in the scales of failed (*p*-value = 0), weak (0 <*p*-value<0.01) and approved (*p*-value>0.01), the number of tests the algorithms being observed in Figure 5.

In the battery of Dieharder tests, the cryptograms generated by the SABER algorithm obtained the best results, being approved in about 63% of the tests. On the other hand, the KYBER obtained theworst results, being failed 40% and weak in 15% of the evaluations.

Based on the tests carried out, we can verify that the cryptograms generated by KYBER have some flaws with randomness. Althoughthese tests do not demonstrate that the algorithm is vulnerable, the results obtained can be used in cryptanalysis models that use as parameters to evaluate the level of randomness to check for possible flaws in the algorithm.

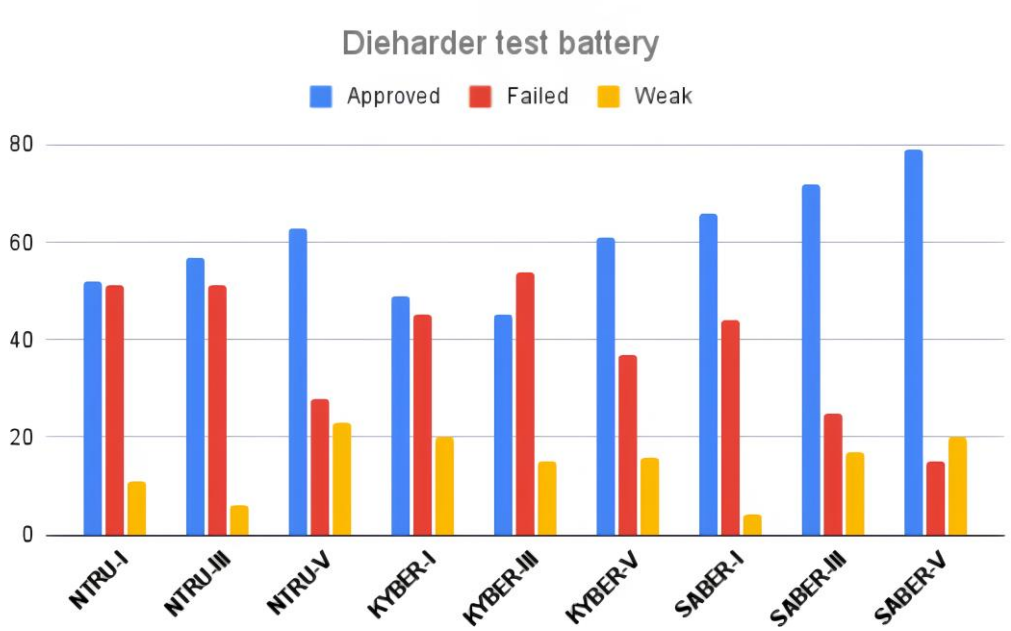

Fig. 5: Number of tests in Dieharder test battery

6. CONCLUSION AND FUTURE WORK

In this work, comparisons were made on the candidates of the third round of the post-quantum public-key cryptography pattern lattice-based. In the simulations, it was possible to observe that the SABER family algorithms obtained better performance in the num-ber of cycles and execution time. Their performance in terms of key size is very close to the other algorithms.

Based on the results observed, it was verified that the SABER scheme obtained the best results in a software environment using the reference versions of the algorithms. Checking the results in re-lated works based on hardware platforms, we can see that differentperformances can be obtained depending on the version and im- plementation platform where the algorithms are used. In the worksevaluated and in this one, the SABER and KYBER algorithms ob-tained the best results. The assumption is that the close results is due to the fact that both algorithms share in their basic structure a similar fundamental problem of lattice-based cryptography.

In addition, evaluations were carried out regarding the randomness of the same algorithms to verify possible vulnerabilities that can serve as a basis for the application of cryptanalysis techniques. In the tests performed, the algorithms of the KYBER had the worst performance. Thus, in future works, a study could be carried out using the parameters in which the algorithm failed to compose a cryptographic pattern recognition model.

Another possibility could be to carry out the simulations on different platforms, such as embedded systems, so it is possible toevaluate the performance of the algorithms in different situations. Furthermore, it would be essential to examine the different versionsof each algorithm in order to verify the best application of each.

7. REFERENCES

[1] RL Rivest, A Shamir, and L Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”.In: *Communications* (1978).

[2] Taher ElGamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[3] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”.In: *SIAM review* 41.2 (1999), pp. 303–332.

[4] Hamid Nejatollahi et al. “Post-quantum lattice-based cryp- tography implementations: A survey”. In: *ACM ComputingSurveys (CSUR)* 51.6 (2019), pp. 1–41.

[5] Ayesha Khalid et al. “Lattice-based cryptography for IoT ina quantum world: Are we ready?” In: *2019 IEEE 8th inter- national workshop on advances in sensors and interfaces (IWASI)*. IEEE. 2019, pp. 194–199.

[6] Hyeongcheol An et al. “Performance evaluation of liboqs in open quantum safe project (part i)”. In: *2018 Symposium on Cryptography and Information Security (SCIS 2018). IE-ICE Technical Committee on Information Security*. 2018.

[7] Jon Barton et al. “Post Quantum Cryptography Analysis ofTLS Tunneling on a Constrained Device”. In: (2022).

[8] Routo Terada and Eduardo T Ueda. “A New Version ofthe RC6 Algorithm, Stronger against $\chi 2$ Cryptanalysis.” In:*AISC*. 2009, pp. 47–52.

[9] Oded Regev. “Lattice-based cryptography”. In: *Annual International Cryptology Conference*. Springer. 2006, pp. 131–141.

[10] Miklós Ajtai. “Generating hard instances of lattice problems”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 99–108.

[11] Daniele Micciancio and Oded Regev. “Lattice-based cryptography”. In: *Post-quantum cryptography*. Springer, 2009,pp. 147–191.

[12] Oded Regev. “The learning with errors problem”. In: *Invited survey in CCC* 7.30 (2010), p. 11.

[13] Abhishek Banerjee, Chris Peikert, and Alon Rosen. “Pseudorandom functions and lattices”. In: *Annual InternationalConference on the Theory and Applications of Crypto-graphic Techniques*. Springer. 2012, pp. 719–737.

[14] Andrew Rukhin et al. *A statistical test suite for random and pseudorandom number generators for cryptographic appli- cations*. Tech. rep. Booz-allen and hamilton inc mclean va,2001.

[15] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. "NTRU: A ring-based public key cryptosystem". In: *International algorithmic number theory symposium*. Springer. 1998, pp. 267–288.

[16] Cong Chen et al. "Algorithm specifications and supporting documentation". In: *Brown University and Onboard security company, Wilmington USA* (2019).

[17] Roberto Avanzi et al. "CRYSTALS-Kyber algorithm spec-ifications and supporting documentation". In: *NIST PQC Round* 2.4 (2017).

[18] Jan-Pieter D'Anvers et al. "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM". In: *International Conference on Cryptology in Africa*. Springer. 2018, pp. 282–305.

[19] Michiel Van Beirendonck et al. "A side-channel-resistant implementation of SABER". In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 17.2 (2021), pp. 1–26.

[20] Douglas Stebila and Michele Mosca. "Post-quantum key exchange for the internet and the open quantum safe project". In: *International Conference on Selected Areas inCryptography*. Springer. 2016, pp. 14–37.

[21] Robert G Brown, Dirk Eddelbuettel, and David Bauer. "Dieharder". In: *Duke University Physics Department Durham, NC* (2018), pp. 27708–0305.