IMPROVING PERFORMANCE, CRYPTOGRAPHIC STRENGTH OF THE POST-QUANTUM ALGORITHM NTRUENCRYPT AND ITS RESISTANCE TO CHOSEN-CIPHERTEXT ATTACKS

ELENA REVYAKINA[1*], LARISA CHERCKESOVA[2], OLGA SAFARYAN[3], NIKITA LYASHENKO[4]

[1,2,3,4]Don State Technical University, 1 Gagarina Square, Rostov-on-Don, 344003, Russia
E-mail: elena.a.revyakina@gmail.com

ABSTRACT

This work is devoted to the development of modification of the post-quantum NTRUEncrypt algorithm to improve its performance and resistance to modern cyberattacks which in turn allows it to be used in modern practical applications. To accomplish this goal, the authors take an approach that involves optimization of polynomial multiplication, which is the most computationally complex part of the algorithm. The Anatoly Karatsuba's algorithm has been successfully applied to significantly increase the speed of key generation and encryption. In spite of the fact that this algorithm involves recursion, it allows to improve performance of the NTRUEncrypt algorithm, since it allows to speed up polynomial multiplication. Another notable improvement is developing the countermeasure against chosen ciphertext cyberattack. Resistance against this type of cyberattacks can be accomplished by employing cryptographic hash function to reject messages sent by malicious users. Performance test is carried out to estimate the average time required to generate keys and perform both encryption and decryption of a message. From the results of the performance test, it has been concluded that in the implemented modification of the algorithm, key generation, encryption and decryption require less time compared to the classical algorithm. The most significant performance gain has been achieved for the key generation stage since it involves numerous complex computations that can be performed much faster due to utilizing the Karatsuba's algorithm. Based on the modified algorithm, an asymmetric encryption system with a graphical interface has been implemented, which allows users to transfer messages with ensured protection against all modern attacks, including quantum cyberattacks.

Keywords: *Post-Quantum Algorithm, NTRUEncrypt Cryptosystem, Cyberattack, Polynomial Multiplication, Karatsuba Algorithm.*

1. INTRODUCTION

In an increasingly digital world, the security of sensitive information and communication is of importance. The increasing shift towards digitalization has impacted various aspects of people lives, including personal communication, financial transactions, healthcare data management, and government operations. This has resulted in the creation of an extensive digital landscape where confidential information is constantly being exchanged.

Within this evolving landscape, the important objective is to develop robust cryptographic frameworks and techniques capable of withstanding sophisticated threats and attacks. And cryptography plays a central role in addressing these concerns. It provides the tools and techniques necessary to safeguard information from unauthorized access, interception, and tampering. Cryptography is becoming more important in the modern world as a result of the development of information technologies, which leads to increase in the frequency and volume of cyberattacks. With the advent of quantum computers, it became possible to carry out quantum cyberattacks, which were previously considered only as a theoretically possible vulnerability of cryptographic algorithms.

The development of quantum cyberattacks has led to the fact that widely used asymmetric encryption algorithms such as Rivest–Shamir–Adleman algorithm (RSA) and ElGamal algorithm no longer provide the required security, as they can be broken within short period time using quantum computer. This fact has resulted in the emergence of post–quantum cryptography, which is the new class of algorithms resistant to cyberattacks, the implementation of which uses quantum computer. It is important to note that any post–quantum algorithm should also be resistant to the classical cyberattacks that do not involve the use of quantum computing.

The main purpose of this research is development of modification of the NTRUEncrypt algorithm, which will provide higher performance, in comparison with the standard version of the algorithm, and will be more resistant to modern cryptographic cyberattacks of high power, including quantum attacks. Based on the purpose of the research, the following objectives were determined:

- To investigate the vulnerabilities of the NTRUEncrypt algorithm to modern cyberattacks and understand the specific attack vectors that threaten its security.
- To develop a modified version of the NTRUEncrypt algorithm and justify of its advantages in comparison with its standard classical version.
- To implement and compare the performance of the modified NTRUEncrypt algorithm against the classical version, focusing on key generation, encryption, and decryption processes.
- To create a software tool based on the modified NTRUEncrypt algorithm, providing users with a secure means of encrypting and exchanging messages in the face of modern cryptographic cyberattacks, including quantum threats.

2. METHODS

The research focused on optimizing polynomial multiplication, a computationally intensive aspect of the algorithm, and implementing countermeasures against chosen-ciphertext attacks.

The NTRUEncrypt algorithm served as the foundation for this research. Secure parameters were adapted to enhance resistance to cyberattacks, leading to its inclusion in cryptographic standards and competitions.

The research introduced a modification to NTRUEncrypt that combined two approaches for performance improvement. Firstly, a polynomial f = 1 + pF was utilized, ensuring f always has an inverse polynomial modulo p. Secondly, Anatoly Karatsuba's algorithm was employed for polynomial multiplication, significantly reducing computational complexity compared to the standard method. Parameters N, q, and p were selected judiciously to enhance cryptographic strength.

Software implementation of the modified NTRUEncrypt algorithm was undertaken, incorporating the optimized polynomial multiplication and protection against chosen-ciphertext attacks. The encryption module converted plaintext to ciphertext using the public key, while the decryption module utilized the private key for retrieval. Additional verification mechanisms ensured message integrity. Data analysis modules allowed for byte distribution analysis in plaintext and ciphertext.

A comparative analysis was conducted between the modified and standard NTRUEncrypt algorithms. Key generation, encryption, and decryption processes were evaluated across 500 test cases.

3. DESCRIPTION OF THE NTRUENCRYPT CRYPTOSYSTEM

The NTRUEncrypt post–quantum algorithm was developed and introduced by scientists Jeffrey Hoffstein, Jill Pipherand Joseph H. Silverman in 1996. After detailed analysis of the published algorithm, the secure parameters were adjusted to make the algorithm more resistant to cyberattacks. In 2011, the NTRUEncrypt algorithm was added to the IEEE P1361.1 cryptographic standard as an asymmetric encryption algorithm [1]. The algorithm was also applied for the 2016 NIST competition for creation of new standard of post–quantum cryptography [2]. The main criteria by which algorithms were selected are performance and resistance to modern attacks [3]. The NTRUEncrypt algorithm has been merged with the similar NTRU-HRSS-KEM algorithm under the general name NTRU. Due to its performance and resistance to quantum cyberattacks, the NTRUE algorithm was selected for further consideration in the third round of the NISTcompetition [4].

Compared to other asymmetric algorithms (such as RSA), NTRUEncrypt has better performance and in both encryption and decryption [5]. Another significant advantage of the NTRUEncrypt algorithm is its low memory requirement, which makes it possible to implement effectively the asymmetric encryption on devices with limited memory (for example, on mobile devices) [6]. All operations performed in the NTRUEncrypt cryptosystem use operations on the ring $Z[X] / (X_N – 1)$ of polynomials, for which the degree does not exceed $N – 1$[7]. It follows that any polynomial A that is used in NTRUEncrypt cryptosystem can be expressed by the following formula:

$$A = a_0 + a_1 X_1 + a_0 X_2 + ... + a_{N-1} X_{N-1}$$

where $a_0, a_1, a_2 ... a_{N-1}$ are integers.

There are three main parameters, the choice of which determines the cryptographic strength and performance of the cryptosystem. These parameters are denoted as N, p, q and must meet the following requirements:

1. N is a prime number;
2. $q > p$;
3. q and p are relatively prime numbers.

In addition to the three main parameters mentioned above, additional parameters df, dg and dr must be defined for the successful functioning of the algorithm. These parameters affect the properties of the polynomials, which are used in the key generation process and in the message encryption.

To demonstrate how the algorithm works, let us consider an example in which Bob needs to send some message to Alice. In order to transmit the message, Bob must first generate a private key, and then obtain a public key from the private key.

The public key must be forwarded to Alice, while ensuring its secrecy during this transmission is not required. After Alice has received the public key, it can be used for encryption the message for Bob. To obtain the public key, Bob must determine two polynomials f and g that satisfy the following conditions:

1. Among the coefficients of the polynomial f there must be exactly df coefficients equal to 1 (this is the additional parameter described above, which must be determined in advance) and *df*–1 coefficients equal to –1. The remaining values of the coefficients of the polynomial f must be equal to 0.

2. Necessary condition is the presence of inverse polynomials, both for the polynomial f and for the polynomial g, defined in advance.

3. There are also requirements for the coefficients values of the polynomial g: this polynomial must have *dg* coefficients that are equal to 1, and the same number of coefficients taking the value –1. Just as in the case of the coefficients of the polynomial *f*, the coefficients of the polynomial g should take the value 0 in cases where they are not equal to –1 or 1 [8].

The next step of the algorithm functioning is to calculate the inverse polynomials for the polynomials f and g. These polynomials are denoted as f_p and f_q, and relations are fulfilled for them, in which p and q represent the parameters of the NTRUEncrypt cryptosystem, shown below:

$$f \times f_p = 1(\mathrm{modp})$$

$$f \times f_q = 1(\mathrm{mod\,p})$$

the polynomials (f, f_p) represent the secret key. The public key (f, f_q) can be calculated using the following formula:

$$h = p \times f_q \times g\,(\mathrm{mod\,q})$$

After the public key generation is successfully completed, Bob sends it to Alice. After receiving the public key, Alice can encrypt her message m using the algorithm:

1. It need to present the Alice's message m as a polynomial with coefficients modulo p from the range *[–p/2, p/2*];

2. The polynomial r should be selected and chosen, which has the name "blinding polynomial". This polynomial must have *dr* coefficients equal to 1 and *dr* coefficients equal to –1. As in the case of polynomials *f* and *g*, the remaining coefficients must be equal to 0. After calculating the coefficients of the blinding polynomial, the ciphertext is calculated by the formula:

$$e = (r \times h + m)(\text{mod } q)$$

After receiving the encrypted message from Alice, Bob has to use his private key to obtain the source text. To do this, Bob calculates the polynomial a by the formula:

$$a = f \times e \, (\text{mod } q)$$

After that, Bob selects coefficients from the range *(–q/2, q/2],* which are used subsequently to calculate the polynomial *b* according to the formula:

$$b = a \, (\text{mod } p)$$

The original message m can be obtained by the formula:

$$m = fp \times b \, (\text{mod } p)$$

4. LITERATURE REVIEW

In the article [9], the researchers tested the modified algorithm on ARM Cortex–M4 and came to the conclusion that the use of TMP led to acceleration of the encryption process by 13%, and decryption–by 17%, compared with classical algorithm. In [10], the implementation of NTRUEncrypt cryptosystem for 8–bit AVR microcontrollers was presented. It highlights the efficiency benefits of NTRUEncrypt, its compliance with specific parameter sets, and the impressive clock cycle performance achieved for encryption and decryption using the ees443ep1 parameters. The key achievement is a novel hybrid technique for multiplication in a truncated polynomial ring, resulting in a new speed record for the arithmetic part of a lattice-based cryptosystem on AVR microcontrollers.

One of the possible approaches to improving the NTRUEncrypt algorithm is to employ multidimensional algebras. The main advantage of modifications based on this approach is higher cryptographic strength compared with the classical algorithm. In the article [11], the scientists have developed new modification QuiTRU of the algorithm NTRU, which utilizes transformations in five-dimensional algebra. In the article [12], three-dimensional algebra ("tripternion algebra") with basis $\{1,x,x^2\}$ was applied to improve the cryptographic strength of the NTRUEncrypt algorithm. Tripternion algebra was utilized to modify the mathematical structure of the public and private keys, as well as the encryption and decryption processes, with the objective of achieving higher security levels. In [13], the modification of NTRTE was developed, which is based on quaternion transformations. In this article, the authors proved that when using parameters that provide the cryptographic strength $2^x$for the standard algorithm NTRUE, the developed modification allows using the same set of parameters to achieve 2^{2x} cryptographic strength.

The NTRUEncrypt algorithm can also be used to enhance the algorithms of quantum key distribution. In the article [14], this approach was applied for strengtheningof the famous quantum algorithm BB84. Another important area of research is the application of the NTRU algorithm to homomorphic encryption. In the articles [15] and [16], the Fully Homomorphic Encryption (FHE) algorithm was implemented using NTRU cryptosystem. In the article [17], the possible cyberattack on this cryptosystem was analyzed, andas a result of thatthe authors came to the conclusion that this attack can be avoided if the parameters of the cryptosystem are chosen correctly. Building of blockchain based on the NTRUEncrypt algorithm also seems quite possible. The potential of this approach was demonstrated in the article [18], using the example of blockchain development for the Internet of Things.

As a result of the analysis of modern publications, it can be concluded that the NTRUEncrypt algorithm can be effectively applied in the various areas of modern cryptography. Its active research with the purpose of development of modification which improves its performance and cryptographic strength is continuing.

5. DEVELOPMENT OF NTRUENCRYPT MODIFICATION

The most expensive calculations part, when executing the NTRUEncrypt algorithm, are actions on polynomials. First of all, the multiplication of polynomials refers to costly computing. It follows from this that the main direction of research to improve the performance of the NTRUEncrypt algorithm is to accelerate actions on polynomials. There are two possible ways to achieve the required performance improvement:

1. Initially choice the polynomials, for which the multiplication operation will be less complicated, in terms of the cost of resources for the required calculations.

2. Optimization the polynomial multiplication algorithm in time.

The modification proposed in this article combines both approaches to achieve the greatest to achieve the greatest increase in performance. Instead of polynomial f, thepolynomial in the form $f = 1 + pF$ is used. In this case, it is guaranteed that f always has an inverse polynomial in modulo p:

$$f - 1 = f_p = 1 \bmod p$$

Since f_p=1, it becomes possible to simplify the secret key to f, instead of using the pair of polynomials (f, f_p) as a secret key.

Changing the selection procedure of polynomial leads to increase in performance, since there is no necessity to select f in such way as to guarantee that it has the inverse polynomial. In addition, when decrypting, it is no longer necessary to perform the multiplication by polynomial p, which also contributes to performance gain of the NTRUEncrypt algorithm.

To optimize the polynomial multiplication algorithm, Anatoly Karatsuba's algorithm is employed. This algorithm makes it possible to achieve faster multiplication of polynomials, since the computational complexity of the standard algorithm is $O(n^2)$, while for the Karatsuba algorithm it is $O(n^{log_2 3}) \approx O(n^{1.58})$ [19].

The transformations performed make it possible to reduce the problem of multiplying two long numbers to the problem of multiplying three numbers of shorter length. In the next step it need to apply the recursion for each of three multiplications, i.e. for each of these multiplication the Karatsuba algorithm should be applied. Despite the lower asymptotic complexity of the Karatsuba algorithm, it is necessary to take into account the impact of recursion accomplishment on the performance .

The correct choice of the parameters N, q and p is decisive in the implementation of the NTRUEncrypt algorithm, since the cryptosystem's cryptographic strength depends on the values of these parameters. There are three sets of parameters that ensures 256 bit security. The first set (N=1087, p=3 i=2048) has the smallest key size. The main advantage of the second set (N=1499, p=3, q=2048) is the best performance. The third set (N=1171, p=3, q=2048) was designed to minimize the value C, that can be calculated by the formula:

$$C = S \times t_2$$

where t is the total execution time of the algorithm, S is the sum of the lengths of the public and private keys [20].

One of the most significant vulnerabilities of NTRUEncrypt cryptosystem is the possibility of cyberattack based on chosen ciphertext. This attack can be implemented against NTRUEncrypt algorithm as follows [21]: The attacker calculates the ciphertext for some text using the public key (which is publicly available and allows any user to encrypt the message):

$$C(x) = y \times h + y$$

where y is an integer number, such that both $y < q/2$ and $2y > q/2$ and $y \bmod p = 0$ are satisfied, and h is the public key.

If there is private key, then generated message can be decrypted using the formula:

$$a = f \times C \bmod q = (y \times f \times h + y \times f) \bmod q = (y \times q + y \times f) \bmod q$$

Each of the coefficients of polynomials g and f belongs to the set *{–1, 0, 1},* which implies that all coefficients of the polynomial a belong to the set

$$\{0, y, -y, 2y, -2y\}.$$

When performing the calculations that involve the polynomials f and q, only those elements that are equal to *2y* or *–2y* are changing. This follows from the fact that the inequalities $y < q/2$ and $2*y > q/2$ are satisfied.

If the i^{-th}coefficient of the polynomial a is equal *2y,* then following equality is true:

$$a(x) \bmod q = y \times g + y \times f - q \times x_i$$

The following formula is used to calculate the message m:

$$m = (y \times fp \times g + y \times f \times f_p - q \times x_i \times f_p) \bmod p$$

From the fact that y mod $p = 0$ it follows that:

$$m = q \times x_i \times f_p \bmod p$$

This allows the attacker to calculate in further the private key *f* using the formula:

$$f = -q \times m - 1 \times x_i \bmod p$$

To counteract the aforementioned attack, the calculation of a cryptographic hash function has been added to the developed attack, as a result of which the calculation of the blinding polynomial must be performed according to the formula:

$$r(x) = H(m \,\|\, R)$$

where *H*–cryptographic–hash function, m–message, *R*–randomly generated bit sequence.

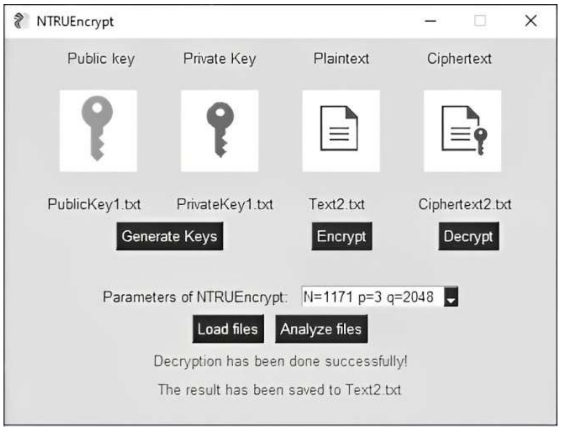
In the implemented modification, the sequence *R* has to be sent with the ciphertext. After receiving and decrypting the message, the additional verification of the message is required. This check of message consists of calculating the value $t = H(m(x)||R)$ and its comparison with the initial ciphertext. If the message is correct, the value of t matches the initial original message, otherwise the message is rejected because it was received from attacker who is carrying out the cyberattack.

6. SOFTWARE DEVELOPMENT

Based on the developed modification, which includes the optimization of polynomial multiplication and protection against cyberattacks with the chosen (selected) ciphertext, software implementation of the NTRUEncrypt algorithm was developed [21, 22].

The encryption module performs the conversion of plaintext into ciphertext using the public key. The decryption module allows to get the original plaintext from the ciphertext received from another user, using a private key. If two ciphertext are the same, then the decryption has been completed. If the texts are different, then the given message has to be rejected because it was sent by an intruder. The data analysis module allows to analyze byte distribution in both plain text and ciphertext to compare them and demonstrate correctness of the algorithm.

Figure 1 shows an example of decrypting the file Ciphertext2.txt received from the second user and successfully passing additional verification of the message.



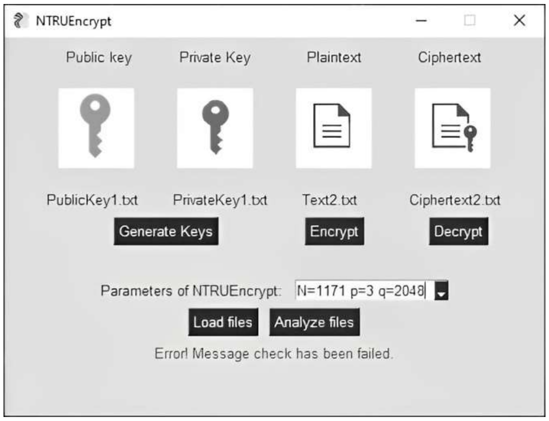
Parameters of NTRUEncrypt: N=1171 p=3 q=2048

Decryption has been done successfully!

The result has been saved to Text2.txt

Figure 1: Example of Successful Decryption of the File

Figure 2 shows an example of decryption the ciphertext for which the verification was not passed and failed, and as a result, the received text was rejected.



Parameters of NTRUEncrypt: N=1171 p=3 q=2048

Error! Message check has been failed.

Figure 2: Message in Case the Check of Ciphertext is not Passed

Figure 3 shows example of data analysis that involves calculating and visualizing distributions for both plaintext and ciphertext to make a comparison between them. The plaintext was written in English and encoded in UTF-8. Its size is 4096 bytes. Comparison between two histograms clearly demonstrates that the ciphertext has a random distribution of bytes. Thus, it can be concluded that the algorithm works correctly.

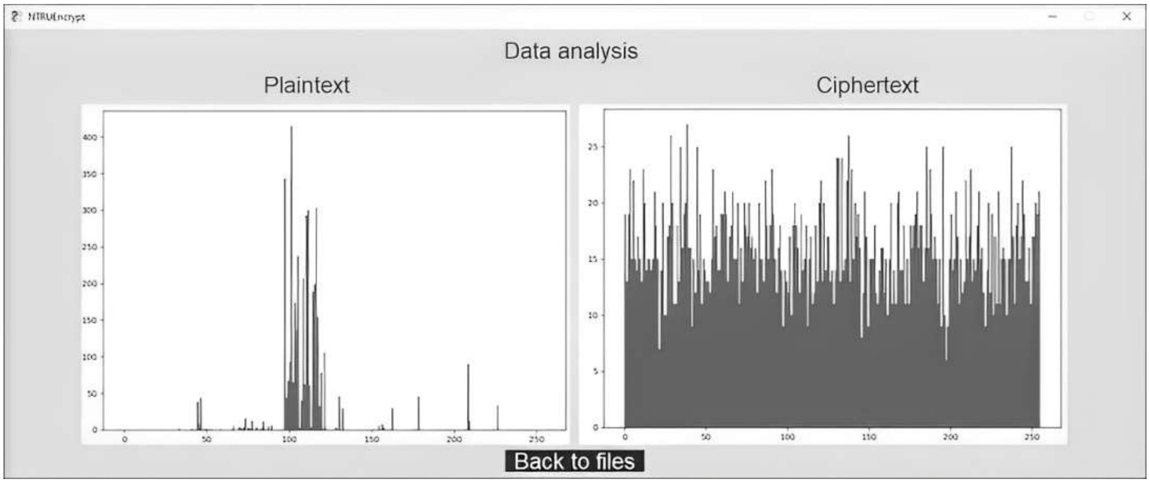


Figure 3: Byte Distributions for Plaintext and Ciphertext

The results of the work of modified and classical NTRUEncrypt algorithm were compared. To do this, the processes of key generation, encryption and decryption of messages were performed for 500 test examples. The average exectution time of key generation is 0,09381 seconds for the standart algorithm and 0,06743 seconds for the modified algorithm. The average encryption time is 0,05219 seconds for the standart algorithm and 0,03891 seconds for the modified algorithm. The average encryption time is 0,07476 seconds for the standart algorithm and 0,07139 seconds for the modified algorithm. For all three phases of the algorithm NTRUEncrypt, the developed modification results in better performance. The modification is 28,12% faster for the key generation, 25,44% faster for encryption and 4,51% faster for decryption.

From the results of implementation of testing, it was concluded that, in comparison with the standard NTRUEncrypt algorithm, the version modified by the authors has higher fast–action, performance and productivity, both in key generation and in the processes of encryption and decryption of messages.

7. CONCLUSION

The main result achieved in the course of the research is the modified NTRUEncrypt algorithm, which allows to achieve higher fast–action and better productivity, thanks to the optimization of the polynomial multiplication algorithm. The effectiveness of the developed modification has been confirmed by testing successfully. Another important significant advantage of the developed modification is protection against cyberattacks based on the selected ciphertext. Another important result is the development of software tool for encrypting data and transferring confidential information between users using the modified by authors algorithm.

Future research may focus on elevating the cryptographic strength, speed, and efficiency of the algorithm by implementing our modified NTRUEncrypt cryptosystem in low-level programming languages. Additionally, adapting this advanced cryptosystem for mobile devices, including laptops, tablets, and smartphones, will extend its reach and utility in an increasingly mobile-oriented world.

REFERENCES:

[1] E.H. Laaji, and A. Azizi, “Boosted performances of NTRUEncrypt post–quantum cryptosystem”, *Journal of Cyber Security and Mobility*, Vol. 10, No. 4, 2021, pp. 725–744. https://doi.org/10.13052/jcsm2245-1439.1045

[2] A. Hülsing, J. Rijneveld, J. Schanck, and P. Schwabe, “High–speed key encapsulation from NTRU”, in W. Fischer, and N. Homma (Eds.), *Cryptographic hardware and embedded systems – CHES 2017. CHES 2017. Lecture notes in computer science*, Vol. 10529, pp. 232–252. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-66787-4_12

[3] L. Malina, S. Ricci, P. Dobias, P. Jedlicka, J. Hajny, and K.-K.R. Choo, “On the efficiency and security of quantum-resistant key establishment mechanisms on FPGA platforms”, in *International Conference on Security and Cryptography*, Lisbon, Portugal, 2022, Vol. 1, pp. 605–613. https://doi.org/10.5220/0011294200003283

[4] F. Farahmand, M.U. Sharif, K. Briggs, and K. Gaj, “High-speed constant–time hardware implementation of NTRUEncrypt” SVES, in *2018 International Conference on Field-Programmable Technology (FPT)*, Naha, Japan, December 2018, pp. 190–197. https://doi.org/10.1109/FPT.2018.00036

[5] O.M. Guillen, T. Pöppelmann, J.M. Bermudo Mera, E. Fuentes Bongenaar, G. Sigl, and J. Sepulveda, “Towards post-quantum security for IoT endpoints with NTRU”, in *Design, Automation & Test in Europe Conference & Exhibition (DATE),* Lausanne, Switzerland, March 2017, pp. 698–703. https://doi.org/10.23919/DATE.2017.7927079

[6] E.H. Laaji, A. Azizi, and S. Ezzouak, “Two quantum attack algorithms against NTRU when the private key and plaintext are codified in ternary polynomials”, in M. Serrhini, C. Silva, and S. Aljahdali (Eds.), *Innovation in information systems and technologies to support learning research. EMENA-ISTL 2019. Learning and analytics in intelligent systems*, Vol. 7, pp. 551–562. Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-36778-7_61

[7] J. Hoffstein, J. Pipher, J.M. Schanck, J.H. Silverman, W. Whyte, and Z. Zhang, “Choosing parameters for NTRUEncrypt”, in H. Handschuh (Ed.), *Topics in cryptology – CT–RSA 2017. Lecture notes in computer science*, Vol. 10159, pp. 3–15. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-52153-4_1

[8] L.V. Cherckesova, O.A. Safaryan, P.V. Razumov, V. Kravchenko, S. Morozov, and A. Popov, “Post–quantum cryptosystem NTRUEnCrypt and its advantage over pre–quantum cryptosystem RSA”, *E3S Web of Conferences*, Vol. 224, 2020, Art. No. 01037. https://doi.org/10.1051/e3sconf/202022401037

[9] İ. Keskinkurt Paksoy, and M. Cenk, “Faster NTRU on ARM cortex–M4 with TMVP–based multiplication”, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 69, No. 10, 2022, pp. 4083–4092. https://doi.org/10.1109/TCSI.2022.3191111

[10] H. Cheng, J. Großschädl, P. Rønne, and P.Y.A. Ryan, “AVRNTRU: Lightweight NTRU–based post–quantum cryptography for 8-bit AVR microcontrollers”, in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, February 2021, pp. 1272–1277.

https://doi.org/10.23919/DATE51398.2021.9474033

[11] H.R. Yassein, H.N. Zaky, H.H. Abo-Alsoo, I.A. Mageed, and W.I. El-Sobky, "QuiTRU: Design secure variant of NTRUENcrypt via new multi–dimensional algebra", *Applied Mathematics & Information Sciences and International Journal*, Vol. 17, No. 1, 2022, pp. 49–53. https://doi.org/10.18576/amis/170107

[12] S. Shahhadi, and H. Yassein, "NTRsh: New secure variant of NTRUEncrypt based on tripternion algebra", *Journal of Physics: Conference Series*, Vol. 1999, 2021, Art. No. 012092. https://doi.org/10.1088/1742-6596/1999/1/012092

[13] R. Hassan, A. Nadia, K. Alaa, "Multi–dimensional algebra for designing improved NTRU cryptosystem", *Eurasian Journal of Mathematical and Computer Applications*, Vol. 8, No. 4, 2020, pp. 97–107. https://doi.org/10.32523/2306-6172-2020-8-4-97-107

[14] E.H. Laaji, and A. Azizi, "Combination of BB84 quantum key distribution and an improved scheme of NTRU post-quantum cryptosystem", *Journal of Cyber Security and Mobility*, Vol. 11, 2022, pp. 673–694, https://doi.org/10.13052/jcsm2245-1439.1152

[15] C. Bonte, I. Ilyashenko, J. Park, H.V.L. Pereira, and N.P. Smart, "FINAL: Faster FHE instantiated with NTRU and LWE", in S. Agrawal, and D. Lin (Eds.), *Advances in cryptology - ASIACRYPT 2022. Lecture notes in computer science*, Vol. 13792, pp. 188–215. Springer, Cham, 2022. https://doi.org/10.1007/978-3-031-22966-4_7

[16] W. Dai, W. Whyte, and Z. Zhang, "Optimizing polynomial convolution for NTRUEncrypt", *IEEE Transactions on Computers*, Vol. 67, No. 11, 2018, pp. 1572–1583.

[17] P. Kirchner, and P. Fouque, "Revisiting lattice attacks on overstretched NTRU parameters", in J.S. Coron, and J. Nielsen (Eds.), *Advances in cryptology – EUROCRYPT 2017. EUROCRYPT 2017. Lecture notes in computer science*, Vol. 10210, pp. 3–26. Springer, Cham. 2017. https://doi.org/10.1007/978-3-319-56620-7_1

[18] F. Wu, B. Zhou, and X. Zhang, "Identity–based proxy signature with message recovery over NTRU lattice", *Entropy*, Vol. 25, 2023, Art. No. 454. https://doi.org/10.3390/e25030454

[19] J. Sepulveda, A. Zankl, and O. Mischke, "Cache attacks and countermeasures for NTRUEncrypt on MPSoCs: Post–quantum resistance for IoT", in *30th IEEE International System-on-Chip Conference (SOCC)*, Munich, Germany, September 2017, pp. 120–125. https://doi.org/10.1109/SOCC.2017.8226020

[20] T. Kim, and M. Lee, "Efficient and secure implementation of NTRUEncrypt using signed sliding window method", *IEEE Access*, Vol. 8, 2020, pp. 126591–126605. https://doi.org/10.1109/ACCESS.2020.3008182

[21] L.V. Cherckesova, O.A. Safaryan, A.N. Beskopylny, and E. Revyakina, "Development of quantum protocol modification CSLOE–2022, increasing the cryptographic strength of classical quantum protocol BB84", *Electronics*, Vol. 11, No. 23, 2022, Art. No. 3954. http://dx.doi.org/10.3390/electronics11233954

[22] N. Lyashenko, L. Cherckesova, E. Revyakina, D. Medvedev, and A. Gavlitsky, "Development of modification of the post-quantum public-key cryptosystem NTRUENCRYPT", *E3S Web of Conferences*, Vol. 389, 2023, Art. No. 07013. https://doi.org/10.1051/e3sconf/202338907013