

Development and Implementation of the NTRUEncrypt Public Key Cryptographic System

Elena A. Revyakina¹, Larisa V. Cherckesova¹ and Elena A. Menshenina²

¹Don State Technical University, Rostov-on-Don, Russia

²Admiral Ushakov Maritime State University, Rostov-on-Don, Russia

Keywords: Cryptographic System, NTRUEncrypt, Document Encryption, Information Security.

Abstract: The article describes the development of a software tool for encrypting a text document and provides a number of formulas that were used in this case. It also describes the process of generating a private and public encryption key in the software, provides formulas for generating keys, and shows the user interface for entering the necessary parameters. In addition, the algorithm for decrypting the encrypted text was described and the output of the program results after decrypting the selected document was shown. A comparative analysis of the speed of key generation, encryption and data decryption of the implemented NTRUEncrypt cryptographic system with RSA and ECC is performed.

1 INTRODUCTION


In today's world, technology has reached the point where a large amount of information is stored in cyberspace, and thanks to this type of storage, it is easier for people around the world to access information than ever before. Until the end of the last century, collecting information was a difficult and long task, but now people can access a huge amount of information in a few minutes, and with the development of high technologies, more and more organizations in the world now depend on their information systems and on the information security of these systems. Information security means the protection of information and information systems from unauthorized access, recording, use, reading, destruction or modification. Cryptography is used in information security to protect information from unauthorized or accidental disclosure, during the transfer of information, or while the information is in storage. Cryptography is inextricably linked to the transmission of data, and in addition to providing user authorization, it guarantees the integrity of the transmitted information and confidentiality. It is obvious that for organizations such as banks, military departments, government agencies, the use of


cryptographic systems is necessary, and cryptography is used by various institutions on a daily basis.


A cryptographic system is a system for ensuring the security of a protected system that uses cryptographic means and methods. There are two main types of cryptographic systems: symmetric and asymmetric. Symmetric key systems require that both the sender and the receiver have the same key.

This key is used by the sender to encrypt the data and again by the receiver to decrypt the data. Each user has both a public and a private key. Messages are encrypted with one key and can only be decrypted with another key. With the development of quantum computers, the risk of hacking such pre-quantum public-key cryptosystems as RSA (a cryptosystem named after its three inventors Rivest, Shamir, and Adleman), elliptic curve cryptosystems (ECC), also increases.

Therefore, post-quantum encryption systems are becoming more relevant than ever. The NTRUEncrypt public key cryptographic system can remain relevant in the post-quantum era due to the fact that this system is based on a lattice cryptosystem and its stability is ensured by the complexity of finding the shortest lattice vector. Until now, there is no algorithm that would solve this problem, so the

^a <https://orcid.org/0000-0003-1577-2671>

^b <https://orcid.org/0000-0002-9392-3140>

^c <https://orcid.org/0000-0003-4487-3498>

NTRUEncrypt cryptographic system can remain relevant not only at the moment, but also in the future.

The purpose of this work is to develop and implement the NTRUEncrypt cryptographic system with a public key.

2 QUANTUM COMPUTERS

Quantum computers are machines that use the properties of quantum physics to store data and perform calculations. This can be extremely useful for certain tasks, where they can significantly outperform even supercomputers. Classic computers, including smartphones and laptops, encode information in binary "bits", which can have a value of 0 or 1. In a quantum computer, the basic unit of memory is the qubit (Wenbo Mao, 2015).

Bouquets are created using physical systems, such as the spin of an electron or the orientation of a photon (Uttar Pradesh, 2019). These systems can be simultaneously arranged in many different configurations, a property known as quantum superposition. Qubits can also be inextricably linked together by a phenomenon called quantum entanglement. As a result, a series of qubits can simultaneously represent different things.

For example, eight bits are enough for a classical computer to represent any number from 0 to 255. But eight dice is enough for a quantum computer to represent every number from 0 to 255 simultaneously. (Uttar Pradesh, 2019). A few hundred entangled qubits will be enough to represent more numbers than the atoms in the universe. In situations where there are a large number of possible combinations, quantum computers can consider them simultaneously (Ryabko, 2012). Examples include trying to find the prime factors of a very large number, or the best route between two places.

However, there may also be many situations where classical computers will still outperform quantum computers. The computers of the future may be a combination of both of these types. At the moment, quantum computers are very sensitive: heat, electromagnetic fields and collisions with air molecules can lead to the loss of the quantum properties of the qubit. This process, known as quantum decoherence, causes the system to fail, and this happens the faster the more particles involved (Schneier B., 2002).

Creating hardware for a quantum computer is a difficult task. Qubits are inherently very fragile and can lose the information encoded in them very quickly. The main objective is to keep the qubits

completely isolated from the environment, while providing high-precision monitoring and reading of the qubit state. In order to effectively separate the qubits from any noise source and therefore maintain a longer coherence time, these systems are usually cooled to extremely low temperatures using liquid helium, however this results in high operating costs.

There are many different ways to implement qubits, such as trapped ions, superconducting rings, and more. Each architecture has its own advantages and disadvantages, and it is not yet clear which qubit material is the most scalable (Ishmukhametov, 2011).

To provide a large number of parallel processes, the quantum computing model uses several relatively simple rules for converting input information. In other words, it becomes possible to calculate its values for all arguments in a single pass of the function. This function is applied to the input data. The input is data that is a superposition of all possible values of the argument, therefore, the function must accept and process such a sequence. In quantum models, a function is a Hermitian matrix, that is, one that has its own Hermitian-conjugate, and the product of these matrices will eventually give a unit one. The conjugate matrix can be obtained in two steps, the first step is the transposition of the original matrix and the second step is the substitution of their complex-conjugate elements instead of the original elements. If you multiply the resulting matrix by the vector of the quantum register, you get a quantum register, such that the sum of the squares of the coefficients for quantum states is equal to one (Cheremushkin, 2009).

2.1 Comparison of RSA, NTRUEncrypt, and ECC Cryptosystems

Most of the most important information today is transmitted via the global Internet, for example: email, chat messages, video conferences, e-commerce, and online banking data. The use of public-key cryptosystems is the main way to protect such data (Bolotov, 2006).

The most widely used cryptosystem is RSA, based on the complexity of factorization of large numbers, the Diffie-Hellman scheme, the digital signature algorithm (DSA), the security of which is based on the complexity of solving the discrete logarithm problem in the field, a family of algorithms based on elliptic curves. But all of them have certain disadvantages, the main ones being either relatively low speed, or relatively low stability at comparable key and parameter sizes (Diffie-Hellman scheme and other algorithms based on the discrete logarithm in

the field), or both at the same time (RSA) (Zhdanov O.N. et al., 2012).

The NTRU algorithm, all operations of which are performed in the ring of truncated polynomials, was created to solve this problem. The cryptographic strength of NTRU is based on the complexity of the problem of finding a short vector in a given lattice (Bertels, K., 2018). All cryptographic systems based on integer factorization, discrete logarithm, and discrete logarithm problems in the elliptic point group of curves are potentially vulnerable to the development of a quantum computer of suitable size, since algorithms are known for such a computer that can solve these problems in polynomial time, depending on the size of the input data. Currently, there are no quantum algorithms with polynomial complexity for NTRU.

Only cryptosystems based on algebraic lattices remain virtually invulnerable to quantum cryptanalysis. One of the main advantages of the NTRU algorithm is also the very high speed of encryption / decryption operations. According to the developer company NTRU Security Innovation, this algorithm is up to two hundred times faster than the elliptic curve and RSA algorithms [12]. Based on the currently available data, we can draw intermediate conclusions about the high level of NTRU security, which is not inferior to the security of algorithms based on elliptic curves.

However, due to the relative novelty and low prevalence of asymmetric algorithms of this class, additional research is needed for possible errors and critical vulnerabilities that can be used to develop effective attacks. As a result of the performance analysis of NTRU, it was found that its performance is much higher than that of RSA and ECC. NTRU showed the best results in terms of performance, as it lends itself very well to parallelization, in contrast to RSA, attempts to parallelize which practically do not give a speed increase.

The class of asymmetric algorithms based on problems on algebraic lattices is significantly superior to other public-key algorithms in the aggregate of all indicators. Therefore, it is NTRU that should replace RSA and ECC in many areas and become the same generally accepted standard for asymmetric cryptography.

2.2 Advantages of the NTRUencrypt System

The NTRUencrypt algorithm was developed back in the mid-90s. In comparison with the RSA cryptosystem, it was not widely used, since it was first necessary to improve the cryptographic strength and performance of this cipher algorithm.

At the moment, all the identified shortcomings have already been fixed, and in practice it is believed that NTRUencrypt works much faster than the RSA algorithm. When comparing the speeds of NTRU, RSA, and ECC with the parameters corresponding to the security level $k = 256$ bits, NTRU is 4 orders of magnitude faster than RSA and 3 orders of magnitude faster than ECC [10]. The graph (Figure 1) clearly shows how much the NTRUencrypt cryptosystem exceeds, in terms of the number of operations per second, most existing pre-quantum algorithms.

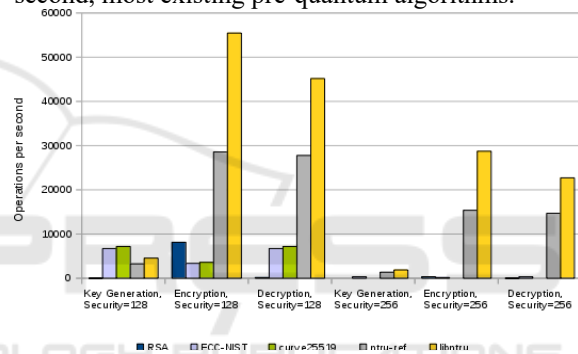


Figure 1: Comparison of NTRU performance with other algorithms.

In addition, the NTRUencrypt cryptosystem is post-quantum, and has a much higher cryptographic resistance to various attacks. And, as mentioned above, due to the lack of an algorithm for finding the shortest lattice vector, the cryptographic stability of the algorithm is ensured (NTRU Cryptolabs. NTRU Algorithms, 2020).

3 THE NTRUencrypt PUBLIC KEY CRYPTOGRAPHIC SYSTEM

Approaches to the implementation of the system.

NTRUencrypt, developed by Hofstein, Piefer, and Silverman, was first introduced to the public at the Crypto '96 supplemental session. At the moment, the NTRUencrypt encryption system is an alternative to encryption schemes based on factorization and

discrete logarithm over finite fields and elliptic curves, which is marked by its inclusion in the P1363 standard.

NTRUEncrypt is considered a viable solution for post-quantum public-key encryption. While the number of attacks and practical improvements of NTRUEncrypt grew, the theoretical part of cryptography with proven security, based on the use of lattices, stopped in development.

The starting point is 1996 and the recognized transformation of the worst-case scenario into normal, which gave rise to stable hash functions, cracking which is as difficult as cracking some problems over worst-case lattices. In the last couple of years, a large number of cryptographic schemes have been introduced that have proven the security of the error-based learning layer (LWE). This list includes secure encryption schemes based on authentication, digital signatures, etc.

The main obstacle to LTE-based cryptography and the Short integer solution (SIS) is its limited efficiency. The key in most cases contains a random matrix, the dimensions of the security parameter are linear; therefore, the space and time requirements are at least a second power with respect to the security parameter.

In 2002, Micciancio was able to convert SIS to structured matrices, while retaining the worst-case to normal transformation. The worst-case scenario is the reduction of general lattice problems to a special family of cyclic lattices. Micciancio's work led to the formation of a family of hash functions that are resistant to preimage attacks.

It is proved that the resulting hash function is collision-resistant for a given stability of the modified normal scenario problem; it was called Ideal-SIS. It provides the same robustness as reducing the worst-case scenario for typical lattices to a special class of lattices (called ideal ones). In 2009, Stele proposed a structured version of LTE, which proved to be as reliable as Ideal-SIZE (ShanYue Bu. Choosing Parameters for NTRU, 2020).

In parallel, Lubashevsky, in his independent work, proposed a variant of LWE on rings, called R-LEE, whose greater flexibility makes it possible to create more natural and efficient cryptographic constructions. Most modern cryptographic algorithms and protocols rely on the computational complexity of certain mathematical problems, such as factorization of the products of two large primes (RSA) or discrete logarithm over certain groups (Diffie-Hellman key exchange, El-Gamal encryption system).

These problems are considered to have no efficient (polynomial-time) solutions, so any cryptographic protocol based on them should be at least as difficult to crack. Since it can be assumed that any potential adversary has limited computing power, it can be expected that these protocols will be secure. However, this is not always the case.

If the opponent has the computing power of a quantum computer, such algorithmic problems are easily solvable. RSA and many other popular protocols, such as Diffie-Hellman key exchange, El Gamal encryption scheme, DES (data encryption standard), or ECC, will be broken.

They will need to be replaced with something more secure, involving more computing power of a potential enemy. There are several algorithmic problems that are difficult to solve, both on classical and quantum computers. There are many different groups of protocols, such as hash-or code-based cryptography, multidimensional quadratic equation cryptography, or lattice-based cryptography (Fergusson N., Schneier B. 2004).

NTRUEncrypt is usually described as a polynomial-based cryptosystem involving convolution products. It can naturally be considered as a lattice cryptosystem for a certain limited class of lattices. The cryptosystem has several natural parameters, and, as with all practical cryptosystems, it is possible to optimize these parameters to increase efficiency, while avoiding all known cryptanalytic attacks. The functionality of the cryptosystem will not be affected if "decryption failures" occur with a very small probability in random messages.

3.1 Recommended NTRUEncrypt Parameters

According to the developers of NTRUEncrypt, this encryption system is 4 orders of magnitude faster than RSA and 3 orders of magnitude faster than ECC and can be used in the "post-quantum" era [10-14].

The NTRUEncrypt public key cryptographic system is based on a lattice cryptosystem and its strength is ensured by the difficulty of finding the "shortest lattice vector". This system is designed for an alternative to RSA, ECC, as it is more resistant to attacks on quantum computers. NTRUEncrypt operates on polynomials of degree at most $N-1$:

$$a(X) = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}$$

NTRUEncrypt parameters:

N – the size of the public key
 q, p – mutually prime numbers that are used to reduce the coefficients of polynomials;

df – the number of coefficients equal to "1" and df-1 equal to "-1" of the polynomial f;
 dg – the number of coefficients equal to "1" and "-1" for the polynomial g;
 dr – the number of coefficients equal to "1" and "-1" for the "blinding" polynomial r.
 To achieve the desired level of protection, we recommend using the following NTRUEncrypt system parameters, which are shown in Table 1:

Table 1: Recommended NTRUEncrypt parameters.

N	q	p	df	dg	dr	Guaranteed durability
167	128	3	61	20	18	Moderate level of durability
251	128	3	50	24	16	Standard level of durability
503	256	3	216	72	55	Highest level of durability

3.2 Description of the Algorithm of the Program

At the first stage of the program, the private and public encryption keys are formed, which will be used in the future when encrypting and decrypting information. This process is shown in Figure 2.

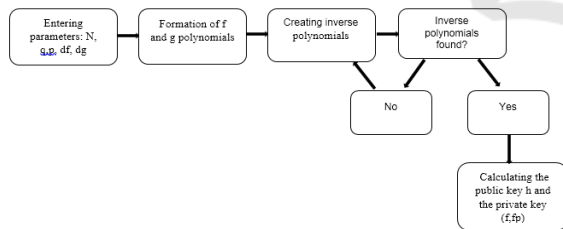


Figure 2: Public key generation scheme.

Next, the user selects the message to be encrypted using the public key. Figure 3 shows the encryption execution scheme.

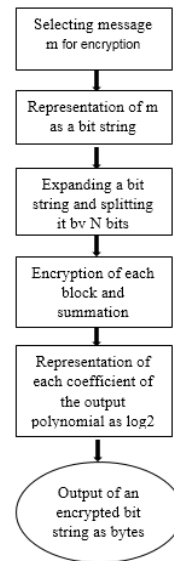


Figure 3: Encryption scheme for the selected message.

After the message has been encrypted, this software tool allows you to decrypt it using a secret key. In this case, the encrypted message is represented as bits and then the bit sequence is divided into blocks of N bits. After that, each block is decrypted and all the decrypted blocks are combined into a common array.

The last step in decrypting the encrypted data is to break the resulting bit string into characters and display it on the screen. Figure 4 shows a diagram of this process.

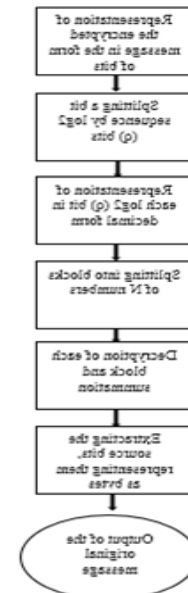


Figure 4: Scheme of decryption of the selected message.

Thus, based on the generalization of general information about the NTRUEncrypt cryptographic system, an algorithm for encryption and decryption by this system, as well as the formation of a public and private key, was developed. Below we will consider the principle of operation of the software product, which is an object-oriented application with a simple and intuitive interface.

4 TESTING AND DEMONSTRATION OF THE SOFTWARE TOOL

The NTRUEncrypt cryptographic system uses six constant parameters: N, p, q, df, dg, dr. These parameters are entered by the user in the "Encryption" panel, as shown in Figure 5.

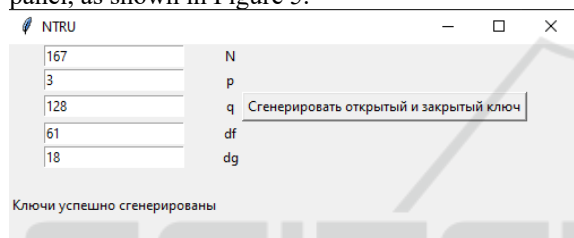


Figure 5: Entering parameters and generating private and public keys.

The number N characterizes the size of the public key, as well as the length of each data block. The integers p and q, whose GCD (Greatest Common Divisor) (p, q) is 1, do not have to be prime.

The p parameter is used to determine the interval to which all coefficients of the polynomials used in the cryptographic system should belong. The parameters df and dg specify the number of units in the program-generated polynomials f and g. The dr parameter specifies the number of coefficients equal to 1 and -1 of the "blinding" polynomial r, which is used for encryption.

First, the user enters the first five parameters, after which the formation of the polynomials f and g takes place. After that, we search for the inverse polynomials for f and g, but it may happen that some inverse polynomial will not be found. Then the program will form another polynomial and try to find the inverse for it again.

To achieve a sufficient level of cryptographic security, all parameters must meet the recommended ones.

Figure 6 shows the stored encryption keys in the "keys" folder. The public key "key. pub "is needed

for data encryption, and the key" key.priv " is needed to decrypt the received ciphertext.



Figure 6: Encryption Keys.

Next, the program will encrypt the file, save it as "encrypted. enc" and output the cipher after clicking on the "Encrypt" button as shown in Figure 7.

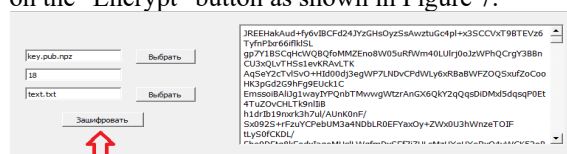


Figure 7: Output of the encrypted file.

The last step of the program is to select the encrypted document and the private key as shown in Figure 8. When decrypting, the encrypted text is again represented as bits. Then the bit sequence is represented as coefficients, after which each series of coefficients develops into blocks of N numbers. Each block is represented as a polynomial, and then decryption is performed to obtain a sequence of bits that are divided into bytes. Thus, the resulting text is the same as the original one.

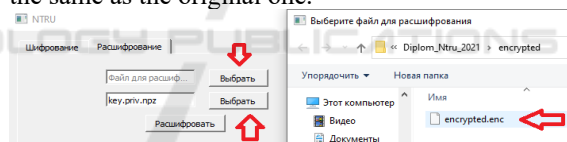


Figure 8: Selecting the ciphertext and private key.

After clicking on the "Decrypt" button, the decryption of the selected encrypted file will begin.

5 CONCLUSIONS

In this paper, we implemented a system that provides a resource for encrypting and decrypting data, the cryptographic strength of which is explained by the complexity of solving the problem of lattice theory, which is the basis of the NTRUEncrypt algorithm. In the framework of this work, the algorithm of the software tool was described: the choice of variables for generating the private and public encryption key, the algorithm for encrypting and decrypting information using the generated keys. In addition, the scheme of the program was built with a decryption of

each stage of the implementation of the implemented algorithm and demonstrated the operation of the software with the output of encrypted and decrypted messages.

The full implementation of the NTRUEncrypt cryptographic system was performed in the Python programming language. It is important to note the fact that after the creation of a quantum computer, the problems of fast discrete logarithm and factorization will soon be solved and such algorithms as RSA, DSA and other similar ones will become irrelevant.

However, the NTRUEncrypt cryptographic system will remain relevant, since there is no algorithm that solves the problem of the shortest lattice vector, which means that it is quite applicable in the "post-quantum" era. As a result of the work, a software implementation of the NTRUEncrypt algorithm with a user interface was developed, and the developed software tool was tested with various parameters.

REFERENCES

- Bertels, K. (2018). *Quantum Computer Architecture: Towards Full-Stack Quantum Accelerators*. K. Bertels, A. Sarkar, M. Serrao [et al.], IEEE Access. pages 102-492. DOI: 10.23919/DATE48585.2020.9116502.
- Bolotov, A. A., Gashkov, S. B. and Frolov, A. B. (2006). *Elementary introduction to elliptic cryptography. Algebraic and algorithmic foundations*, Chasovskikh-Moscow: KomKniga, p. 255.
- Cheremushkin A.V. (2009). *Cryptographic protocols: basic properties and vulnerabilities. Institute of Cryptography, Communications and Computer Science*. Textbook. Moscow: Stringer, page 116.
- Fergusson N., Schneier B. (2004). *Practical cryptography*. M.: Dialectics. P. 432.
- Ishmukhametov, Sh. T. (2011). *Methods of factorization of natural numbers: a textbook*. Kazan: Kazan University, p. 290.
- NTRU CryptoLabs. NTRU Algorithms (2020). – Text: electronic. NTRUEncrypt: <http://ntru.com/cryptolab/>
- Ryabko B. Ya. (2012). *Cryptographic methods of information protection: textbook. manual for universities*. 2nd ed., Moscow: Hotline. Telecom, page 229.
- Schneier, B. (2002). *Applied cryptography. Protocols, algorithms*, source texts in the C language. M.: Triumph, page 816.
- ShanYue, Bu. *Choosing Parameters for NTRU*, (2020). ShanYue Bu, Hui Zhang [et al.], pages 154-230. DOI: 10.1109/MINES.2009.133.
- Revyakina, E. (2020). Possibilities of conducting XSS-attacks and the development of countermeasures. Cherkesova L., Safaryan O., Korochentsev D., Boldyrikin N. and Ivanov Yu. [et al.]. electronic // IEEE Access. page.224 - E3S Web of Conferences 224, 01040 (2020) doi.org/10.1051/e3sconf/202022401040
- Weichi Yu et al. (2019). *Study on NTRU Decryption Failures*, pages 116-305. – DOI: 10.1109/ICITA.2005266.
- Wenbo Mao (2015). *Modern cryptography*. M. Venbo: Moscow: Publishing house "Williams" page 768.
- Zhdanov O.N. et al. (2012). *Fundamentals of theory and cryptographic applications*. SSAU named after M. F. Reshetnikov. Moscow: URSS LIBROCOM,