КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2020.10.123134
UDC 004.03

Ilyenko Anna
Candidate of Technical Sciences, assistant professor , assistant professor of Information Security Systems Department National Aviation University of Kyiv, Faculty of Cyber Security, Computer and Software Engineering, Ukraine
ORCID: 0000-0001-8565-1117
Ilyenko.a.v@nau.edu.ua

Ilyenko Sergii
Candidate of Technical Sciences, assistant professor , assistant professor of Automation and Energy Management Department National Aviation University of Kyiv, Aerospace Faculty, Ukraine
ORCID: 0000-0002-0437-0995
Ilyenko.s.s@nau.edu.ua

Prokopenko Olena
Assistant of Information Security Systems Department National Aviation University of Kyiv, Faculty of Cyber Security, Computer and Software Engineering, Ukraine
ORCID: 0000-0001-9895-888X
Bortnik.olena.v@nau.edu.ua

THE IMPROVEMENT OF NTRUENCRYPT PUBLIC KEY CRYPTOSYSTEM: DESIGN AND PERFORMANCE EVALUATION

Abstract. Today cryptographic systems provide secure communication between users. In the present paper we describe existing cryptographic systems such as: systems based on the complexity of factorization of a large integer (RSA); systems based on the complexity of solving a discrete logarithm in finite Galois field (eigamal, DSA); systems based on the complexity of solving a discrete logarithm in a group of points of an elliptic curve (ECC); lattice-based systems (NTRU). Authors focus their attention on ntruencrypt encryption and decryption algorithm. The ntruencrypt public key cryptosystem guarantees the integrity and confidentiality of information when transmitting, storing and processing information messages in modern computer systems and networks. The conducted studies of public key cryptosystem made it possible to determine the way of the improve ntruencrypt public key cryptosystem. In this paper, we present improved ntruencrypt public key cryptosystem which is based on the correct selection of parameters *p* and *q*. The authors concluded that, to reduce the difference between the length of ciphertext and plaintext, it is necessary to take *p* and *q* closer to each other. At the same time it is necessary to consider that at too close values *p* and *q* the cryptosystem can be weakened. The main difference between the proposed schemes was the reducing the size of ciphertext which can minimizes the time for software encryption and decryption operations. As a result is a software implementation of the procedure for the encryption and decryption of the improve ntruencrypt public key cryptosystem using a programming language Ruby 1.9.3 was obtained using the cryptolib library. Improved algorithm will be a perfect tool for ensuring the confidentiality of information, using “cloud” computing, because protecting information from unauthorized access is one of the most pressing problems. The authors further plan a number of scientific and technical solutions to develop and implement effective methods, tools to meet the requirements, principles and approaches to cyber security and cryptosystems for provide integrity and confidentiality of information in experimental computer systems and networks.

Keywords: Public key cryptosystem, Integrity, Confidentiality, Encryption, Ciphertext.

1. INTRODUCTION

Our time is characterized by a real "information boom", the unstoppable growth of entire data sets in various areas of activity that need protection from unauthorized access, the

deepening of complex problems of information security. Their solution provides a systematic complex approach, an important part of which are the methods of cryptography.

Asymmetric cryptography is widely used to encrypt messages, as electronic digital signatures system, in network protocols, and many other protocols. Most modern cryptographic "giants", such as RSA and ECC, can be easily hacked on quantum computers, which time has shown is "just around the corner." Another threat to them is the rapid development of discrete logarithm. This is explained by the fact that with the advent of the quantum computer it will be possible to implement the Shore algorithm, which allows to solve the problem of factorization and discrete logarithm. It becomes clear that RSA, ECC and other similar ciphers are not so secure. However, the disadvantages of these cryptosystems are not typical for the relatively young NTRU cryptosystem, based on the lack of algorithms, including quantum, to find the shortest lattice vector, and therefore it is quite acceptable in the "post-quantum" era.

The NTRU algorithm stands out for simple calculations, does not require large numbers and floating point operations, so it is ideal for processors and microcontrollers of all types of architectures. The mathematical model of the cryptosystem also has advantages over competitors: with the same key size, the NTRU is more secure and performs calculations several times faster.

The purpose of this article is to analyze and consider the practical bases of the asymmetric encryption and decryption system. In this paper, we propose an improved *ntruencrypt* public key cryptosystem due to the selection of parameters *p* and *q*. The main difference between the proposed scheme was the reducing the size of ciphertext which can minimizes the time for software encryption and decryption operations and increases cryptographic resistance.This approach allows solving the problem of information protection, including stored information, processed and transmitted in modern information networks on the basis of confidentiality and integrity.

2. RELATED WORK

The concept of public key cryptography, or asymmetric cryptography as it is called, was put forward by Whitfield Diffie and Martin Hellman, and independently by Ralph Merkel. Their contribution to cryptography was the belief that keys could be used in pairs - an encryption key and a decryption key - and that it might be impossible to obtain one key from another. Diffie and Hellman first presented this idea at the 1976 National Computer Conference [1], a few months later, their fundamental work "New Directions in Cryptography" was published [2]. Due to the impartiality of the publication process, Merkel's first contribution to this area appeared only in 1978 [3]. Since 1976, many cryptographic algorithms with public keys have been proposed. Many of them are unstable, others are unsuitable for practical implementation or use too large a key, or the size of the received ciphertext is much larger than the size of plaintext.

Few algorithms are both secure and practical. Usually these algorithms are based on one of the difficult problems. Some of these secure and practical algorithms are only suitable for key distribution. Others are suitable for encryption (and for key distribution). The third are only useful for digital signatures. Today there are the following asymmetric cryptosystems: 1) systems based on the complexity of factorization of a large integer (RSA); 2) systems based on the complexity of solving a discrete logarithm in finite Galois field (eigamal, DSA); 3) systems based on the complexity of solving a discrete logarithm in a group of points of an elliptic curve (ECC); 4) lattice-based systems (NTRU) [6]. All these cryptosystems belong to

the class of probable-stable. To improve cryptographic strength, system designers are constantly increasing the size of the system-wide parameters for these algorithms. They encrypt and decrypt data much more slowly than symmetric algorithms. Usually their speed is insufficient to encrypt large amounts of data.

Public key algorithms are designed to withstand disclosure with the selected public text. Their security is based on both the difficulty of obtaining a private key in public and on obtaining public text in ciphertext [4].

The Diffie-Hellman algorithm [2] helps the secret key exchange for symmetric cryptosystems. The range of possibilities of this algorithm is determined by the fact that two parties need to perform a confidential determination, and in their distribution there is no initially determined secret key. However, there is a channel between them that is protected from modification, ie the data transmitted on it can be listened to, but not changed, such conditions occur quite often. In this case, the two parties can create the same secret key, never transmitting it over the network, according to the following algorithm.

At the moment, the RSA algorithm [5] seems extremely reliable. It survived more than 20 years of study and has been widely recognized in the world. The attack that RSA is most often exposed to is public key factorization. If the attack is successful, all messages written with this public key can be decrypted. The fact is that with very large numbers, factorization takes too much time. It has not been proven that breaking the RSA algorithm is equivalent to large numbers factorization (there may be another, simpler way), but the opposite has not been proven either.

Various public key algorithms are used for EDS, among which there is a specially developed DSA.

Elgamal scheme [6,7] can be used for both digital signatures and encryption, its security is based on the complexity of calculating discrete logarithms in the finite field.

Ntruencrypt, originally called NTRU, was introduced in 1996 and presented worldwide at conferences [8]. The reason that served as the beginning of development of the algorithm in 1994, was an article that talked about the ease of hacking existing algorithms on quantum computers, which, as time has shown, are not far off. In the same year, mathematicians Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman, who co-developed the system with Daniel Liemann, founder of NTRU Cryptosystems, Inc. (later renamed to securityinnovation), patented their invention [9,10].

In 1985, Neal Koblitz and Victor Miller independently suggested the use of elliptic curves for public key cryptosystems [11,12]. They did not invent a new cryptographic algorithm that uses elliptic curves over finite fields, but implemented existing algorithms, similar to Diffie-Hellman, using elliptic curves.

Ntruencrypt cryptosystem. Ntruencrypt operates over polynomials of degree at most $N - 1$

$$a = a_0 + a_1 X + a_2 X^2 + ... + a_{N-2} X^{N-2} + a_{N-1} X^{N-1},$$

Where the coefficients are integers. Regarding the operations of addition and multiplication modulo of polynomial X^{N-1} these polynomials form a ring R, called truncated polynomial ring, that is isomorphic to relations ring.

The NTRU uses truncated polynomial ring R together with modulo division into mutually prime numbers p and q to reduce the coefficients of the polynomials.

The algorithm also uses inverse polynomials in a truncated polynomial ring. It should be noted that not every polynomial has an inverse polynomial, but if an inverse polynomial exists, it is easy to find.

ISSN 2663 - 4023

To send a message from Alice to Bob you need public and private keys. Both Bob and Alice know the public key, only Bob knows the private key, that he uses to generate the public key. To do this, Bob chooses two "small" polynomials f_g from *R*. The "smallness" of the polynomials is implied in the sense that it is small with respect to an arbitrary polynomial modulo *q*: in an arbitrary polynomial the coefficients should be approximately evenly distributed modulo *q*, and in a small polynomial they are much less than *q*. The "smallness" of the polynomials is determined by numbers *df* and *dg* : Polynomial *f* has *df* coefficients equal to "1" and *df* - 1 coefficients equal to "-1", and the others - "0"; Polynomial *g* has *dg* coefficients equal to "1" and as many equal to "-1", the others - "0" The reason why polynomials are chosen in this way is that *f* may have a inverse polynomial, and *g* - definitely may not (*g* (1) = 0, and the zero element has no inverse).

Bob must keep these polynomials a secret. Bob then calculates inverse polynomials. Next, B calculates the inverse polynomials *fp* and *fq*, that is, such that:

$$f \ f_p \equiv 1 (\mathrm{mod}\ p)$$

$$f \ f_q \equiv 1 (\mathrm{mod}\ q)$$

If *f* does not have an inverse polynomial, then Bob chooses another polynomial *f*. The secret key is a pair, and the public key *h* h is calculated with a formula:

$$h = (pf_q \cdot g) \bmod q$$

For example, take *df*=4, and *dg*=3. Then as polynomials can be chosen

$$f = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10},$$

$$g = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}.$$

Next, for the polynomial f inverse polynomials modulo *p*=3 and *q*=32 are sought:

$$f_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9,$$

$$f_q = 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 +$$

$$+ 22X^7 + 20X^8 + 18X^9 + 30X^{10}.$$

The final step is the calculation of the public key *h* :

$$h = (pf_q \cdot g) \bmod 32 = 8 + 25X + 22X^2 + 20X^3 + 12X^4 +$$

$$+ 24X^5 + 15X^6 + 19X^7 + 12X^8 + 19X^9 + 16X^{10}.$$

Now that Alice has a public key, she can send an encrypted message to Bob. To do this, the message must be represented as a polynomial *m* with coefficients modulo *p*, selected from the range [- *p*/2, *p*/2]. That is, *m* is a "small" polynomial modulo *q*. Next, Alice needs to choose another "small" polynomial *r*, called "blinding", defined by the number *dr*. The polynomial *r* has *d*r coefficients equal to "1" and as many equal to "-1", the others - "0"

Using these polynomials, the encrypted message is obtained by the formula:

$$e = (r \cdot h + m) \bmod q .$$

In this case, anyone who knows (or can find) a blinding polynomial *r*, will be able to read the message *m*.

Next we describe the decryption procedure. Now that it has received the encrypted message *e*, Bob can decrypt it using its secret key. First he gets a new intermediate

ISSN 2663 - 4023

polynomial:

$$a = (f \cdot e) \bmod q = \left(f \cdot \left(r \cdot h + m\right)\right) \bmod q = \left(f \cdot \left(r \cdot p \cdot f_p \cdot g + m\right)\right) \bmod q = \\ = \left(pr \cdot g + f \cdot m\right) \bmod q.$$

After Bob has calculated the polynomial a modulo *q*, he must choose its coefficients from the range (-*q*/2, *q*/2] and then calculate the polynomial b obtained from the polynomial *a* by modul *p*:

$b = a \bmod q = \left(f \cdot m\right) \bmod q$, because $\left(pr \cdot g\right) \bmod p = 0$.

Now, using the second half of the secret key and obtained by the polynomial *b*, Bob can decrypt the message:

$c = \left(f_p \cdot b\right) \bmod p$,

$c \equiv f_p \cdot f \cdot m \equiv m \left(\bmod p\right)$

Therefore, the polynomial c thus obtained is indeed the original message *m*.

Stability of the NTRU cryptosystem. Let $\{b_1, b_2, ..., b_n\}$ - linearly independent system of vectors. Lattices *L* is called a set of integer linear combinations:

$$L(b_1, \ldots, b_2) = \{\sum x_i b_i : x_1, \ldots, x_n \in Z\}.$$

The frontal attack on the NTRU is based on lattices and the search for the shortest vector in the lattice. For secret key *f* (*x*) disclosure the attacker can construct a matrix (1).

And generate a lattice *L* from the rows of this matrix. Since Alice's public key is *h* (*x*) = *g* (*x*) * *f*-1 (*x*) this lattice will contain the vector *t* = (*a* * *f* (*x*), *g* (*x*)). Moreover, this vector is the shortest in the lattice *L*. Accordingly, discovering such a vector will lead to finding the secret key *f* (*x*). The task of finding the shortest lattice vector is considered to be computationally difficult. The time estimate for a lattice attack on an NTRU can be calculated using the formula *T* = 2 (0.4*N*-3, 5). For *N* = 251, this amounts to approximately 2^{100}.

The detailed analysis allows the authors to state the following. The fact is that with the advent of the quantum computer it will be possible to implement the Shore algorithm, that allows to solve the factorization task, and a discrete logarithm task at the same time. Naturally, in light of this, RSA, DSA and other similar algorithms become useless. But the situation is different with NTRU, because a quantum algorithm for solving the problem of the shortest lattice vector does not exist, which means that it is quite applicable in the "post-quantum" era.

$$\left(\begin{array}{cccc|cccc} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n-1} \\ 0 & \alpha & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right) \qquad (1)$$

Some words about stability. The fact is that although NTRU as well as RSA do not guarantee stability in case of proof of inequality $P \neq NP$ (this is explained by the fact that the problem is classified as NP even if there is at least one difficult-to-solve option or, more simply, in the worst case, while the other options can have an easy solution. Accordingly, no one can guarantee that even if it is proved that P ≠ NP, the attacker will not get an easier option and breaking will not be possible in polynomial time), some lattice-based problems have a proven stability relation of average and worst-case situations. This gives hope that in the future there may be a cryptosystem similar to the NTRU, that guarantees polynomial strength where the condition $P \neq NP$ is satisfied. These two differences make NTRU so different from their predecessors.

Analysis of the principles of construction and size of the length of the cryptographic key suggests that in ECC and RSA, public and private keys can be selected with approximately same length, while NTRU public key size differs from private with a much larger ratio [13,14,15]. The size of the NTRU cryptosystem's public key provides useful information about bandwidth usage if the cryptosystem is used in key exchange schemes.

Table 1. Gives the corresponding NTRU, ECC and RSA key sizes for equivalent security levels of 80 bits, 112 bits and 128 bits, etc. From Table 1 it can be seen that among the three methods, ECC is best used in terms of bandwidth, and NTRU bandwidth is more efficient against RSA to increase security. Although RSA is the most studied, tested and thoroughly researched cryptosystem, it is emphasized that ECC will gain significant trust over time, and even now, many security vendors include ECC modules in their own products [16,17,18,19].

It is easy to conclude that the NTRU has better performance. For the same key sizes, the NTRU can encrypt / decrypt more messages per second than the RSA, and key generation is faster in the NTRU.

The C code built into GMP is used when measuring key generation time, encryption and decryption of ECC. The curves used are NIST or/and SEK recommended elliptic curves with simple fields, and encoding and decoding measurements are taken in several coordinate systems such as affine, design, Jacobian, Chudnovsky, and modified functional determinant.

Table 1.

The sizes of public keys (in bits)

Security level (bits)	Public key sizes (bits)		
	NTRU	ECC	RSA
80	2008	160	1024
112	3033	224	2048
128	3501	256	3072
160	4383	320	4096
192	5193	384	7680
256	7690	521	15360

NTRU seems faster than ECC with all security levels considered. This is mainly due to the fact that the operations in the NTRU are relatively simple and there is no need for special conditions, as in ECC. For example, they do not even need to use long arithmetic when it is necessary for ECC. In addition, although time measurement may have been influenced by various factors such as runtime, compiler options, code optimization, and in the case of ECC - the advantages and disadvantages of using the library to work with long arithmetic, it is unlikely that ECC has higher performance. Next, the authors will present the direction of improvement based on reducing the encrypted text and reducing the time for

encryption and decryption procedures.

3. PROPOSED IMPROVEMENT NTRUENCRYPT PUBLIC KEY CRYPTOSYSTEM

The authors of the work identify the main areas of improving the efficiency of functioning of *ntruencrypt* public key cryptosystem. First, if you look closely at the formulas used for encryption and decryption, it is easy to see that the polynomial $f_p^{-1}(x)$ is successfully reduced and is not used anywhere else. So, you can not waste time on its calculation. Secondly, if we no longer need to calculate $f_p^{-1}(x)$, then we can choose the number q so that it is more convenient to perform calculations from this module. It is logical to use the power of two. For example, 64, 128, 256 , etc. Then, instead of a slow operation of taking the remainder of the division, you can use a fast bitwise "and" operation (or similar).Third, the description of the algorithm states that the modules p and q do not necessarily have to be simple, it is enough that they are mutually simple. This means that for complex p in the ring of numbers from this module p there will always be irreversible elements. Such disadvantage can affect the sarch for a polynomial. To avoid this problem, it is recommended to use prime numbers. And it is not necessary to spend time generating a prime number (methods of check on simplicity are too slow to use them without the reason), it is enough to take value p from set {3,5,7,11, ...}.

The developed way to improve the ntruencrypt algorithm is aimed at reducing the size of ciphertext. It is realized by proper selection of parameters p and q. Since the coefficients of the encrypted polynomial do not exceed q, it can be encoded in a data block of length $n \cdot \log_2(q)$. Similarly, the polynomial message can be represented as a data block $n \cdot \log_2(p)$. Thus ciphertext is in $\log_2(q)/\log_2(p) = \log_p(q)$. Hence, to reduce the difference between the length of ciphertext and plaintext, it is necessary to take p and q closer to each other. At the same time it is necessary to consider that at too close values the cryptosystem can be weakened.

Let us denote the relation of the size of the ciphertext to the size of the plaintext by the magnification coefficient k. Thus, to achieve the desired magnification, the parameters must be selected according to formula $q=p^k$.

As can be seen from table 2, the magnification coefficient k decreases significantly when overcoming the parameter p powers of two. For example, increasing p from 5 to 7 does not bring the desired effect, because $[\log_2 5] = [\log_2 7]$. It follows that the parameter p must be chosen slightly greater than the power of two, for example 5, 9, 17, or equal to the powers of two. Similarly, the parameter q should be chosen as close as possible to the power of two on the lower side. It is recommended to choose p- the power of a prime number, a little more than the power of two, and q-the power of two. The complex of means of protection offered by authors includes a set of applications, for example antivirus, the firewall and others. Each element of the complex plays an important role in ensuring security. Based on the topic of the work, the main object discussed in this article is an advanced software module for asymmetric encryption using the ntruencrypt algorithm, called ntruoptimized.

As a result of the proposed method, a software implementation of the procedure for the encryption and decryption of a information in Ruby 1.9.3 was obtained using the cryptolib library. The algorithms were tested in the Crypto ++ 5.6.0 software environment on a dual-core Intel Core 1.83 ghz processor running Windows 8 32 bit x86 (table 3).

Table 2.

The relation of the size of ciphertext to the size of plaintext.

Q	*P*	Plaintext length	Ciphertext length	Coefficient
256	3	N	8 • N	8
128	3	N	7 • N	7
128	4	2 • N	7 • N	3.5
128	5	2 • N	7 • N	3.5
128	7	2 • N	7 • N	3.5
128	9	3 • N	7 • N	2.33
128	17	4 • N	7 • N	1.75
64	17	4 • N	6 • N	1.5

Software that implements asymmetric encryption using the optimized ntruencrypt algorithm does not require a network for encryption. But as you know, the essence of public key encryption is that there are a pair of keys that complement each other: public and private. Each of the keys included in the pair is suitable for decrypting a message encrypted using another key from the same pair. Knowing the public key, it is almost impossible to calculate the private one. The public key can be published and widely distributed on communications networks. Therefore, the network is necessary for the most asymmetric encryption.

Table 3.

Time of key generation, encryption and decryption.

Cryptosystem	Crypto-graphic key size (bits)	Key generation (ms)	Encryption (ms)	Decryption (ms)
NTRU	251	0,076	0.002	0,008
Ntruoptimised	251	0,29	0,001	0,003
NTRU	491	0,288	0,006	0,031
Ntruoptimised	491	0,375	0,003	0,025
NTRU	587	0,412	0,008	0,044
Ntruoptimised	587	0,522	0,006	0,016

The developed software implements asymmetric encryption using the optimized ntruencrypt algorithm. Because ntruencrypt uses polynomials for encryption, the ntrupoly class has been created to perform all necessary operations with polynomials. Graphically, the dependence of classes for the developed program, which implements ntruoptimized is shown in Fig.1.To see the result in the software, must enter the command *test(N,p,q,df,dg)*, where *N* – size of the polynomial selected for the keys, *p* – mutually prime with *q*, and determines the interval to which all the coefficients of the polynomial must belong, *q* - mutually prime with *p*, *df,dg* – serve to define the polynomial.

4. CONCLUSION

Thus, in this article we give a full description the way of improvement the ntruencrypt algorithm method for the encryption and decryption of information. Proposed way of improvement, which allows the reducing the size of ciphertext which can minimizes the time for software encryption and decryption operations and increases cryptographic resistance due to the selection of parameters *p* and *q*.

ISSN 2663 - 4023

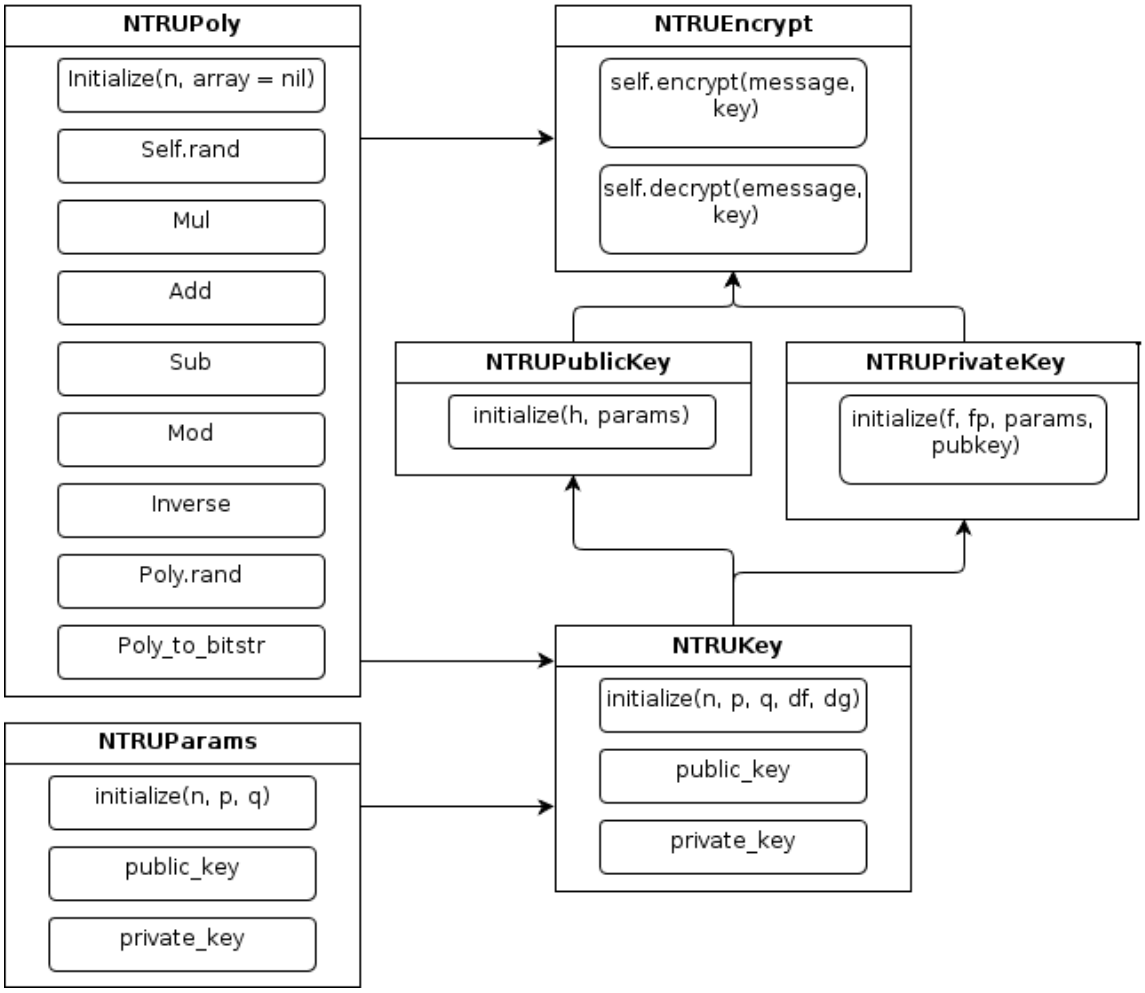

Fig.1. Dependence of classes of the developed program ntruoptimised

We can say that this proposed method is more simple compared to others and much more economical than computing resources. As a result of the modification, the size of ciphertext has decreased in the range from 1,3 times to 2,5 times depending parameters *p* and *q* and the time for encryption and decryption decreased in the range from 1,24 times to 2,75 times depending on the cryptographic key size. Thus, in this way, the proposed algorithm ntruoptimised for the encryption and decryption of an information with the ability to provide the function confidentiality and integrity in modern information networks. Our future research will focus on building more other improved encryption and decryption algorithms of information for provide cryptographic stability, software performance assessment and reliability of an algorithm as for cryptoanalysis.

REFERENCES

[1] W. Diffie and M.E. Hellman, Multiuser Cryptographic Techniques, Proceedings of AFIPS National Computer Conference, 1976, pp. 109-112.
[2] W. Diffie and M.E. Hellman, New Direc- tions in Cryptography, IEEE Transactions on Information Theory, v. IT-22, n. 6, Nov 1976, pp. 644
[3] R.C. Merkle, Secure Communication Over Insecure Channels, Communications of the ACM, v. 21, n. 4, 1978, pp. 294-299
[4] Schneier, B.: Applied Cryptography, 2nd edn. John Wiley & Sons, Inc., New Jersey, USA (2015).
[5] Rivest, R, Adleman, L, Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of

secure computation, Academic Press, pp 169–177 (1978)
[6] T. Elgamal, A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
[7] J. H. Silverman, Almost Inverses and Fast NTRU Key Creation, Tech. Rep. 14,NTRU Cryptosystems, Inc., March 1999. Version 1.
[8] H. Silverman, Communitive NTRU: Pseudo-code Implementation, Tech.Rep. 1, NTRU Cryptosystems, Inc., August 1997. Version 2.
[9] J. H. Silverman High-Speed Multiplication of Truncated Polynomials, Tech.Rep. 10, NTRU Cryptosystems, Inc., January 1999. Version 1
[10] N. Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, v. 48, n. 177,1987, pp. 203-209;
[11] V.S. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology CRYPTO '85 Proceedings, Springer-Verlag, 1986, pp.417-426
[12] Colleen Marie O'Rourke, Efcient NTRU Implementations,Master's thesis,ECE Department, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, April 2002
[13] Holstein and J. H. Silverman, Optimizations for NTRU, in Proceedings of Public Key Cryptography and Computational Number Theory, de Gruyter, Warsaw, September 2000.
[14] Kazmirchuk, S., Anna, I., Sergii, I.: Digital signature authentication scheme with message recovery based on the use of elliptic curves. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds.) ICCSEEA 2019. AISC, vol. 938, pp. 279–288. Springer, Cham (2020). Https://doi.org/10.1007/978-3-030-16621-2_26.
[15] Kazmirchuk, S.: New secure digital signature scheme: mathematical principles, speed and security analysis. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds.) ICCSEEA 2020. AISC, vol. 1247, pp. 327–337. Springer, Cham (2021). Https://doi.org/10.1007/978-3-030-55506-1_30
[16] Zhengbing Hu, Dychka, I., Onai, M., Zhykin. Y.: Blind Payment Protocol for Payment Channel Networks. International Journal of Computer Network and Information Security 6(11), 22-28 (2019).
[17] István, V.: Construction for Searchable Encryption with Strong Security Guarantees. International Journal of Computer Network and Information Security 5(11), 1-10 (2019).
[18] Goyal, R., Khurana M.: Cryptographic Security using Various Encryption and Decryption Method. International Journal of Mathematical Sciences and Computing 3(3), 1-11 (2018).
[19] Jayashree, A., Ashalatha, R.: Security and Privacy for Data Storage Service Scheme in Cloud Computing. International Journal of Information Engineering and Electronic Business 4, 7-12 (2017).

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023

Ільєнко Анна Вадимівна
Кандидат технічних наук, доцент, доцент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет,
факультет кібербезпеки комп'ютерної та програмної інженерії, Київ, Україна
ORCID: 0000-0001-8565-1117
Ilyenko.a.v@nau.edu.ua

Ільєнко Сергій Сергійович
Кандидат технічних наук, доцент, доцент кафедри автоматизації та енергоменеджменту
Національний авіаційний університет університет,
аерокосмічний факультет, Київ, Україна
ORCID: 0000-0002-0437-0995
Ilyenko.s.s@nau.edu.ua

Прокопенко Олена Володимирівна
Асистент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет,
факультет кібербезпеки комп'ютерної та програмної інженерії, Київ, Україна
ORCID: 0000-0001-9895-888X
Bortnik.olena.v@nau.edu.ua

ВДОСКОНАЛЕННЯ КРИПТОСИСТЕМИ NTRUENCRYPT: ПРОЕКТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

Анотація. На сьогодні криптографічні системи забезпечують безпечний зв'язок між користувачами. У цій роботі ми описуємо існуючі криптографічні системи, такі як: системи, засновані на складності факторизації великого цілого числа (RSA); системи, засновані на складності розв'язку дискретного логарифму в кінцевому полі Галуа (eigamal, DSA); системи, засновані на складності розв'язування дискретного логарифму в групі точок еліптичної кривої (ECC); системи на базі решітки (NTRU). Автори зосереджують свою увагу на алгоритмі шифрування та дешифрування ntruencrypt. Криптосистема з відкритим ключем ntruencrypt гарантує цілісність та конфіденційність інформації при передачі, зберіганні та обробці інформаційних повідомлень в сучасних комп'ютерних системах та мережах. Проведені дослідження криптосистем з відкритим ключем дали змогу визначити шлях удосконалення криптосистеми з відкритим ключем ntruencrypt. У цій роботі ми представляємо удосконалену криптосистему з відкритим ключем ntruencrypt, яка базується на правильному виборі параметрів p та q. Автори дійшли висновку, що, щоб зменшити різницю між довжиною зашифрованого та відкритого тексту, необхідно взяти p і q ближче один до одного. У той же час необхідно враховувати, що при занадто близьких значеннях p і q криптосистема може бути ослаблена. Основною відмінністю між запропонованими схемами було зменшення розміру зашифрованого тексту, що може мінімізувати час на операції шифрування та дешифрування. Як результат - програмна реалізація процедури шифрування та дешифрування удосконаленої криптосистеми з відкритим ключем ntruencrypt з використанням мови програмування Ruby 1.9.3 була отримана за допомогою бібліотеки cryptolib. Удосконалений алгоритм стане ідеальним інструментом для забезпечення конфіденційності інформації за допомогою «хмарних» обчислень, оскільки захист інформації від несанкціонованого доступу є однією з найактуальніших проблем. Далі автори планують ряд науково-технічних рішень для розробки та впровадження ефективних методів, інструментів для задоволення вимог, принципів та підходів до кібербезпеки та криптосистем для забезпечення цілісності та конфіденційності інформації в експериментальних комп'ютерних системах та мережах.

Ключові слова: Криптосистема з відкритим ключем, цілісність, конфіденційність, шифрування, зашифрований текст.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] W. Diffie and M.E. Hellman, Multiuser Cryptographic Techniques, Proceedings of AFIPS National Computer Conference, 1976, pp. 109-112.
[2] W. Diffie and M.E. Hellman, New Direc- tions in Cryptography, IEEE Transactions on Information Theory, v. IT-22, n. 6, Nov 1976, pp. 644
[3] R.C. Merkle, Secure Communication Over Insecure Channels, Communications of the ACM, v. 21, n. 4, 1978, pp. 294-299
[4] Schneier, B.: Applied Cryptography, 2nd edn. John Wiley & Sons, Inc., New Jersey, USA (2015).
[5] Rivest, R, Adleman, L, Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of secure computation, Academic Press, pp 169–177 (1978)
[6] T. Elgamal, A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
[7] J. H. Silverman, Almost Inverses and Fast NTRU Key Creation, Tech. Rep. 14,NTRU Cryptosystems, Inc., March 1999. Version 1.
[8] H. Silverman, Communitive NTRU: Pseudo-code Implementation, Tech.Rep. 1, NTRU Cryptosystems, Inc., August 1997. Version 2.
[9] J. H. Silverman High-Speed Multiplication of Truncated Polynomials, Tech.Rep. 10, NTRU Cryptosystems, Inc., January 1999. Version 1
[10] N. Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, v. 48, n. 177,1987, pp. 203-209;
[11] V.S. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology CRYPTO '85 Proceedings, Springer-Verlag, 1986, pp.417-426
[12] Colleen Marie O'Rourke, Efcient NTRU Implementations,Master's thesis,ECE Department, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, April 2002
[13] Holstein and J. H. Silverman, Optimizations for NTRU, in Proceedings of Public Key Cryptography and Computational Number Theory, de Gruyter, Warsaw, September 2000.
[14] Kazmirchuk, S., Anna, I., Sergii, I.: Digital signature authentication scheme with message recovery based on the use of elliptic curves. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds.) ICCSEEA 2019. AISC, vol. 938, pp. 279–288. Springer, Cham (2020). Https://doi.org/10.1007/978-3-030-16621-2_26.
[15] Kazmirchuk, S.: New secure digital signature scheme: mathematical principles, speed and security analysis. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds.) ICCSEEA 2020. AISC, vol. 1247, pp. 327–337. Springer, Cham (2021). Https://doi.org/10.1007/978-3-030-55506-1_30
[16] Zhengbing Hu, Dychka, I., Onai, M., Zhykin. Y.: Blind Payment Protocol for Payment Channel Networks. International Journal of Computer Network and Information Security 6(11), 22-28 (2019).
[17] István, V.: Construction for Searchable Encryption with Strong Security Guarantees. International Journal of Computer Network and Information Security 5(11), 1-10 (2019).
[18] Goyal, R., Khurana M.: Cryptographic Security using Various Encryption and Decryption Method. International Journal of Mathematical Sciences and Computing 3(3), 1-11 (2018).
[19] Jayashree, A., Ashalatha, R.: Security and Privacy for Data Storage Service Scheme in Cloud Computing. International Journal of Information Engineering and Electronic Business 4, 7-12 (2017).