ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT

UDC 004.942 https://doi.org/10.23947/1992-5980-2019-19-2-185-194

Comparative analysis of NTRUEncrypt modified post-quantum cryptographic system and standard RSA cryptosystem*

P. V. Razumov[1], I. A. Smirnov[2], I. A. Pilipenko[3], A. V. Selyova[4], L. V. Cherkesova[5]**
[1,2,3,4,5] Don State Technical University, Rostov-on-Don, Russian Federation

Сравнительный анализ модифицированной постквантовой криптографической системы NTRUENcrypt и общепринятой криптосистемы RSA***

П. В. Разумов[1], И. А. Смирнов[2], И. А. Пилипенко[3], А. В. Селёва[4], Л. В. Черкесова[5]**
[1,2,3,4,5]Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Introduction. The NTRUEncrypt cryptographic system, the calculation of the algorithmic complexity of the development of the NTRUEncrypt cryptosystem and its modifications are considered. The study objectives are to develop NTRUEncrypt, an efficient post-quantum cryptographic algorithm, which has high cryptographic resistance to quantum computer attacks, to work out a modification of the proposed algorithm, to analyze and experimentally validate its advantages.

Materials and Methods. A description of the NTRUEncrypt encryption system is proposed. The modification of the considered algorithm is studied; the block diagram of the implementation of the software based on it is presented. An example of the software operation and its characteristic is given. The reliability of the results is proved using the Mann-Whitney U test. During the experiment, the third-party software implementation of the RSA cryptosystem was used. A Stopwatch class element was introduced in the source code of all three programs of NTRUEncrypt, RSA, and NTRUEncrypt modifications. This class provides a set of methods and properties that can be used for the precise measurement of the execution time. Thus, it became possible to record the results of the time spent on all three basic stages: key creation, encryption and decryption of the message.

Research Results. The advantages of the developed cryptosystems in terms of the performance characteristics are proved. An experimental comparison of the implemented NTRUEncrypt algorithm and its modification is performed. All advantages of the latter are indicated.

Discussion and Conclusions. The advantage of using the NTRUEncrypt algorithm modification is experimentally validated. The new application is 25% faster to perform general work on key generation, encryption and decryption. In

Введение. Статья посвящена исследованию криптографической системы NTRUEncrypt, расчету алгоритмической сложности разработки криптосистемы NTRUEncrypt и ее модификации. Цели исследования: разработка эффективного постквантового криптографического алгоритма NTRUEncrypt, обладающего высокой криптостойкостью к атакам с квантового компьютера, а также разработка модификации предложенного алгоритма, анализ и экспериментальное доказательство его преимуществ.

Материалы и методы. Предложено описание системы шифрования NTRUEncrypt. Изучена модификация рассматриваемого алгоритма, представлена блок-схема реализации основанного на нем программного средства. Приведен пример работы программного средства и дана его характеристика. Достоверность результатов обоснована с помощью *U*-критерия Манна — Уитни. При проведении эксперимента использована сторонняя программная реализация криптографической системы RSA. В исходный код всех трех программ NTRUEncrypt, RSA, модификации NTRUEncrypt был внедрен элемент класса Stopwatch. Данный класс предоставляет набор методов и свойств, которые можно использовать для точного измерения времени, затраченного на выполнение. Таким образом, появилась возможность фиксировать результаты затраченного времени на всех трех основных этапах: создание ключей, шифрование и расшифрование сообщения.

Результаты исследования. Доказаны преимущества разработанных криптосистем по характеристикам производительности. Выполнено экспериментальное сравнение реализованного алгоритма NTRUEncrypt и его модификации. При этом обозначены все преимущества последней.

Обсуждение и заключения. Экспериментально доказано преимущество использования модификации алгоритма NTRUEncrypt. Новое приложение на 25 % быстрее выполняет общую работу по генерации ключей, шифрованию и расшифрованию. Помимо этого оптимизируется использование внутренней памяти за счет уменьшения веса ис-

*The research is done within the frame of the independent R&D.

**E-mail: therazumov@gmail.com, terran.doatk@mail.ru, ipilipenko@donstu.ru, tone4ka.selyova@yandex.ru, chia2002@inbox.ru

*** Работа выполнена в рамках инициативной НИР.

addition, the internal memory usage is optimized through reducing the weight of the source program file and the size of the secret key. When attempting to crack a ciphertext, cryptographic robustness and complexity of using quantum algorithms are shown.

ходного файла программы и размера секретного ключа. При попытке взлома шифротекста проявляется криптографическая стойкость и сложность использования квантовых алгоритмов.

Keywords: cryptographic system, post-quantum cryptographic algorithm, cryptographic strength, Mann-Whitney U test, encryption.

Ключевые слова: криптографическая система, постквантовый криптографический алгоритм, криптостойкость, *U*-критерий Манна — Уитни, шифрование.

For citation: P.V. Razumov, et al. Comparative analysis of NTRUEncrypt modified post-quantum cryptographic system and standard RSA cryptosystem. Vestnik of DSTU, 2019, vol. 19, no. 2, pp. 185−194. https://doi.org/10.23947/1992-5980-2019-19-2-185-194

Образец для цитирования: Сравнительный анализ модифицированной постквантовой криптографической системы NTRUENcrypt и общепринятой криптосистемы RSA / П. В. Разумов [и др.] // Вестник Дон. гос. техн. ун-та. — 2019. — Т. 19, № 2. — С.185−194. https://doi.org/10.23947/1992-5980-2019-19-2-185-194

Introduction. The paper [1] gave an impulse to the development of a new cryptographic system. It shows that quantum computers potentially threaten to hack into all widely used cryptographic algorithms.

The software presented in this paper is cryptosecure versus possible quantum attacks and surpasses its counterparts (for example, RSA cryptosystem) in the algorithm speed characteristics and in the quantity of spendable resources [2]. This explains the urgency of the research.

The object of the study is the NTRUEncrypt cryptosystem.

The subject of the research is the algorithmic complexity of the development of the NTRUEncrypt cryptosystem and its modifications.

The work objectives are to develop an efficient post-quantum cryptographic NTRUEncrypt algorithm, which has high cryptographic resistance to quantum computer attacks, to work out modifications of the proposed algorithm, to analyze and experimentally validate its advantages.

To achieve the objective, the following tasks were defined.

1. To investigate the algorithm of the NTRUEncrypt cryptosystem.
2. To develop an algorithm for modifying NTRUEncrypt.
3. To implement the NTRUEncrypt cryptosystem software and its modifications.
4. To analyze and compare two programs with each other and with their counterpart - RSA cryptosystem.

Materials and Methods. Consider the description of the encryption NTRUEncrypt system. The cryptographic system with the public NTRUEncrypt key uses operations over the $Z[X] / (X^N - 1)$ ring of polynomials of degree not exceeding $N - 1$ [3]:

$$a = a_0 + a_1 * X^1 + a_0 * X^2 + \cdots + a_{N-1} * X^{N-1},$$

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

The operations of addition and multiplication are performed as usual, except that X^N is replaced by 1, X^{N+1} is replaced by X^1, X^{N+2}, so on.

The cryptosystem is determined by a number of parameters, the key parameters are: *N, p* and *q*. To maintain the algorithm strength, it is necessary for *p* and *q* parameters to be coprime.

To provide high resistance of the algorithm to various attacks, it is recommended to use the following parameters (Fig. 1):

Indication	N	q	p	df	dg	dr	Guaranteed resistance
NTRU167:3	167	128	3	61	20	18	Moderate level of resistance
NTRU251:3	251	128	3	50	24	16	Standard level of resistance
NTRU503:3	503	256	3	216	72	55	Highest level of resistance
NTRU167:2	167	127	2	45	35	18	Moderate level of resistance
NTRU251:2	251	127	2	35	35	22	Standard level of resistance
NTRU503:2	503	253	2	155	100	65	Highest level of resistance

Fig.1. Recommended parameters

Research Results

Key generation. Bob wants to send a message to Alice. For this, he needs public and private keys. Therefore, he chooses randomly two small polynomials f and g from the ring of truncated polynomials R. The smallness of polynomials means that the small polynomial will be much less than q with respect to the arbitrary polynomial modulo q, in which the coefficients are uniformly distributed [4]. To determine the smallness of polynomials, the numbers *df* and *dg* are used which Bob chooses independently.

The polynomial f will have *df* coefficients equal to one, (*df* – 1) coefficients equal to minus one, and the rest coefficients equal to zero.

The polynomial g will have dg coefficients equal to one, as many coefficients equal to minus one, and the rest coefficients equal to zero.

Bob should keep the selected polynomials in secret since anyone he learns them will be able to decrypt the message.

Further, Bob calculates the inverse polynomials f_p and f_q modulo p and q, respectively, such that:

$$f \times f_p = 1(mod\ p) \text{ и } f \times f_q = 1(mod\ q).$$

If by chance these inverse polynomials do not exist, then Bob goes back and re-selects the polynomial f.

The secret key is the pair (f, f_p), and the public key h is calculated using the formula:

$$h = p \times f_q \times g\ (mod\ q).$$

Encryption. Alice wants to send a message to Bob using the public key h. To do this, Alice needs to present her message as a polynomial m with the coefficients modulo p selected from the range (–*p*/2, *p*/2]. Then, Alice needs to choose another small polynomial r which is called “blinding”, and calculate the ciphertext:

$$e\ =\ (r \times h + m)(mod\ q).$$

Decryption. Bob receives an encrypted message e from Alice and wants to decrypt it. First, using his secret key, Bob calculates:

$$a = f \times e\ (mod\ q).$$

Since Bob calculates the value a modulo q number, he should choose its coefficients from the range (–*q*/2, *q*/2] and then calculate:

$$b = a\ (mod\ p).$$

Finally, Bob, using the second part of the secret key, receives the original message from Alice:

$$c = f_p \times b\ (mod\ p).$$

Modification of the NTRUEncrypt algorithm. As can be seen from the description of the algorithm, the polynomial f shall comply with the following requirements:

- the polynomial f is invertible modulo p,
- the polynomial f is invertible modulo q,
- the polynomial f is a small polynomial.

In the algorithm itself, the invertibility modulo p and q was guaranteed as follows. If the polynomial f being not invertible in one of the moduli was generated, then it was discarded and the next one was generated — and so on as long as the required polynomial was found.

The proposed modification is to replace the polynomial f by a polynomial of the form:

$$f = 1 + pF, \tag{1}$$

where F is a small polynomial.

This approach has the following advantages.

1. From the expression (1), it is clear that the polynomial *f* is always invertible modulo *p*. This fact accelerates the key generation since it is not necessary to additionally calculate f_p.

2. Since $f^{-1} = 1$ *mod p*, then decoding does not require additional multiplication by f^{-1}, which speeds up the decryption process itself. In this case, the private key will not be the pair (f, f_p), but (*f*).

Key generation. As in the original algorithm, Bob first selects the encryption parameters *N, p, q* and the numbers *df, dg*. Then, he selects randomly two small polynomials *F* and *g* from the ring of truncated polynomials *R*.

He calculates the modified polynomial *f* using the formula (1).

Next, Bob calculates the inverse polynomial f_q modulo *q:*

$$f \times f_q = 1(mod\ q).$$

If by chance an inverse polynomial is not found, Bob goes back and re-selects the polynomial *f.*

The secret key is the polynomial *f*, and the public key *h* is calculated as follows:

$$h = p \times f_q \times g\ (mod\ q).$$

Encryption. Encryption remains unchanged; all is quite as in the original NTRUEncrypt algorithm.

Alice wants to send a message to Bob using the public key *h*. To do this, Alice needs to present her message as the polynomial *m* with coefficients modulo *p* selected from the range (–*p*/2, *p*/2]. Then, Alice needs to choose another small polynomial *r*, which is called “blinding”, and calculate the ciphertext:

$$e\ =\ (r \times h + m)(mod\ q).$$

Decryption. Bob receives the encrypted message *e* from Alice and wants to decrypt it. In the first place, using his secret key, Bob calculates:

$$a = f \times e\ (mod\ q).$$

Since Bob calculates the value *a* modulo *q* number, he should choose its coefficients from the range (–*q*/2, *q*/2], and then calculate:

$$b = a\ (mod\ p).$$

That is all, calculation is finished at that; we have received the original message from Alice: *b* = *m* [5].

Proof of the modified algorithm. To prove the algorithm, consider the decryption process itself.

Alice’s encrypted message looks like:

$$e\ =\ (r \times h + m)(mod\ q).$$

Bob uses his secret key — polynomial *f*:

$$a = f \times e\ (mod\ q)\ = (f\ \times\ (r\ \times\ h\ +\ m))(mod\ q)\ = (f\ \times\ (r\ \times\ pf_q \times g +\ + m))(mod\ q).$$

As a result:

$$a = (pr\ \times\ g\ +\ m\ \times\ f)(mod\ q).$$

Thereafter, Bob obtains the polynomial *b* through decreasing the coefficients of the polynomial *a* modulo *p*:

$$b = a(modp) = m \times f(modp) = (m\ +\ m\ \times\ p\ \times\ F)(mod\ p) = m\ (mod\ p).$$

Thus, we have checked and proved that the polynomial *b* is the original message *m* indeed.

Implementation of the algorithm. The programming language used is the object-oriented programming language *C#*, which belongs to the family of languages with a *C*-like syntax. Development framework was Microsoft Visual Studio 2015 Enterprise. The primary advantage of this software product is the application with a graphical interface that allows the user to make short work with the device and the operation scheme of this software product.

Fig. 2 shows a generalized block diagram of the software.

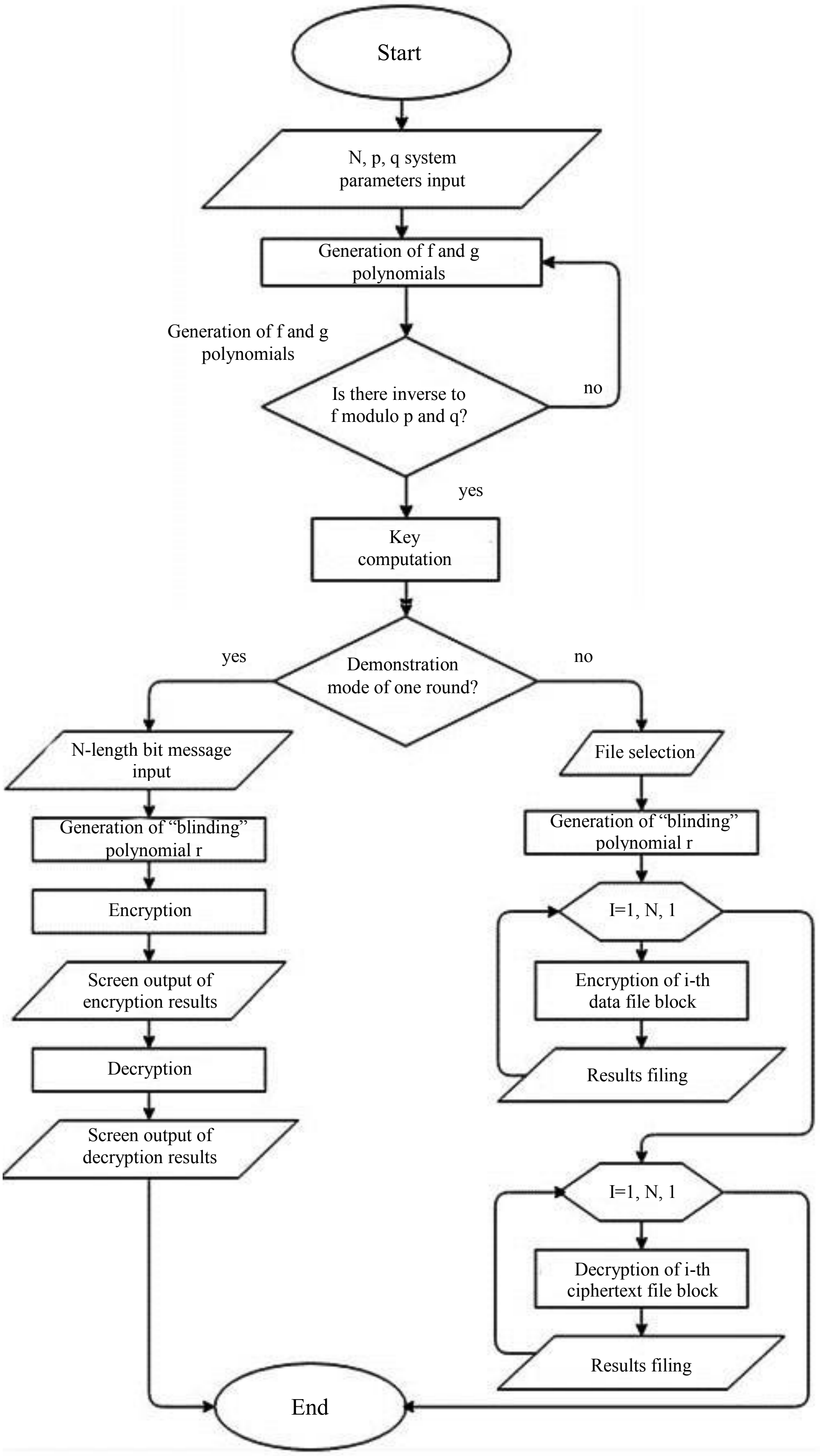

Start
N, p, q system parameters input
Generation of f and g polynomials
Generation of f and g polynomials
Is there inverse to f modulo p and q?
no
yes
Key computation
yes
Demonstration mode of one round?
no
N-length bit message input
Generation of "blinding" polynomial r
Encryption
Screen output of encryption results
Decryption
Screen output of decryption results
File selection
Generation of "blinding" polynomial r
I=1, N, 1
Encryption of i-th data file block
Results filing
I=1, N, 1
Decryption of i-th ciphertext file block
Results filing
End

Fig. 2. Flowchart of software implementation

Example of the software. NTRUEncrypt uses three constant parameters: *N, p, q*. The user enters them in the *System Settings* panel.

Next, there is the *Key Generation* panel. The parameters entered by the user in the previous step are input. Then, two polynomials *f* and *g* are randomly formed, so that:

- the number of coefficients equal to one and zero to be equal to the number that was previously determined in the program;

- the degree of polynomials to be corresponded to the input parameter N.

Then, the Euclidean theorem on computing the greatest common divisor for polynomials and its backward course is used to calculate two polynomials inverse to f modulo p and q, respectively. The public key h is computed.

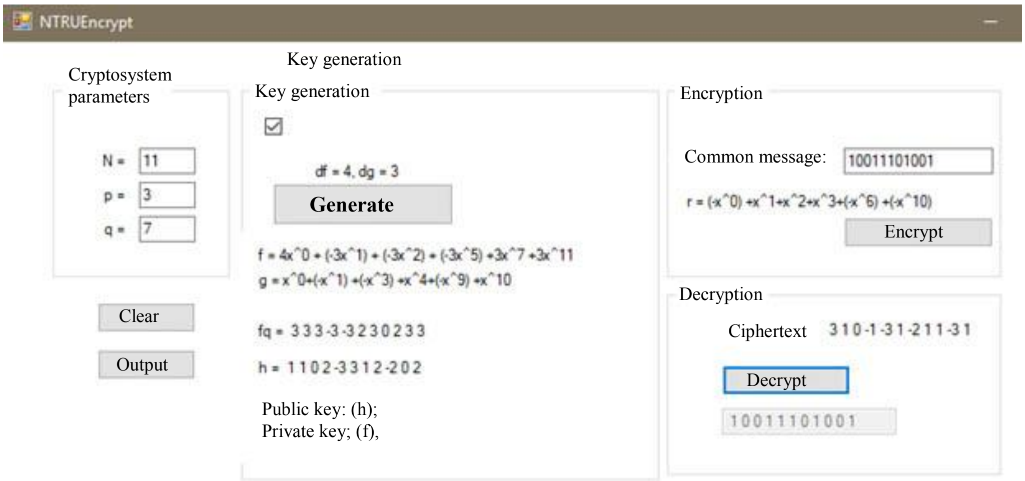
The user enters the binary message type m in the *Encryption* panel in the *Message* window. It should be mentioned that the original message should be divided into blocks of N bits, each of which will be processed separately and converted into a polynomial with the coefficients {–1, 0, 1} (in this program, instead of the value "–1", "2" is used).

For an encryption operation, the program needs to perform preliminary preparation — to generate one blinding polynomial. It is formed in the same way as f and g polynomials.

The resulting public key and the polynomial r make it possible to encrypt message m using the appropriate formula. Then, the result is checked on the membership to the ring of truncated polynomials of a degree not exceeding $N – 1$, and is displayed on the screen.

The next block implements the decryption mechanism. For this, actions are consistently performed considering the ring $Z[X]/(X^N – 1)$.

The operation result is shown in Fig. 3.



N = 11
p = 3
q = 7

df = 4, dg = 3

f = 4x^0 + (-3x^1) + (-3x^2) + (-3x^5) +3x^7 +3x^11
g = x^0+(-x^1) +(-x^3) +x^4+(-x^9) +x^10

fq = 3 3 3-3-3 2 3 0 2 3 3

h = 1 1 0 2-3 3 1 2-2 0 2

Public key: (h);
Private key; (f),

Common message: 10011101001

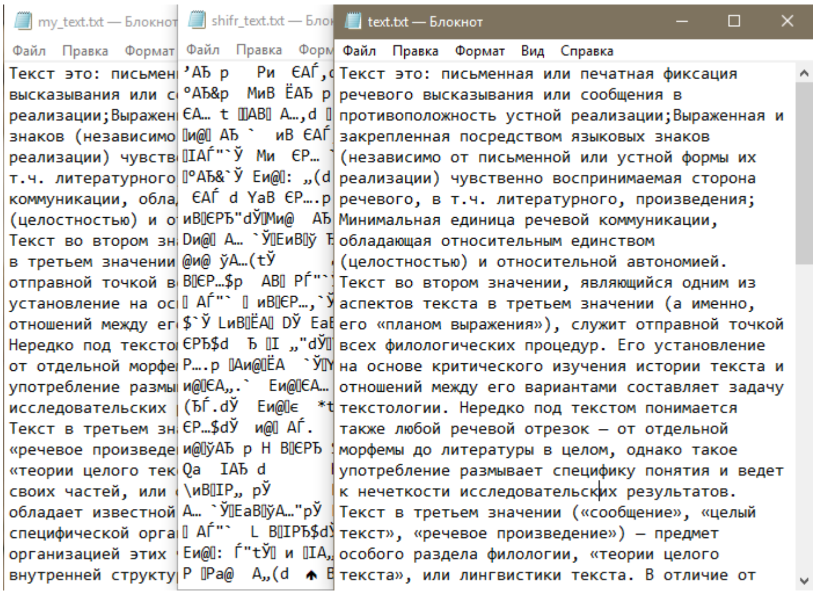
r = (-x^0) +x^1+x^2+x^3+(-x^6) +(-x^10)

Ciphertext 3 1 0-1-3 1-2 1 1-3 1

1 0 0 1 1 1 0 1 0 0 1

Fig. 3. Appearance of NTRUEncrypt program

Fig. 4 shows the software results in the form of source text, encrypted and decrypted.



```
Текст это: письмен
высказывания или с
реализации;Выражен
знаков (независимо
реализации) чувств
т.ч. литературного
коммуникации, обла
(целостностью) и о
Текст во втором зн
в третьем значении
отправной точкой в
установление на ос
отношений между ег
Нередко под тексто
от отдельной морфе
употребление размы
исследовательских
Текст в третьем зн
«речевое произведе
«теории целого тек
своих частей, или
обладает известной
специфической орга
организацией этих
внутренней структу
```

```
’АЂ р   Ри  ЄАЃ,
°АЂ&р  МиВ ЁАЂ р
ЄА… t ▯АВ▯ А…,d ▯
▯и@▯ АЂ `  иВ ЄАЃ
▯IАЃ"`Ў  Ми  ЄР… `
▯°АЂ&`Ў Еи@▯: „(d
 ЄАЃ d YaВ ЄР….р
иВ▯ЄРЂ"dЎ▯Ми@  АЂ
Dи@▯ А… `Ў▯ЕиВ▯ў Ђ
@и@ ўА…(tЎ
В▯ЄР…$р  АВ▯ РЃ"`
▯ АЃ"` ▯ иВ▯ЄР…,`Ў
$`Ў LиВ▯ЁА▯ DЎ Еа
ЄРЂ$d  Ђ ▯I „"dЎ▯
Р….р ▯Аи@▯ЁА  `Ў▯Y
и@▯ЄА„.`  Еи@▯ЄА…
(ЂЃ.dЎ  Еи@▯є  *t
ЄР…$dЎ  и@▯ АЃ.
и@▯ўАЂ р Н В▯ЄРЂ
Qа  IАЂ d
\иВ▯IР„ рЎ
А… `Ў▯ЕаВ▯ўА…"рЎ
▯ АЃ"`  L В▯IРЂ$dЎ
Еи@▯: Ѓ"tЎ▯ и ▯IА„
Р ▯Ра@  А„(d  ↑ В
```

```
Текст это: письменная или печатная фиксация
речевого высказывания или сообщения в
противоположность устной реализации;Выраженная и
закрепленная посредством языковых знаков
(независимо от письменной или устной формы их
реализации) чувственно воспринимаемая сторона
речевого, в т.ч. литературного, произведения;
Минимальная единица речевой коммуникации,
обладающая относительным единством
(целостностью) и относительной автономией.
Текст во втором значении, являющийся одним из
аспектов текста в третьем значении (а именно,
его «планом выражения»), служит отправной точкой
всех филологических процедур. Его установление
на основе критического изучения истории текста и
отношений между его вариантами составляет задачу
текстологии. Нередко под текстом понимается
также любой речевой отрезок – от отдельной
морфемы до литературы в целом, однако такое
употребление размывает специфику понятия и ведет
к нечеткости исследовательских результатов.
Текст в третьем значении («сообщение», «целый
текст», «речевое произведение») – предмет
особого раздела филологии, «теории целого
текста», или лингвистики текста. В отличие от
```

Fig. 4. Software result

Experiments. The experimental research objective is to show and prove the advantages in the performance characteristics of the developed cryptosystems in comparison with the analogue. (In this paper, an RSA public key cryptosystem was chosen as an alternative [6].) In addition, it was necessary to compare the implemented NTRUEncrypt algorithm and its modification in order to note the advantages of the latter and provide experimental evidence.

To verify the results, the statistical Mann – Whitney *U*-test was used, which is intended for comparing two independent samples by the level of any feature measured quantitatively. The criterion enables to determine the degree of difference between samples and is more powerful than the Rosenbaum criterion.

For the experiment, a third-party software implementation of the RSA cryptographic system was used. An element of the Stopwatch class was introduced in the source code of all three programs – NTRUEncrypt, RSA, and NTRUEncrypt modifications. This class provides a set of methods and properties that can be used to accurately measure the time spent on the execution. Thus, it became possible to record the results of the elapsed time at all three main stages: key creation, encryption and decryption of the message. The experiment results are shown in Table 1.

Table 1

Experiment results

№	NTRUEncrypt operate time, s	NTRUEncrypt modification operate time, s	RSA operate time, s
Key generation			
1	0.0239676	0.0190014	0.0964144
2	0.0149743	0.0109994	0.1023557
3	0.0129915	0.0099936	0.1372658
4	0.0170503	0.0099772	0.0491137
5	0.0139885	0.0099773	0.0869555
6	0.0139880	0.0099782	0.0587503
7	0.0129761	0.0099936	0.0986608
8	0.0139931	0.0109775	0.0707417
9	0.0139918	0.0099773	0.0595015
10	0.0169748	0.0099777	0.0517874
Encryption			
1	0.0069961	0.0069770	0.0453772
2	0.0049961	0.0059796	0.0598658
3	0.0059954	0.0049965	0.0265461
4	0.0049966	0.0069966	0.0402407
5	0.0069765	0.0060854	0.0783415
6	0.0049961	0.0069989	0.0097463
7	0.0059795	0.0059954	0.0247876
8	0.0069961	0.0059954	0.0272320
9	0.0089956	0.0049798	0.0494907
10	0.0059954	0.0049744	0.0100916
Decryption			
1	0.0009843	0.0009843	1.0624476
2	0.0010212	0.0010026	0.3692022
3	0.0010193	0.0009974	0.7651423
4	0.0009989	0.0009975	0.9922103
5	0.0030012	0.0009988	0.3056423
6	0.0010017	0.0009975	0.3757183
7	0.0009989	0.0009988	1.2844272
8	0.0010021	0.0009989	0.5902735
9	0.0010026	0.0009984	0.9932694
10	0.0009844	0.0009970	0.8740392

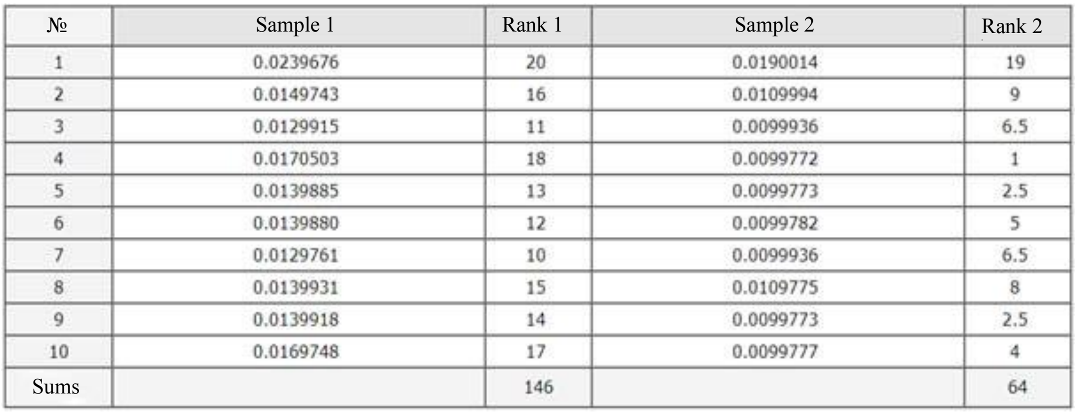
The Mann – Whitney *U*-test is applicable to the results given in Table 1. Compare the key generation rate of the NTRUEncrypt algorithm and its modifications.

Following the algorithm of the Mann-Whitney *U*-test, we criterion write the computation step by step.

1. Create a common ranked list of both samples assigning a lower value to a lower rank.

2. Divide the total ranked list into two consisting of the elements of the first and second samples.

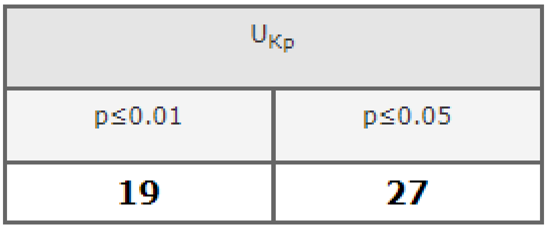
3. Calculate the sum of ranks for the first and second samples separately, as shown in Fig. 5.



№	Sample 1	Rank 1	Sample 2	Rank 2
1	0.0239676	20	0.0190014	19
2	0.0149743	16	0.0109994	9
3	0.0129915	11	0.0099936	6.5
4	0.0170503	18	0.0099772	1
5	0.0139885	13	0.0099773	2.5
6	0.0139880	12	0.0099782	5
7	0.0129761	10	0.0099936	6.5
8	0.0139931	15	0.0109775	8
9	0.0139918	14	0.0099773	2.5
10	0.0169748	17	0.0099777	4
Sums		146		64

Fig. 5. The second step of calculating the Mann-Whitney *U*-test

4. Calculate the value of the Mann-Whitney *U*-test: *U* = 9. The critical value of the criterion for the data *n*1 and *n*2 should be determined from to the table of statistical significance level (Fig. 6).



U_{Kp}	
p≤0.01	p≤0.05
19	**27**

Fig. 6. The forth step of calculating Mann-Whitney *U*-test

Since the values of *n*1 and *n*2 are the same for all experiments, this table will be used in each calculation.

Hence it follows that the obtained empirical value *U* = 9 is in the zone of significance (Fig. 7). Consequently, there is a significant difference between the speed of the NTRUEncrypt program and its modification. If this fact is expressed percentagewise, it turns out that the NTRUEncrypt modification is 28% faster than the program itself.

Axis of significance:

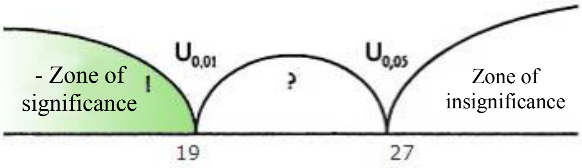


Fig. 7. Axis of significance

Similarly, in shorthand, we present the results of the Mann-Whitney *U*-test and the percentage superiority for the remaining cases.

Key generation

1. NTRUEncrypt modification is 28% faster than NTRUEncrypt. The obtained empirical value *U* = 9 is in the significance zone.

2. NTRUEncrypt program is faster than RSA by 80%. The obtained empirical value *U* = 0 is in the area of significance.

3. NTRUEncrypt modification is faster than RSA by 86%. The obtained empirical value *U* = 0 is in the area of significance.

Message encryption

1. The obtained empirical value *U* = 47 is in the zone of insignificance.

2. NTRUEncrypt program is f80%aster than RSA. The obtained empirical value $U = 0$ is in the area of significance.

3. NTRUEncrypt modification is faster than RSA by 84%. The obtained empirical value $U = 0$ is in the area of significance.

Decryption

1. NTRUEncrypt modification is 17% faster than NTRUEncrypt. The obtained empirical value $U = 24$ is in the zone of uncertainty.

2. NTRUEncrypt program is faster than RSA by 99%. The obtained empirical value $U = 0$ is in the area of significance.

3. NTRUEncrypt is99% faster than RSA. The obtained empirical value $U = 0$ is in the area of significance.

Discussion and Conclusions. In the framework of this research, the following is developed:

- software that implements the NTRUEncrypt cryptosystem operation;
- software that implements this cryptosystem modification.

One of the NTRUEncrypt advantages over its counterpart - the RSA cryptosystem, a higher speed of operation can be specified. Performing encryption and decryption operations requires $O(n^2)$ operations, unlike $O(n^3)$ in the same RSA. As for the experimental data, the NTRUEncrypt program wins significantly on the speed of the algorithm compared to RSA. In addition, there is a slight increase in durability with the same key length. The disadvantage of the system is the necessity for using the recommended parameters.

Regarding the NTRUEncrypt resistance, after the creation of quantum computers, the problems of fast factorization and discrete logarithmation will be solved [7]. In this case, RSA, DSA, and similar algorithms will become useless. The relevance of NTRUEncrypt will remain: it will be fully applicable in the "post-quantum" era since there is no algorithm that solves the problem of the shortest lattice vector.

The advantage of using the NTRUEncrypt algorithm modification is experimentally proven. The developed application performs the general work on key generation, encryption and decryption 25% faster. In addition, it optimizes the use of internal memory through reducing the weight of the original program file and the size of the private key. When attempting to crack a ciphertext, cryptographic robustness and the complexity of using quantum algorithms are manifested.

References

1. Shor, P. Algorithms for Quantum Computation: Discrete Log and Factoring. Murray Hill: AT&T Bell Labs, 1994, pp. 124–134.

2. Shakleina, T.A. «Mozgovye tsentry» i ikh rol' v formirovanii vneshney politiki SShA. ["Brain centers" and their role in making US foreign policy.] Vvedenie v prikladnoy analiz mezhdunarodnykh situatsiy. [Introduction to the applied analysis of international situations.] Moscow: Aspect press, 2014, p. 112 (in Russian).

3. Alferov, A.P., et al. Osnovy kriptografii. [Cryptography basics.] Moscow: Gelios ARV, 2002, pp. 209–220 (in Russian).

4. Laponina, O.R. Kriptograficheskie osnovy bezopasnosti. [Cryptographic Security Basics.] Moscow: National Open University INTUIT, 2016, pp. 118 (in Russian).

6. Ishmukhametov, Sh.T. Metody faktorizatsii natural'nykh chisel. [Methods of factorization of natural values.] Kazan: Kazan University Publ. House, 2011, pp. 74–82 (in Russian).

5. Bakhtiari, M., Maarof, M.A. Serious Security Weakness in RSA Cryptosystem. International Journal of Computer Science and Information Security, 2012, no. 3, pp. 175–178.

7. Vasilenko, O.N. Teoretiko-chislovye algoritmy v kriptografii. / [Number theoretic cryptoalgorithms.] Moscow: MTsNMO, 2003, pp. 73–74 (in Russian).

Submitted 23.10.2018
Scheduled in the issue 14.03.2019

Authors:

Razumov, Pavel V.,
student of the Cybersecurity of IT Systems Department, Don State Technical University
(1, Gagarin sq., Rostov-on-Don, 344000, RF),
ORCID: https://orcid.org/0000-0003-2454-3600
therazumov@gmail.com

Smirnov, Ivan A.,
student of the Cybersecurity of IT Systems Department, Don State Technical University
(1, Gagarin sq., Rostov-on-Don, 344000, RF),
ORCID: https://orcid.org/0000-0001-6533-4368
terran.doatk@mail.ru

Pilipenko, Irina A.,
postgraduate student of the Cybersecurity of IT Systems Department, Don State Technical University
(1, Gagarin sq., Rostov-on-Don, 344000, RF),
ORCID: https://orcid.org/0000-0003-3236-6069
ipilipenko@donstu.ru

Selyova, Antonina V.,
student of the Cybersecurity of IT Systems Department, Don State Technical University
(1, Gagarin sq., Rostov-on-Don, 344000, RF),
ORCID: https://orcid.org/0000-0003-0990-7429
tone4ka.selyova@yandex.ru

Cherkesova, Larisa V.,
professor of the Cybersecurity of IT Systems Department, Don State Technical University
(1, Gagarin sq., Rostov-on-Don, 344000, RF), Dr.Sci. (Eng.), professor,
ORCID: https://orcid.org/0000-0002-9392-3140
chia2002@inbox.ru