Post-Quantum Cryptosystem NTRUEnCrypt and Its Advantage over Pre – Quantum Cryptosystem RSA

Larissa Cherckesova[1,*], *Olga* Safaryan[1], *Pavel* Razumov[1], *Veronica* Kravchenko[1], *Sergey* Morozov[2], *Alexey* Popov[2]

[1]Don state technical University Rostov–on–Don, 1, Gagarin square, Russia
[2]Don state technical University Shakhty, Rostov region, 147, Shevchenko str., Russia

Abstract. Cryptography is inextricably linked to the transfer of data, and in addition to ensuring user authorization; it is designed to guarantee the integrity of the transmitted information and its confidentiality. The cryptographic system NTRUEncrypt is able to provide the necessary level of security at an extremely low cost, while possessing high speeds and low memory requirements. This is the main reason for the attractiveness of the NTRUEncrypt cryptosystem and its widespread use at present. The goal of this work is to develop a software application that implements the NTRUEncrypt cryptosystem in a modern high level C # programming language.

1 Introduction

In today's world, it is hard to imagine the protection of confidential data without cryptography. This industry is actively developing and is receiving great support from states and large private companies. You can meet elements of cryptography almost everywhere - from encrypting messages in instant messengers and secure protocols to digital signatures of authoring documents [1].

In this regard, attackers are improving and developing attacks on cryptographic algorithms such as RSA, El –Gamal, DES, and others. Attacks based on quantum algorithms have been particularly effective [2].

A proof of the effectiveness of quantum algorithms before classical ones was the test of the algorithm for factoring Shore integer numbers, conducted in 2001 on a 7–qubit computer of IBM. If on standard computers the factorization of the 129 –bit number requires more than eight months of continuous operation of the system of 1600 workstations connected via the Internet, then Shor's quantum algorithm, using parallel computing, speeds up this process millions of times.

As for public-key cryptographic systems, using the algorithm of Peter Shor, it becomes possible to break them.

* Corresponding author: chia2002@inbox.ru

For example, the RSA cryptosystem is based on the computational complexity of the problem of factoring large numbers, and uses the public key M, which is the product of two large primes [3].

To crack a cryptosystem, it suffices to find the factors of the number M. Using the potential capabilities of a quantum computer, P. Shor's algorithm is able to solve the factorization problem not just for polynomial time, but even for the length of time it takes to perform an integer multiplication operation.

Thus, with the introduction of quantum technologies into operation, many cryptographic systems become useless, including the RSA cryptosystem.

It is for this reason that the NTRUEncrypt cryptosystem was created - more resistant to attacks from quantum computers due to the difficulties of finding the shortest lattice vector. As of mid-2010's (2015), not a single quantum algorithm is known that can cope with this cryptosystem better than conventional pre–quantum algorithms.

2 Description of RSA Algorithm

RSA cryptosystem is used in a wide variety of products, on various platforms and in many industries. It is used by Microsoft, Apple, Sun and Novell operating systems. In hardware, the RSA algorithm is used in secure phones, on Ethernet network cards, on smart cards, and is widely used in Zaxus cryptographic equipment (Racal). In addition, the algorithm is part of all major protocols for secure Internet communications, including S/MIME, SSL and S/WAN, and is also used in many institutions, for example, in government services, in most corporations, in government laboratories and universities. The technologies using the RSA algorithm were licensed by more than 700 companies in 2000.

The encryption algorithm is presented below:

1. A recipient takes two random numbers p_1 and p_2, from which forms the public key $n = p_1 * p_2$.
2. Next, the recipient calculates the value of the Euler function $F(n) = (p_1 - 1)(p_2 - 1)$.
3. The recipient calculates a simple integer e, 1 < e < F (n), which is coprime with the value of the function F(n).
4. The recipient creates secret exponent d = (k * F(n) +1)/e that will be in the private key. In addition, sends to the Sender the values of the public key n and e.
5. The sender, having public key values, encrypts the message and sends it to the Recipient.
6. The recipient decrypts the message $c^d = m^1 mod\ n$.

3 Description of the Post–Quantum NTRUEncrypt Algorithm

Lattice cryptography is an approach to building asymmetric encryption algorithms using lattice theory problems. The advantage of post–quantum cryptography is the use of lattice cryptography for solving problems.

The NTRUEncrypt public key cryptosystem uses operations on the Z[X] / (X^N–1) ring of polynomials of degree not exceeding N – 1:

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

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} is replaced by X^2, etc.

The NTRUEncrypt cryptosystem is determined by a number of parameters, the main ones being N, p and q. To preserve the cryptographic strength of the algorithm, it is necessary that the cryptosystem parameters p and q be mutually prime numbers.

Key generation. Let Bob want to send a message to Alice, and for this he needs public and private keys. Therefore, he randomly chooses two "small" polynomials f and g from the ring R. Bob must keep the selected polynomials in secret, since anyone he knows will be able to decipher the message.

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

$$f * f_p = 1(mod\ p) \text{ and } f * f_q = 1(mod\ q). \quad (2)$$

If by chance these inverse polynomials do not exist, then Bob goes back a step and reselects the polynomial f.

The secret key is a pair $(f,\ f_p)$, and the public key h is calculated by the formula:

$$h = p * f_q * g\ (mod\ q). \quad (3)$$

Encryption. Let Alice want to send a message to Bob with the public key h. To do this, Alice needs to present her message as a 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 ciphertext according to expression:

$$e = r * h + m\ (modp). \quad (4)$$

Decryption. Let Bob receive an encrypted message e from Alice and want to decrypt it. First of all, using his secret key, Bob calculates:

$$a = f * e\ (mod\ q). \quad (5)$$

Since Bob calculates the value of a modulo the number q, he must choose his coefficients from the range (–q/2, q/2] and then calculates:

$$b = a\ (mod\ p), \quad (6)$$

i.e., bring all coefficients of polynomial a modulo p.

Finally, Bob, using the second part of the secret key, can get the original message from Alice using the transform:

$$c = f_p * b\ (mod\ p). \quad (7)$$

Advantages of the NTRUEncrypt cryptosystem. In April 2011, the American Accredited Standards Committee X9 approved the use of the NTRU's fastest asymmetric encryption algorithm (NTRUEncrypt). The NTRU algorithm was developed as early as the mid–1990s.

Unlike the RSA cryptosystem, it was not widely used, because from the very beginning it was necessary to increase the cryptographic strength and performance of this encryption algorithm. To date, all the flaws have been fixed, and in practice, NTRUEncrypt is considered much faster than RSA. This fact is confirmed by RSA Labs themselves, as well as independent researchers.

One of these comparative studies conducted cryptologists from the Catholic University of Leuven (Belgium). They found that when testing with maximum security settings, the NTRUEncrypt asymmetric encryption algorithm is four orders of magnitude faster than RSA and three orders of magnitude faster than ECC. The graph (Figure 1) clearly shows how the

NTRUEncrypt cryptosystem exceeds, by the number of operations per second, the majority of the existing pre – quantum algorithms.

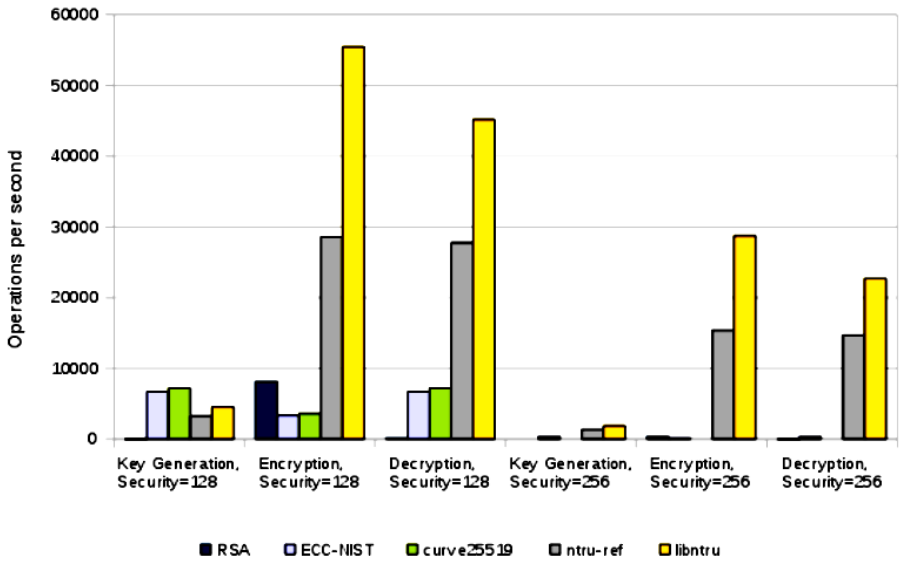

Fig. 1. Comparison of NTRU performance with other algorithms.

In addition, the NTRUEncrypt cryptosystem is post-quantum, and has higher cryptographic resistance to various attacks. In addition, as mentioned above, the cryptographic strength of the algorithm is ensured by the absence of an algorithm for finding the shortest lattice vector.

Disadvantages of the NTRUEncrypt cryptosystem. At this moment, there is one major drawback – the need to use only the recommended parameters. It is the same requirement that caused general discontent during the transition to elliptic curves and contributed to all sorts of suspicions about the presence of backdoors.

Table 1. Recommended Parameters NTRUEncrypt Cryptosystem.

	N	p	q	df	dg	dr
NTRU167:3	167	3	128	61	20	18
NTRU251:3	251	3	128	50	24	16
NTRU503:3	503	3	256	21	72	55
NTRU167:2	167	2	127	45	35	18
NTRU251:2	251	2	127	35	35	22
NTRU503:2	503	2	253	155	100	65

4 Attacks and Defense Against Attacks on NTRUEncrypt

Brute force. When carrying out the attack “Brutus Force” or “Hacking Brute Force”, the main task of the enemy is to pick up Alice’s secret key, i.e. polynomial $f(x)$. The adversary knows that the polynomial $f(x)$ of length N has d_f unit coefficients and $(d_f - 1)$ coefficients – 1.

Selection of such a polynomial will require checking

$$\binom{N}{d_f}\binom{N-d_f}{d_f-1} \tag{8}$$

options. For N = 251 and df = 50, this expression is 3 * 10100.

Comparing this assessment with the assessment of the complexity of solving the problem of finding the shortest lattice vector, we can conclude that, as in the case of RSA, brute – force key is not the most successful attack against the NTRU.

Attack "Meeting in the middle". Andrew Odlyzko offered the option of an attack meeting in the middle, which for the successful opening of the NTRU secret key takes $\binom{N/2}{d/2}/\sqrt{r}$ time and exactly the same amount of hard disk space (r - integer is not greater than N). As a matter of fact, attacks of this type are called a meeting in the middle because they allow you to spend the time required for memory calculations necessary for storing temporary data.

The attack proposed by Odlyzko is as follows. The definition of the public key $h = f_q * g \ mod \ q$ implies the equality g$= h * f \ mod \ q$. In this case, the attacker presents f as the concatenation of two polynomials of length N/2.

$$f(x) = f_1||f_1, g = h * (f_1||f_1) mod \ q. \tag{9}$$

We know that the polynomial g consists of coefficients {1, 0, –1} for the case p = 3 or coefficients {1, 0} for the case p = 2, i.e. in other words

$$f1 * h = \{1, 0\} – f2 * h. \tag{10}$$

Thus, if the cryptosystem key space of the NTRU is of a size $\binom{N}{d}$, then the search for a key by the method of meeting in the middle requires going through all $\binom{N/2}{d/2}$ the options.

In order to guarantee a 2^x level of resilience, it is necessary to select the cryptosystem parameters of the NTRU with a key space of 2^{2x}.

5 Protection against attack with selected ciphertext

In order to protect the NTRU from such an attack, it is recommended to use the NTRU in conjunction with the FORST supplement scheme. When encrypting using the NTRU – FORST method, Bob, as in the usual NTRU scheme, calculates the plaintext polynomial $m(x)$.

Complementing the polynomial with a random set of k bits of R, Bob computes

$$r(x) = H(m(x)||R), \tag{11}$$

where $H(x)$ is a cryptographically strong hash function.

Further, to obtain a ciphertext as in the usual NTRU scheme, Bob forms the polynomial

$$c(x) = r(x) * h(x) + m(x) mod \ q. \tag{12}$$

After receiving the ciphertext, Alice recovers the message $m\ (x)$, and calculates $H(m\ (x)||R)$.

Then Alice calculates:

$$H(m\ (x)||R) * h(x) + m(x) mod \ q \tag{13}$$

and compares the obtained value with $c(x)$.

$$\text{If } H(m\ (x)||R) * h(x) + m(x) mod \ q = c(x), \tag{14}$$

then Alice accepts the message, otherwise she rejects it.

6 Conclusion

Post–quantum cryptosystem NTRUENCrypt has a huge advantage over pre–quantum RSA cryptosystem, in speed, reliability and durability.

As for the NTRUEncrypt algorithm for cryptographic resistance, it is important to note the fact that after the creation of quantum computers, the problems of fast factorization and discrete logarithm will be solved.

Naturally, in the light of this fact, RSA, DSA and other public–key cryptosystems that are widely used at present are becoming useless.

But in the case of the NTRUEncrypt asymmetric encryption algorithm, the situation is different.

After all, the algorithm that solves the problem of the shortest lattice vector still does not exist (although searches are actively conducted from the first half of the 1990s), which means that the NTRUEncrypt cryptosystem is quite applicable in the "post–quantum" era.

References

1. Shor P 1994 *35th Annual Symposium on Computer Science (FOCS)* (Santa Fe, NM, USA) p 124 doi: 101109 / SFCS1994365700
2. http://wwwnydailynewscom/ news/ world / kasparov–deep–blues–losingchess–champ–rooke– article–1762264
3. Bakhtiari M, Maarof M A 2012 *IJCSI* **9(1).3** 175 ISSN 1694–0814; 1694–0784
4. Ishmukhametov Sh T 2011 *Methods of factorization of natural numbers: Tutorial* (Kazan: Kazan University) p 190
5. Vasilenko O N 2013 *Theoretic–numerical algorithms in cryptography* (Moscow: ICNMO) p 323 ISBN 5–94057–103–4
6. http: // poivstsputru / en / Math / Geometry / Convex_ Geometry / Hadamards_Inequality
7. Ajtai M 1997 *Electronic Colloquium for Computational Complexity* **TR97** 0047
8. Electronic resource: https://x9org/ (In Russian)
9. Cassels J 1965 *Introduction to the geometry of numbers* (M : Mir)