МЕТОДЫ И СРЕДСТВА АССИМЕТРИЧЕСКИХ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

UDK 004 056 55

FEATURES OF PARAMETERS CALCULATION FOR NTRU ALGORITHM

IVAN GORBENKO, OLENA KACHKO, KOSTYANTYN POGREBNYAK

The paper considers the practical aspects of parameters generation for the NTRU algorithm (ANSI X9.98). The obtained results will allow to generate new sets of parameters to meet the growing requirements for cryptographic security and the availability of new attacks on the lattice.

Keywords: NTRU parameters, cryptographic complexity, error decryption, combined attack.

INTRODUCTION

The NTRU algorithm is used for encryption and digital signatures, and has much better speed characteristics than the rest of the algorithms of this class [3, 9]. In spite of problems of using this algorithm for digital signature, its application to asymmetric encryption is based on standard [1] (the Standard). The aim of this work is the analysis of NTRU encryption algorithm parameters proposed in [1]. It is this choice of parameters that ensures the reliability of the algorithm in terms of repayment calculation operations of a private key, encryption and decryption and the required cryptographic security.

The fullest justification for the choice of parameters is considered in [2]. It is the parameters, which are proposed in this paper, are used in [1]. We have considered the formation of some parameters [2] for the purpose of their inspection and provision of the generation of other parameters which are not inferior to the proposed ones.

The parameters which are used in the NTRU encryption algorithm are as follows:

N is the polynomial degree that determines its the maximum exponent which ensures multiplicative group maximum order [5];

q is a "big" module, a positive integer number for which one can easily calculate the module value ($q = 2048$ for the Standard);

p is a "small" module, usually a small positive integer number or a polynomial with small coefficients of small degree, p and q are relatively prime ($p = 3$ for the Standard).

The designations d_f, d_g are used to set the amount of 1 and –1 in polynomials. These values are used to generate a private key; they actually determine the power of a set of private keys, $d_g = \left\lfloor \frac{N}{3} \right\rfloor$, $d_f < \frac{N}{3}$ for the Standard;

The designation d_r is used to set the amount of 1 and –1 in the blinding polynomial. It is applied for encrypting ($d_r = d_f$ for the Standard).

The remaining parameters depending on N, d_f and security level are not considered in the paper.

Thus, it is sufficient to choose the following parameters: N, d_f .

The reference [2] proposes the following criteria for choosing parameters:

– ensuring of the required security level;

– ensuring of minimal computational complexity;

– minimum memory requirement for storing key data and encrypted text.

According to the criteria listed three classes of parameters are defined.

1. Using minimal storage for keys and the encrypted block (N*11 bits) – size metric is achieved due to the minimum order of the polynomial for ensuring a given security level. In [1, 2] this class includes polynomials ((401, 113) (449.134) (677, 157) (1087, 120)). The first digit of the pair specifies the value of N, and the second one gives the value of d_f .

2. By reducing the number of non-zero elements for private keys one decreases the time required to perform a multiplication of polynomials (computational complexity is equal to $N * d_f$) – performance metric. In [1, 2] this class includes polynomials of (659, 38), (761, 42), (1087, 63), (1499, 79)) orders.

3. The combination of size and performance metrics makes it possible to obtain parameters that at small dimensions of additional memory enable to get a significant speed boost. In [1, 2] this class includes polynomials of ((541, 49), (613, 55), (887, 81), (1171, 106) orders.

The paper deals with the generation parameters that are determined by all the metrics.

CHOOSING PARAMETER *N*

This parameter considerably affects the resistance of the cryptographic algorithm. Since the very beginning of using this parameter it has been recommended to choose N as a prime number, which increases the probability of the existence of an inverse element [6] for the private key. Furthermore, [7] shows the availability of additional attacks if N is not a prime number. The probability of having an inverse element and effectiveness of attacks, based on the factorization of parameter q in module $X^N - 1$, depends on the group order n of number N. The degree of polynomial N is chosen if the appropriate group order of $n \geq \frac{N-1}{2}$.

The value of the group order for N (401, 449, 761, 887, 1087) is equal to $\frac{N-1}{2}$. The value of the group order for N (541, 613, 659, 677, 1171, 1499) is equal to N–1. Thus, the condition for the group order is performed for all values of N. 169 primes are in the range [401, 1499], of which 108 prime numbers satisfy the demand for the group order. Thus, in addition to the Standard preset one can choose other primes.

CHOICE OF PARAMETERS IN VIEW OF DECODING ERRORS

In order to determine the conditions of decoding error appearance it is enough to consider the decryption algorithm. The ambiguity appears when the coefficients of the polynomial exceed $q/2$. If one foresees the normal low of distribution of the coefficients values, the probability of exceeding a given value of c for at least one polynomial coefficient is determined according to the formula (1) with due account of $d_r = d_f$ [2]:

$$p_{err} = N * erfc\left(\frac{c}{\sigma\sqrt{2}}\right), \quad (1)$$

where $c = \frac{q-2}{2p}, \sigma = \sqrt{\frac{8d_f}{3}}$.

To provide a given cryptographic security level, this probability shouldn't exceed 2^{-S}, where S, respectively, equals {112, 128, 192, 256}.

The maximum values d_f, which provide the required decoding error absence probability, are shown in Table 1. The values of d_f, that are used in the Standard, are given in parentheses.

Table 1

Maximum values of d_f to ensure correct decoding

S/N	401	449	677	1087	541	613
112	133 (113)				180 (49)	
128		149 (134)				204 (55)
192			161 (157)			
256				122 (120)		
S/N	887	1171	659	761	1087	1499
112			219 (38)			
128				236 (42)		
192	161 (81)				161 (63)	
256		120 (106)				120 (79)

Solving the problem of finding private keys by public ones depends on the method of forming the private key. In the Standard the private key is formed according to the formula: $f = pF + 1$. In this case, the problem of finding the private key refers to Closest Vector Problem (CVP) [9]. The most effective solution to the problem is the hybrid attack [2].

To perform the hybrid attack the whole lattice is divided into 2 parts.

The first part includes y_2 first rows of the lattice ($y_2 = [N, 2N]$). The reduction operation is performed for it.

For the second part of the lattice the number of rows is equal to $2N - y_2$. The attack Meet-In-The-Middle (MITM) is performed for it [8]. The value of c – numbers 1 (–1) remaining in the second part of the lattice, is determined. It is assumed that the numbers 1 (–1) are identical. This is what provides the worst conditions for the attack.

Recovery is performed for the lattice, the number of rows is determined by values: $[y_1, y_2]$, where

$$y_1 = [0, N-1], y_2 = [N, 2N-1],$$

$$y_1 = \frac{2N - y_2(1+\alpha)}{1-\alpha}.$$

Obviously, the quality of recovery depends on the number of rows for which the reduction is performed, with increasing their number, the quality of recovery increases. To increase their number requires increasing the value of $\alpha = (0,1)$.

According to [2], to assess the time required to recover the specified number of lattice rows uses the equation:

$$w = \frac{2m(y_2 - N)}{(1-\alpha)^2} + 3\ln\frac{2(y_2 - N)}{(1-\alpha)} + c, \quad (2)$$

where $m = 0.2$, $c = -50$ constants determined experimentally; y_2 determines the number of the last lattice row, for which the reduction is realized; α is the factor that determines the number of rows that are takes into account for recovery; w determines the time needed for such an attack. Actually, the time is equal to 2^w.

The value of α can be found for a given cryptographic security level, solving equation (2) with respect to α. The maximum value of α can be found using a formula that determines the value y_1 by solving the imparity: $\frac{2N - y_2(1+\alpha)}{1-\alpha} \geq 0$.

So, $\alpha_{\max} = \frac{2N - y_2}{y_2}$. Due to the fact that the attack on the lattice is performed in parallel with a MITM attack the value of α may be less than it is necessary to obtain the current security level. In this case, the time required for the MITM attack, should not be less than it is necessary to ensure the security level. The total time required for a hybrid attack, is determined by the maximum time. Fig. 1 shows a graph of the dependence of the value α on y_2 ($\alpha_{\max}$).

Besides, a graph of values α is specified under which the required security level is achieved (labeled AlphaMin) to attack on the lattice. As shown in Fig. 1, the relative value range α (Delta graph) increases, as the value of y_2 increases. The graphs for different values of N are of similar character.

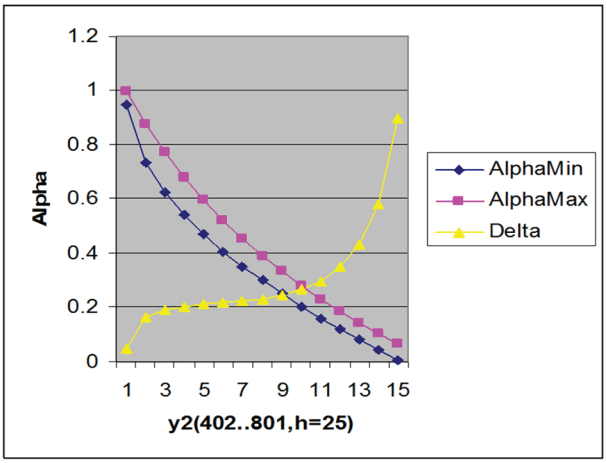

Figure 1. Graf of the dependencies of the α on y_2 ($N = 401$)

Figure 2 shows a graph of w-α, relationship obtained at the following values (N = 401, y_2 =693, α = [0.0668–0.1573]).

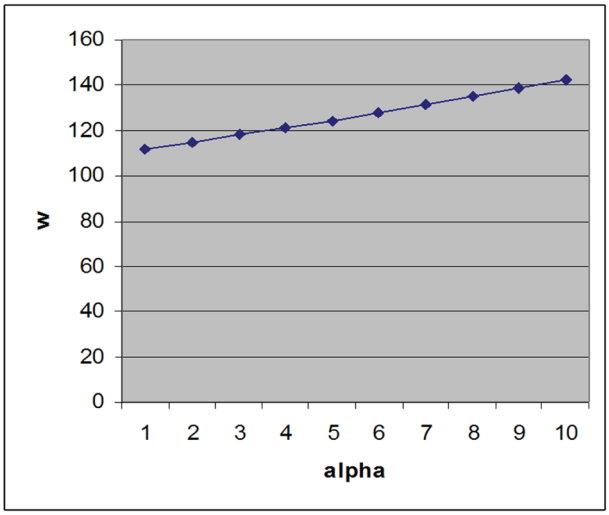

Fig. 2. Graph of w (α)

The initial value α is selected to get $w = 112$. The graph shows that the increase α leads practically to the linear increase w. Changing the value of α allows to balance lattice reduction time and recovery time of the rest of the private key.

As it can be seen from (2), the maximum value of α corresponds to the minimum y_2. With increasing y_2 it is necessary to reduce α to ensure the necessary security level.

Table 2 shows the calculated values of α for y_2 compared to the anticipated ones according to [2].

The values are determined only for those prime numbers that comply with the security level. As can be seen from Table 2, calculated and anticipated values practically coincide. The optimum values of parameters y_2,α will be determined forth. Time is calculated for this which is required to perform a MITM attack, realized for $2N - y_2$ rows of the matrix.

To calculate the parameters with respect to a MITM attack, one should consider the following:

– the numbers 1(–1) in the remaining rows ($c = 2N - y_2$) must be correct, i.e. with regard to their total (d_f). The author [2] designates the probability of this fact as p_{Split};

– the result obtained by means of the MITM attack must be consistent with the result that has been obtained with the help of the previous attack after finding the nearest lattice vector, i.e. the total number of 1 and –1 must be correct. The probability of this fact is denoted as p_s [2];

Table 2

Minimum values of α for specified security level

S	N	y_2	Anticipated values α	Obtained values α
112	401	693	0.095	0.0949443
	541	800	0.149	0.148151
	659	902	0.175	0.175155
128	449	770	0.100	0.0993571
	613	905	0.142	0.141385
	761	1026	0.183	0.182416
192	677	1129	0.096	0.0960767
	887	1294	0.143	0.142555
	1087	1464	0.175	0.174973
256	1087	1630	0.127	0.126437
	1171	1693	0.144	0.143584
	1499	1984	0.174	0.174653

The total time required to perform the attack is defined by the formula [2]:

$$w_2 = \frac{t\Pi}{p_{Split}}, \qquad (3)$$

where: t is the time required to perform one iteration; Π is the number of iterations.

Calculation of individual components of the formula (3) will be further examined.

p_{Split} CALCULATION

The designation $\begin{pmatrix} n \\ r_1 \quad r_2 \end{pmatrix}$, which means the number of variations that can be placed in n positions of r_1 data of one type (i.e. +1) and r_2 data of other type, for example (-1) will be used in the subsequent formulas [2]. All definitions below assume that $r_1 = r_2$, that is, the designation $\begin{pmatrix} n \\ r \quad r \end{pmatrix}$ will be actually used. The corresponding number of variations is calculated using the formula $C_n^r * C_{n-r}^r = \frac{n!}{(r!)^2 (n-2r)!}$.

p_{Split} is determined by the formula:

$$p_{Split} = \begin{pmatrix} y_2 - N \\ d_f - c \quad d_f - c \end{pmatrix} * \begin{pmatrix} 2N - y_2 \\ c \quad c \end{pmatrix} * \begin{pmatrix} N \\ d_f \quad d_f \end{pmatrix}^{-1}. \qquad (4)$$

The expression between the first brackets determines the number of options that can accommodate $d_f - c$ (1) and (–1) in the first $y_2 - N$ rows. The expression between the second brackets determines the number of options that can accommodate c (1) and

(–1) in the $2N - y_2$ rows. The expression between the third brackets determines the total number of options that can accommodate all (1) and (–1) (d_f) in N rows.

With the possibility of further use of the key rotation the formula to calculate p_{Split} take the form: $p_{Split} = \left(1 - p_{Split}\right)^N$.

Let us specify the limits at which the formula (4) makes sense considering restrictions on y_2, c: $y_2 \in [N, 2N-1]$, $c \in [0, d_f]$:

$$c_{\min} = \begin{cases} d_f - 0.5(y_2 - N) & 0.5(y_2 - N) < d_f \\ 0 & 0.5(y_2 - N) < d_f \end{cases};$$

$$c_{\max} = \begin{cases} 0.5(2N - y_2) & 0.5(2N - y_2) < \ d_f \\ d_f & 0.5(2N - y_2) \geq d_f \end{cases}.$$

The values of all components of (4) must be integer.

Table 3 shows set values of N, d_f, y_2, c and corresponding value of p_{Split} calculated by the formula (4) for the security levels 112, 128, 192 and 256 respectively. Besides, Table 3 shows the anticipated values according to [2]. The values obtained in the calculation and anticipated ones practically coincide.

The influence of the value c on the p_{Split} is considered using an example of calculations for $S = 112$, $N = 541$, $d_f = 49$, $y_2 = 800$. In this case, $c_{\min} = 0, c_{\max} = 49$. The calculation results have

Table 3

Calculated and anticipated values of p_{Split}

Sec	N	y_2	c	$\log_2 p_{Split}$ (anticipated)	$\log_2 p_{Split}$ (calculated)
112	401	693	27	–0.6	–0.58093
	541	800	15	–13.1	–13.0804
	659	902	13	–17.7	–17.6503
128	449	770	35	–0.3	–0.299504
	613	905	17	–14.9	–14.9203
	761	1026	15	–20.9	–20.8762
192	677	1129	45	–2.0	–2.03799
	887	1294	27	–21.9	–21.8938
	1087	1464	23	–31.9	–31.9323
256	1087	1630	39	–24.9	–24.9227
	1171	1693	37	–28.7	–28.7098
	1499	1984	29	–47.8	–47.8223

shown that at the beginning the probability increases monotonically, has a maximum in the set interval and then decreases monotonically.

p_s CALCULATION

In fact, the probability p_s determines that the norm for the calculation result obtained by the reduction (key G and part of key F without c (1) and (–1)) does not exceed the anticipated norm $2d_g + 2(d_f - c)$. At that it is assumed that the values distribution law for key G and key F is normal, in this case the error function can be used for the calculation.

To calculate this probability according to [2], one can use the formula:

$$p_s = \left(1 - \frac{2}{3q}\right)^{y_1} * \Pi_{i=0}^{y_2 - y_1} \left(1 - \overline{f_{D,\sigma}}\right), \quad (5)$$

where

$$y_1 = \frac{2N - y_2(1+\alpha)}{1-\alpha};$$

$$D = q^{\frac{\alpha(y_2 - y_1) + i(1-\alpha)}{y_2 - y_1}};$$

$$\overline{f_{D,\sigma}} = erfc\left(\frac{D}{\sigma\sqrt{2}}\right) - \frac{\sigma\sqrt{2}}{D\sqrt{\pi}}\left(e^{\frac{D^2}{2\sigma^2}} - 1\right);$$

$$\sigma = \sqrt{\frac{2d_g + 2(N-c)}{y_2}}.$$

The value of σ determines the part of non-zero elements among y_2 elements.

From (5) we get:

$$\log_2 p_s = y_1 \log_2\left(1 - \frac{2}{3q}\right) + \sum_{i=0}^{y_2 - y_1} \log_2\left(1 - \overline{f_{D,\sigma}}\right). \quad (6)$$

The results of the calculation according to (6) compared with the values obtained in [2] are presented in Table 4.

Table 4

Probability of selected key correctness according to norm

N	y_2	d_f	C	$\log_2 p_s$ (anticipated)	$\log_2 p_s$ (calculated)
401	693	113	27	–45.4	–45.4676
541	800	49	15	–26.9	–26.8954
659	902	38	13	–21.9	–22.0056
449	770	134	35	–49	–48.9491
613	905	55	17	–31.5	–31.5344
761	1026	42	15	–23.1	–23.0858
677	1129	157	45	–67.4	–67.6896
887	1294	81	27	–43.9	–43.851
1087	1464	63	23	–34.2	–34.3427
1087	1630	120	39	–64.1	–64.2834
1171	1693	106	37	–56	–55.1001
1499	1984	79	29	–44.4	–44.6091

As the results of the calculation have shown, the results obtained are close to these obtained in [2].

This paper presents the results of the generation of the main components that are necessary to determine the parameter d_f for the selected value N. The obtained results agree with those given in [2]. This will allow to generate and explore the parameters for the NTRU algorithm.

References

[1] American National Standard for Financial Services ANSI X9.98 – 2010. Lattice-Based Polynomial Public Key Establishment Algorithm for the Financial Services Industry.

[2] *P. Hirschhorn, J. Hoffstein, N. Howgrave-Graham3, W. Whyte.* Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches, https://www.securityinnovation.com/uploads/Crypto/params.pdf / (visited 22.12.2015).

[3] *Горбенко И.Д., Качко Е.Г., Балагура Д.С.* Применение криптосистем NTRU в инфраструктуре от-

крытых ключей. Міжнародна наукова конференція «Питання оптимізації обчислень (ПОО-XL)», Україна, Крим, Велика Ялта, смт. Кацивелі, 30.09.2013–04.10.2013.

[4] *Качко Е.Г., Балагура Д.С., Горбенко Ю.И.* Обоснование и исследование практической реализации улучшенного алгоритма цифровой подписи NTRUSIGN/ Прикладная радиоэлектроника, т.11, № 2, 2012 г. , с. 195-200.

[5] NTRU Cryptosystems Technical Report. http://www.securityinnovation.com/uploads/Crypto/NTRUTech014.pdf / (visited 22.12.2015).

[6] NTRU Cryptosystems Technical Report. Report #009, Version 1. Invertibility in Truncated Polynomial Rings https://www.securityinnovation.com/uploads/Crypto/NTRUTech009.pdf / (visited 22.12.2015).

[7] *Craig Gentry*. Key Recovery and Message Attacks on NTRU-Composite http://www.iacr.org/archive/eurocrypt2001/20450181.pdf / (visited 22.12.2015).

[8] *Nick Howgrave-Graham*. A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU. https://www.securityinnovation.com/uploads/Crypto/hybrid.pdf / (visited 22.12.2015).

[9] *Jeff Hoffstein, Nick Howgrave-Graham, Jill Pipher, William Whyte*. Practical lattice-based cryptography: NTRUEncrypt and NTRUSign. https://www.securityinnovation.com/uploads/Crypto/lll25.pdf/(visited 22.12.2015).

Manuscript received October, 30, 2015

Gorbenko Ivan, professor, doctor of technical sciences, State Prize Laureate (science and technology), chief designer of the private corporation «Institute of Information Technologies». Research interests: cryptology and cyber security.

Kachko Olena, Ph.D., assistant professor of the Department of Software Engineering of KhNURE. Research interests: cryptography, cryptanalysis, parallel computing.

Pogrebnyak Kostyantyn, Ph.D., assistant professor of the Department of Information Technology Security of KhNURE. Research interests: application of methods of algebraic geometry in cryptology, asymmetric cryptanalysis.

УДК 004 056 55

Особливості обчислення параметрів для алгоритму NTRU / І.Д. Горбенко, О.Г. Качко, К.А. Погребняк // Прикладна радіоелектроніка: наук.-техн. журнал. – 2015. – Том 14. – № 4. – С. 272–276.

Розглянуто практичні аспекти генерації параметрів для алгоритму NTRU (ANSI X9.98). Отримані результати дозволять генерувати нові набори параметрів з урахуванням зростаючих вимог до криптографічної складності та наявності нових атак на решітку.

Ключові слова: параметри NTRU, криптографічна складність, помилка розшифрування, комбінована атака.

Табл.: 4. Іл.: 2. Бібліогр.: 9 найм.

УДК 004 056 55

Особенности вычисления параметров для алгоритма NTRU / И.Д. Горбенко, Е.Г. Качко, К.А. Погребняк // Прикладная радиоэлектроника: научн.-техн. журнал. – 2015. – Том 14. – № 4. – С. 272–276.

Рассмотрены практические аспекты генерации параметров для алгоритма NTRU (ANSI X9.98). Полученные результаты позволят генерировать новые наборы параметров с учетом растущих требований к криптографической сложности и наличия новых атак на решетку.

Ключевые слова: параметры NTRU, криптографическая сложность, ошибка дешифрования, комбинированная атака.

Табл.: 4. Ил.: 2. Библиогр.: 9 назв.