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 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 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 \( d_f, d_s \) 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, \( d_s = \left\lfloor \frac{N}{3} \right\rfloor, \ d_s < \frac{N}{3} \) for the Standard;

The designation \( d_f \) is used to set the amount of \( 1 \) and \( -1 \) in the blinding polynomial. It is applied for encrypting \( (d = d_f \) 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 \( (N^3 \text{ bits}) \) – 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 \( N \cdot d_f \) – performance metric. In [1, 2] this class includes polynomials \( ((541, 49), (613, 55), (887, 81), (1171, 63), (1499, 79)) \) 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 \( ((541, 49), (613, 55), (887, 81), (1171, 106)) \) 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 observes 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]:

\[
p_{cr} = N^{*} \text{erfc}\left(\frac{c}{\sigma \sqrt{2}}\right),
\]

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

To provide a given cryptographic security level, this probability shouldn’t exceed \( 2^{-5} \), 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>
<thead>
<tr>
<th>( S/N )</th>
<th>401</th>
<th>449</th>
<th>677</th>
<th>1087</th>
<th>541</th>
<th>613</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>133</td>
<td></td>
<td></td>
<td>180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>149</td>
<td></td>
<td></td>
<td>204</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>161</td>
<td></td>
<td></td>
<td>122</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>887</td>
<td>1171</td>
<td>659</td>
<td>761</td>
<td>1087</td>
<td>1499</td>
</tr>
</tbody>
</table>

Table 1

Maximum values of \( S/N \) to ensure correct decoding

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: \( \mu = pF + 1 \). 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 \( y_1 \), first rows of the lattice \((y_1 = [N, 2N])\). 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 \((-1)\) 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

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

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:

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

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; \( \alpha \) is the factor that determines the number of rows that are taken into account for recovery; \( w \) determines the time needed for such an attack. Actually, the time is equal to \( 2^w \).

The value of \( \alpha \) can be found for a given cryptographic security level, solving equation (2) with respect to \( \alpha \). The maximum value of \( \alpha \) 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 \( \alpha \) 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 \( \alpha \) on \( y_2 \) (\( \alpha_{max} \)).

Besides, a graph of values \( \alpha \) 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 \( \alpha \) (Delta graph) increases, as the value of \( y_2 \) increases. The graphs for different values of \( N \) are of similar character.
МЕТОДЫ И СРЕДСТВА АССИМЕТРИЧЕСКИХ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

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

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

The initial value $\alpha$ is selected to get $w = 112$. The graph shows that the increase $\alpha$ leads practically to the linear increase $w$. Changing the value of $\alpha$ 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 $\alpha$ corresponds to the minimum $y_2$. With increasing $y_2$ it is necessary to reduce $\alpha$ to ensure the necessary security level.

Table 2 shows the calculated values of $\alpha$ 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 $y_2, \alpha$ 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 $p_2$ [2];

Table 2

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

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

$$w_2 = \frac{t\Pi}{P_{Split}},$$

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

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

$P_{Split}$ CALCULATION

The designation $\left\{ \begin{array}{c} n \\ r_1 \ r_2 \end{array} \right\}$, 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 $\left\{ \begin{array}{c} n \\ r \ r \end{array} \right\}$ will be actually used.

The corresponding number of variations is calculated using the formula $C_n^r C_r^r = \frac{n!}{(r!)^2(n-2r)!}.$

$P_{Split}$ is determined by the formula:

$$P_{Split} = \left( \begin{array}{cc} y_2 - N & d_f - c \\ d_f - c & d_f - c \end{array} \right)^{2N - y_2} \left( \begin{array}{cc} c & c \\ c & c \end{array} \right)^{2N - y_2}.$$

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
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 \( p_{\text{Split}} \) take the form:

\[
p_{\text{Split}} = (1 - p_{\text{Split}})^N.
\]

Let us specify the limits at which the formula (4) makes sense considering restrictions on \( y_2, c \):

\[
e_{\min} = \begin{cases} 
    d_f - 0.5(y_2 - N) & 0.5(y_2 - N) < d_f, \\
    0 & 0.5(y_2 - N) < d_f,
  \end{cases}
\]

\[
e_{\max} = \begin{cases} 
    0.5(2N - y_2) & 0.5(2N - y_2) < d_f, \\
    d_f & 0.5(2N - y_2) \geq d_f.
  \end{cases}
\]

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

Table 3 shows sets values of \( N, d_f, y_2, c \) and corresponding value of \( p_{\text{Split}} \) 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 \( p_{\text{Split}} \) is considered using an example of calculations for \( S’ = 112, N = 541, d_f = 49, y_2 = 800. \) In this case, \( e_{\min} = 0, e_{\max} = 49. \) The calculation results have shown that at the beginning the probability increases monotonically, has a maximum in the set interval and then decreases monotonically.

\( p_i \) \text{CALCULATION}

In fact, the probability \( p_i \) 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 \( 2d_f + 2(d_f - c). \) 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:

\[
p_i = \left(1 - \frac{2}{3q}\right)^{y_1} \prod_{i=0}^{y_1} \left(1 - f_{D,\sigma}\right), \tag{5}
\]

where

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

\[
D = q \frac{y_2}{y_2};
\]

\[
\frac{\sigma}{\sqrt{D}} = \text{erfc}\left(\frac{D}{\sigma\sqrt{2}}\right);
\]

\[
\sigma = \sqrt{2d_f + 2(N - c)}.
\]

The value of \( \sigma \) determines the part of non-zero elements among \( y_2 \) elements.

From (5) we get:

\[
\log_2 p_i = y_1 \log_2 \left(1 - \frac{2}{3q}\right) + \sum_{i=0}^{y_1} \log_2 \left(1 - f_{D,\sigma}\right). \tag{6}
\]

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

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

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 exploit the parameters for the NTRU algorithm.

References


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


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: cryptography 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 cryptography, asymmetric cryptanalysis.

УДК 004 056 55


Розглянуто практичні аспекти генерації параметрів для алгоритму NTRU (ANSI X 9.98). Отримані результати дозволяють 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


Рассмотрены практические аспекты генерации параметров для алгоритма NTRU (ANSI X9.98). Полученные результаты позволяют 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.