Int. J. Communications, Network and System Sciences, 2011, 4, 475-481
doi:10.4236/ijcns.2011.47058 Published Online July 2011 (http://www.SciRP.org/journal/ijcns)

Double-Moduli Gaussian Encryption/Decryption with Primary Residues and Secret Controls

Boris S. Verkhovsky
Computer Science Department, *New Jersey Institute of Technology*, *University Heights*, *Newark*, *USA*
E-mail: *verb73@gmail.com*
Received June 22, 2011; *revised July* 5, 2011; *accepted July* 19, 2011

Abstract

In this paper an encryption-decryption algorithm based on two moduli is described: one in the real field of integers and another in the field of complex integers. Also the proper selection of cryptographic system parameters is described. Several numeric illustrations explain step-by-step how to pre-condition a plaintext, how to select secret control parameters, how to ensure feasibility of all private keys and how to avoid ambiguity in the process of information recovery. The proposed public key cryptographic system is faster than most of known public key cryptosystems, since it requires a small number of multiplications and additions, and does not require exponentiations for its implementation.

Keywords: Ambiguity-Free Information Recovery, Complex Modulus, Cryptosystem Design, Cycling Identity, Information Hiding, Plaintext Preconditioning, Primary Residue, Public-Key Cryptography, Secret Controls, Threshold Parameters

1. Introduction and Primary Residues

This paper describes and briefly analyzes a public key cryptographic (PKC) based on primary residues and Gaussian modulus. The framework of the proposed PKC partially resembles NTRU PKC [1,2] {more details are provided in www.ntru.com} that was introduced in 1996 and later patented by three mathematicians from Brown University. Their PKC was analyzed in several papers [3-5]: in [3] it was pointed out that the decryption did not always recover the initial plaintext. Nevertheless, the NTRU had such a computational appeal that its authors were granted a USA patent even before the flaws in the algorithm were eliminated. Papers [4,5] provided several scenarios of cryptanalysis of the NTRU.

In this paper we consider a public key cryptographic system with two modulo reductions:

- Real integer modulus n and
- Complex (Gaussian) modulus R [6].

As a result, all public and private keys of each user and secret controls S are also Gaussians. Since plaintext blocks are also Gaussian, to avoid ambiguity in information recovery a concept of primary residues is introduced. It is demonstrated how to ensure that all keys of the proposed cryptosystem provide unambiguous recovery of initially pre-conditioned and subsequently-encrypted information.

In the proposed cryptosystem there is no necessity to consider polynomials with binary coefficients as it is done in papers [1] and [2].

1.1. Complex Modulo Reduction

Real modulus: In a group based on real modulo reduction n there are two results, whether n is either prime or composite: if $a \bmod n = b > 0$, then $a \bmod n = b - n \le 0$ is also correct.

In order to avoid ambiguity, we can stipulate that only non-negative results are feasible.

Complex modulus: Consider Gaussian integers $B := (b_1, b_2)$, and $R := (r_1, r_2)$. In an arithmetic based on modulo reduction with complex integer R there are four possible results: if $A \bmod R = B$, where both A and B are complex integers, then

$$A \bmod R = \{(b_1, b_2); (b_1 - r_1, b_2 - r_2); (b_1 + r_2, b_2 - r_1); (b_1 - r_1 + r_2, b_2 - r_1 - r_2)\}$$

are also correct. In order to avoid ambiguity in this case, it is stipulated in this paper that only *primary residues* are feasible {a definition and details are provided below}.

Let's define the norm N of R as

$$N := \|R\| := r_1^2 + r_2^2 \quad (1.1)$$

Then
$$\begin{aligned}(x, y) &:= (a, b) \bmod (r_1, r_2)\\ &= (a,b) - \lfloor (a,b)(r_1, -r_2)/N \rfloor (r_1, r_2)\end{aligned} \quad (1.2)$$

1.2. Primary Residues

Let's define two functions of integer variables x_1 and x_2 with integer parameters r_1 and r_2:

$$H(x_1, x_2) := r_1 x_2 - r_2 x_1; \quad (1.3)$$

and
$$V(x_1, x_2) := r_1 x_1 + r_2 x_2. \quad (1.4)$$

Definition 1.1 {primary residue}: A Gaussian integer $A = (a_1, a_2)$ is called a primary residue modulo R if it satisfies four inequalities:

$$0 \le H(a_1, a_2) \le N - 1; \quad (1.5)$$

and
$$0 \le V(a_1, a_2) \le N - 1. \quad (1.6)$$

Property 1.1: If a Gaussian integer G is a primary residue modulo Gaussian R, then

$$G \bmod R = G. \quad (1.7)$$

In the cryptographic scheme described below a plaintext M is divided onto pairs of blocks $M = (m_1, m_2)$, where each pair is treated as a Gaussian integer. In the cryptographic algorithm below, it is important to assure that the numeric representation $M = (m_1, m_2)$ is a *primary residue* modulo R.

1.3. Plaintext as Primary Residue

In general, $M = (m_1, m_2)$ is the *primary residue* modulo R, if m_1 and m_2 satisfy the following inequalities:

$$0 \le H(m_1, m_2) \le N - 1; \quad (1.8)$$

$$0 \le V(m_1, m_2) \le N - 1. \quad (1.9)$$

Remark 1.1: If both components in R are positive, then $(a_1, a_2) = (1, 0)$ is not a primary residue modulo R since (1.5) does not hold. Indeed, $(1,0) \equiv (1 - r_2, r_1) \bmod (r_1, r_2)$.

And, as a result, property (1.7) does not always hold.

However, **if** $r_2 < 0 < r_1$, then (1,0) is the primary residue.

If $r_2 < 0$, then (1.8) implies

$$0 \le r_1 m_2 + |r_2| m_1 < r_1^2 + r_2^2; \quad (1.10)$$

and (1.9) implies that

$$0 \le r_1 m_1 - |r_2| m_2 < r_1^2 + r_2^2. \quad (1.11)$$

Therefore, the right inequalities in (1.10) and (1.11) are respectively equivalent to

$$0 \le r_1(r_1 - m_2) + |r_2|(|r_2| - m_1)$$

and
$$0 \le r_1(r_1 - m_1) + |r_2|(|r_2| + m_2),$$

which hold **if**

$$m_2 \le r_1; m_1 \le |r_2|; \text{ and } m_1 \le r_1. \quad (1.12)$$

In addition, the left inequality in (1.11) holds **if**

$$(m_2/m_1) \le (r_1/|r_2|). \quad (1.13)$$

1.4. Geometric Interpretation

All primary residues are located inside a tilted square (rhomb) with vertices $(0,0); R; iR; (1+i)R$ and with sides equal
$\sqrt{r_1^2 + r_2^2}$ (1.1).

If $\gcd(r_1, r_2) = 1$, then there are exactly $N - 1$ primary residues inside this rhomb.

2. Cryptographic System Based on Primary Residues

1) All users (i = 1, 2, ⋯) agree to select a large real integer n {the same for all of them};

2) The i-th user has private and public keys, and secret controls P_i, R_i, U_i, Q_i, S_i with index i; {in the forthcoming discussion index i is omitted for the sake of simplicity of notations};

3) ***Variables***: P, R, U, Q, S, F, W, where each of them is a complex (Gaussian) integer;

4) User's ***private*** keys: $P = (p_1, p_2)$; $R = (r_1, r_2)$; where R is also a Gaussian prime

and
$$\gcd(p_1, p_2) = 1; \quad \gcd(r_1, r_2) = 1; \quad (2.1)$$

{the second condition in (2.1) holds if R is a Gaussian prime};

Remark 2.1: The stipulation that R is a Gaussian prime is sufficient to assure that certain conditions hold, but not necessary. Hence, it can be omitted under other considerations.

5) Every user pre-computes inverse

$$F := (f_1, f_2) := (p_1, p_2)^{-1} \bmod n \quad [7]; \quad (2.2)$$

Remark 2.2: a Gaussian multiplicative inverse F of P modulo real integer n exists

if
$$\gcd(\|P\|, n) = 1; \quad \{\|P\| := p_1^2 + p_2^2\}; \quad (2.3)$$

6) Every user pre-computes her/his *public* key

$$U := (u_1, u_2) := (f_1, f_2)(r_1, r_2) \bmod n; \quad (2.4)$$

7) Every user pre-computes a multiplicative inverse Q of P modulo Gaussian prime R:

$$Q := (q_1, q_2) := (p_1, p_2)^{-1} \bmod (r_1, r_2); \quad (2.5)$$

Multiplicative inverse Q of P modulo R exists if

$$\gcd(P, R) = (1, 0). \quad (2.6)$$

Remark 2.3: As demonstrated in [7], P has multiplicative inverse modulo R even if $\gcd(\|P\|, \|R\|) > 1$. For instance, although $\|(r_1, r_2)\| = \|(r_2, r_1)\|$,

yet $$\gcd[(r_1, r_2), (r_2, r_1)] = (1, 0). \quad (2.7)$$

Therefore, **if** R is a Gaussian prime, then every Gaussian is co-prime with R, *i.e.*, it has a multiplicative inverse modulo R. Primality of R is sufficient, but not necessary condition. The algorithm for computation of Q in (2.5) is provided below in Section 9.

Remark 2.4: Condition (1.13) is not directly verifiable by a sender since R is the private key of the receiver. Yet, the sender has an option to *indirectly* satisfy (1.13). Indeed, **if** $|r_2| \le r_1$ and $m_2/m_1 \le 1$, then (1.13) holds; otherwise switch m_1 and m_2 in M:

$$w_2 := m_1; \; w_1 := m_2 \quad (2.8)$$

Then, as a result,

$$(w_2/w_1) \le 1 \le (r_1/|r_2|). \quad (2.9)$$

Remark 2.5: Since r_1 and r_2 are design *parameters* of the cryptographic algorithm, they can be properly selected. On the other hand, m_1 and m_2 are *inputs* of the algorithm. As a result, a designer of this algorithm must ensure that both inequalities (1.11) hold for every pair (m_1, m_2) by partitioning the plaintext onto blocks of appropriate sizes.

Remark 2.6: In the forthcoming discussion it is assumed that $W := (w_1, w_2)$ is already *pre-conditioned* plaintext; *i.e.*, in every Gaussian block $w_1 \ge w_2$.

3. Hiding Information and Its Recovery

3.1. Threshold Parameter

Suppose that a sender (Sam) transmits a plaintext message $M = (m_1, m_2)$ to a receiver (Rene). The size of plaintext blocks m_1 and m_2 must be selected in such a way that

$$0 \le m_1, \; m_2 \le u \le r_1; \quad (3.1)$$

and plaintext M must be a primary residue modulo R {see (1.8-1.13)}. Here variable u (threshold) is the same for all users; its value is established below.

3.2. Sender's Secret Key

For security reason, the sender periodically selects a randomized secret key $S := (s_1, s_2)$. S plays two roles: it is a *screen/veil* that hides information; and at the same time it is a *control* that enables the system user to satisfy certain constraints. Proper selection of S is discussed below.

Encryption: Using Rene's public key U, Sam selects secret control S and computes ciphertext:

$$C := (M + SU) \bmod n. \quad (3.2)$$

Decryption: {requires real and Gaussian modulo reductions}:

Stage 1 {Real modulo n reduction}:

$$D := PC \bmod n; \quad (3.3)$$

Stage 2 {Gaussian modulo R reduction}:

$$Z := QD \bmod R. \quad (3.4)$$

3.3. Algorithm for Multiplicative Inverse of *P* Modulo Complex *R*

The algorithm computes the user's private key

$$Q = P^{-1} (\bmod R), \quad \text{where} \quad R = (p, q). \quad (3.5)$$

If R is a Gaussian prime, then

$$Q = P^{N-2} (\bmod R) \;, \text{ where } \; N = p^2 + q^2. \quad (3.6)$$

Computation (3.6) of multiplicative inverse (3.5) is based on the following identity.

Proposition 3.1 {cyclic identity}: If $\gcd[(a,b),(p,q)] = (1,0)$ and (p,q) is a prime, then the following identity holds:

$$(a,b)^{N-1} \bmod (p,q) = (1,0) \quad [6]. \quad (3.7)$$

4. Validation of Encryption-Decryption Algorithm

Proposition 4.1: If W is a primary residue and private keys P, R and secret control S are selected in such a way that holds

$$(PW + RS) \bmod n = PW + RS, \quad (4.1)$$

then in (3.4)

$$Z = W. \quad (4.2)$$

Proof: (2.2, 2.4, 2.5, 3.3 and 4.1) imply that

$$\begin{aligned} & PW + RS \xleftrightarrow{(4.1)} (PW + (1,0) RS) \bmod n \\ & \xleftrightarrow{(2.2)} [PW + (PF \bmod n) \times RS] \bmod n \\ & = P[W + (FR \bmod n) \times S] \bmod n \xleftrightarrow{(2.4)} \\ & P(W + US) \bmod n \xleftrightarrow{(3.2)} PC \bmod n \xleftrightarrow{(3.3)} D. \end{aligned} \quad (4.3)$$

Equation (4.3) holds since W, P, R, S are properly selected to ensure Equation (4.1).

Then

$$Z = QD \bmod R \xleftrightarrow{(4.3)} \left[Q\left(PW + RS\right)\right] \bmod R$$
$$= \left(QP \bmod R\right)\left(W \bmod R\right)$$
$$+ \left(QS \bmod R\right)\left(R \bmod R\right) \quad (4.4)$$
$$\xleftrightarrow{(2.5);\,(1.7)} \left(1, 0\right) \times W + \left(QS \bmod R\right) \times 0$$
$$= W.$$

Finally, the latter equality in (4.4) holds since W is a primary residue modulo R (1.5)-(1.7), *i.e.*, because $W \bmod R = W$. Q. E. D.

Proposition 4.2: if

● Absolute value of every component of private keys P and R is larger than threshold parameter $u = \sqrt{n/6}$ and does not exceed $2u$;

● Each component of plaintext W is positive and does not exceed u; and

● Absolute value of each component in secret control S does not exceed u,

then the encryption/decryption cryptosystem (3.2)-(3.4) provides unambiguous results.

5. Cryptosystem Design

Inputs m_1 and m_2 are independent variables known only to the sender (Sam). There are two types of variables: long-term static system parameters (strategic variables) and short-term dynamic controls (tactical variables): *System parameter n*; *Strategic variables P* and *R*; *Dynamic controls S*; and *Observable inputs*: W $\{w_1 > w_2\}$. Here it is assumed that plaintext (w_1, w_2) is already preconditioned; {more details are provided below}.

In addition, every W must be a primary residue for the receiver, *i.e.*, W and modulus R for every user must satisfy the following system of inequalities with eight integer variables:

$$0 \le w_1 \le u \le r_1;\ 0 \le w_2 \le u \le |r_2|; \quad (5.1)$$

$$0 \le r_1 w_2 + |r_2| w_1;\ r_1 w_2 + |r_2| w_1 < r_1^2 + r_2^2; \quad (5.2)$$

$$|r_2| w_2 \le r_1 w_1;\ r_1 w_1 < r_1^2 + r_2^2 + |r_2| w_2. \quad (5.3)$$

(5.1)-(5.3) are conditions that ensure that W is a primary residue modulo R.

If $s_1 < 0$ and $p_2 < 0$, then controls S and private key P must satisfy constraints:

$$0 \le p_1 w_1 - r_1 |s_1| + |p_2| w_2 + |r_2| s_2; \quad (5.4)$$

$$p_1 w_1 - r_1 |s_1| + |p_2| w_2 + |r_2| s_2 \le n; \quad (5.5)$$

$$0 \le p_1 w_2 - |p_2| w_1 + r_1 s_2 + |r_2||s_1|; \quad (5.6)$$

$$p_1 w_2 - |p_2| w_1 + r_1 s_2 + |r_2||s_1| \le n. \quad (5.7)$$

If (5.4)-(5.7) hold, then (4.1) also holds.

6. Equalizing the Feasibility Intervals

Notice that at most three terms in (5.5) and (5.7) are positive. Hence, if every product does not exceed $n/3$, then the sum of three terms does not exceed n.

Let

$$0 < w_k \le u;\ |s_k| \le u;\ u \le |p_k| \le v;\ u \le |r_k| \le v, \quad (6.1)$$

where u and v are unknown real numbers.

Hence $PC \le 3uv \le n$, *i.e.*, $uv \le n/3$. (6.2)

Select such u and v that the lengths of feasibility intervals for private keys P and R, secret key S and plaintext W are equal. Hence, $u = v - u$, which implies $2u = v$.

Thus,

$$2u^2 \le n/3 \quad (6.3)$$

which implies that

$$u \le \sqrt{n/6} \ \text{ and } \ v \le \sqrt{2n/3}. \quad (6.4)$$

Therefore, the following inequalities must hold:

$$0 < w_k \le \sqrt{n/6};\ |s_k| \le \sqrt{n/6};$$
$$\sqrt{n/6} \le |p_k| \le \sqrt{2n/3};\ \sqrt{n/6} \le |r_k| \le \sqrt{2n/3}. \quad (6.5)$$

Notice that Sam (the sender)

● Knows the input w_1 and w_2;

● Does not know P and R of Rene (the receiver);

● Dynamically selects controls s_1 and s_2.

Corollary 6.1: If $|p_i| w_j \le n/3$ and $|r_k| s_l \le n/3$, the value of each component in W and S is smaller than $\sqrt{n/6}$, p_1, $|p_2|$, r_1 and $|r_2|$ are on interval $\left[\sqrt{n/6}, \sqrt{2n/3}\right]$, then it ensures that W is a primary residue and that $\left(PW + RS\right) \bmod n = PW + RS$ (4.1).

Remark 6.1: By analogy with (5.1-5.3, 4.1) means that $PW + RS$ is a "primary residue" modulo n.

7. Plaintext Preconditioning and Recovery

Plaintext preconditioning: Compute

$$w_1 := m_1 + m_2; \quad (7.1)$$

if

$$m_1 \ge m_2, \text{ then } w_2 := m_1 - m_2 \text{ else } w_2 := m_2 - m_1 - 1. \quad (7.2)$$

Plaintext recovery: After decryption, the receiver compares parities of w_1 and w_2:

if $w_1 \equiv w_2 \pmod 2$, **then**

$$m_1 := \left(w_1 + w_2\right)/2 \ \text{ and } \ m_2 := w_1 - m_1; \quad (7.3)$$

Table 1. Public keys {*n*-real, *U*-Gaussian} and private keys {*P*, *Q*, *R*-all Gaussian}.

R Private key	*P Private key*	$Q = P^{-1}(\bmod R)$ *Private key*	$U = FR \bmod n$ *Public key*
$R = (2270, -2203)$	$P = (2291, -2180)$	$Q = (2858, 421)$	$U = (7624492, 258305)$

Table 2. {Encryption/Decryption}: $n = 10006001$; $0 \le W \le 1291$; $0 \le |S| \le 1291$.

$W = (w_1, w_2)$	$S = (s_1, s_2)$	$C = (W + SU) \bmod n$	$D = PC \bmod n$	$Z = QD \bmod R$	*Plaintext M Recovered*
(1223, 973)	(−859, 949)	(9511830, 9559186)	(5063750, 3609610)	(1223, 973)	(1098, 125)
(959, 941)	(−999, 1234)	(9149875, 5092460)	(4699221, 5067188)	(959, 941)	(950, 9)
(1234, 95)	(−954, 1285)	(8880702, 5324391)	(3699469, 2546137)	(1234, 95)	(569, 665)
(1267, 1201)	(−999, 1234)	(9150183, 5092720)	(5971649, 4991408)	(1267, 1201)	(1234, 33)
(18, 17)	(−16, 1291)	(4812437, 3187326)	(2886051, 2965525	(18, 17)	(0, 18)

else
$$m_1 := (w_1 - w_2 - 1)/2 \text{ and } m_2 := w_1 - m_1 = (w_1 + w_2 + 1)/2. \quad (7.4)$$

8. Numeric Illustrations

Let $n = 10006001$; the user's private keys *P*, *Q*, *R* and public key U are listed in **Table 1**. Here

$$\|P\| = \|(2291, -2180)\| = 10001081;$$

P is a primary residue modulo *R*; $\|R\|$ = 10006109; and feasibility threshold parameters are equal:
$u = \sqrt{n/6} = 1291$; and $2u = \sqrt{2n/3} = 2582$.

In **Table 2** every block of plaintext *W* is primary residue of *R*, and the following constraints are satisfied:

$$0 < |s_1| < s_2 \le \sqrt{n/6};\ 0 < w_2 \le w_1 \le \sqrt{n/6}.$$

Notice that for each of five blocks *W* we considered different secret controls *S*.

9. Algorithm for Multiplicative Inverse of *P* Modulo complex *R*

This algorithm computes

$$Q = P^{-1}(\bmod R),$$

where

$$R = (p, q). \ (1.1) \quad (9.1)$$

If *R* is a Gaussian prime, then $Q = P^{N-2}(\bmod R)$, where

$$N = \| R \| \quad (9.2)$$

If $R = R_1 R_2$, where each factor in *R* is a Gaussian prime, then

$$Q = P^{\varphi(N)-1} \bmod R, \quad (9.3)$$

where $\varphi(N)$ is Euler totient function and

$$N = \| R \| = \| R_1 \| \times \| R_2 \|. \quad (9.4)$$

Computation (9.2) and (9.3) of multiplicative inverse (9.1) is based on the following identity.

Proposition 9.1 {cyclic identity}: If
gcd[(*a*, *b*), (*p*, *q*)]=(1,0),
then the following identity holds:

$$(a,b)^{\varphi(\|(p,q)\|)-1} \equiv (a,b)^{-1}(\bmod(p,q)). \quad (9.5)$$

Proof follows from identity

$$(a,b)^{\varphi(\|(p,q)\|)} \bmod(p,q) = (1,0) \quad [6]. \quad (9.6)$$

Example 9.1: Suppose $R = (9,-2)$ and $P = (3,2)$; then $N = \|(9,-2)\| = 85$ and $\varphi(85) = 64$.

Hence,

$$Q = P^{-1}(\bmod R) = (3,2)^{63} \bmod (9,-2) = (5,-2).$$

Indeed,

$$(3,2)(5,-2) \equiv (1,0)(\bmod R).$$

Remark 9.1: The inverse of *P* can be also computed via solution of a Diophantine equation, but that is beyond the scope of this paper.

10. Computational Complexity

Encryption of each *W* requires three multiplications and five additions of real integers.

Decryption requires twice as many of these operations. Since addition/subtractions are much faster than multiplications, they can be neglected [8]. Therefore, we need nine multiplication of $\log\left(\sqrt{n}/2\right)$-digit long integers, which means that bit-wise complexity is of order $O\left(\log^2 n\right)$. This complexity can be reduced if we apply more elaborate algorithms for multiplication of multi-digit long real integers [9,10].

11. Conclusions

In this paper an encryption-decryption algorithm based on real and complex modulo reductions is considered and analyzed. A concept of primary residues is introduced to avoid ambiguity in information recovery. Several numeric illustrations explain step-by-step how to pre-condition a plaintext, how to select public and private keys for every user, and how to select secret controls for every block of the plaintext in order to ensure unambiguous recovery of the initial information. The proposed cryptosystem requires a small number of multiplications and additions, and as a result, it is extremely fast.

Although certain steps in the proposed cryptosystem resemble the NTRU cryptosystem, yet it differs from the NTRU in many features. One of them is absence of polynomials.

In paper [8] is provided a brief history on the NTRU, which is reiterated below. The NTRU that was initially presented at Crypto '96 was cryptanalyzed and broken in [11] by the method of lattice-basis reduction methods [12] that determines short vectors in a lattice, which arise on the decryption stage. Soon after that in papers [13] and [14] were described two other successful attempts to break the NTRU. An NTRU signature scheme was proposed in [15], but that scheme and its revision were broken in [16] and [17].

12. Acknowledgements

I express my appreciations to P. Garrett and C. Pomerance for suggestions on Gaussian modulus reduction, and to R. Rubino for comments that improved this paper. Numerical illustrations provided in this paper were facilitated thanks to programming support by S. Sadik and B. Saraswat.

13. References

[1] J. Hoffstein, J. Pipher and J. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," *Algorithmic Number Theory*: *3rd International Symposium* (*Lecture Notes in Computer Science*), Portland, Vol. 1423, 21-25 June 1998, pp. 267-288.

[2] J. Hoffstein, J. Pipher and J. Silverman, "NSS: An NTRU Lattice-Based Signature Scheme," *Advances in Cryptology—EUROCRYPT* 2001: *International Conference on the Theory and Application of Cryptographic Techniques* (*Lecture Notes in Computer Science*), Innsbruck, Vol. 2045, 6-10 May 2001, pp. 211-228.

[3] N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. Silverman, A. Singer and W. Whyte, "The Impact of Decryption Failures on the Security of NTRU Encryption," *Advances in Cryptology—CRYPTO* 2003: 23*rd Annual International Cryptology Conference* (*Lecture Notes in Computer Science*), Santa Barbara, Vol. 2729, 17-21 August 2003, pp. 226-246.

[4] D. Coppersmith and A. Shamir, "Lattice Attacks on NTRU," *Advances in Cryptology—EUROCRYPT* '97: *International Conference on the Theory and Application of Cryptographic Techniques* (*Lecture Notes in Computer Science*), Konstanz, Vol. 1233, 11-15 May 1997, pp. 52-61.

[5] E. Jaulmes and A. Joux, "A Chosen Ciphertext Attack against NTRU," *Advances in Cryptology—CRYPTO* 2000: 20*th Annual International Cryptology Conference* (*Lecture Notes in Computer Science*), Santa Barbara, Vol. 1880, 20-24 August 2000, pp. 20-35.

[6] B. Verkhovsky, "Protection of Sensitive Messages Based on Quadratic Roots of Gaussians: Groups with Complex Modulus," *International Journal Communications*, *Network and System Sciences*, Vol. 4, No. 5, 2011, pp. 287-296. doi:10.4236/ijcns.2011.45033

[7] B. Verkhovsky, "Cubic Root Extractors of Gaussian Integers and Their Application in Fast Encryption for Time-Constrained Secure Communication," *International Journal Communications*, *Network and System Sciences*, Vol. 4, No. 4, 2011, pp. 197-204. doi:10.4236/ijcns.2011.44024

[8] N. Koblitz and A. J. Menezes, "A Survey of Public-Key Cryptosystems," Research Report, Department of Combinatorics & Optimization, University of Waterloo, Waterloo, August 2004, pp. 1-47.

[9] A. L. Toom, "The Complexity of a Scheme of Functional Elements Realizing the Multiplication of Integers," *Soviet Mathematics Doklady*, No. 3, 1963, pp. 714-716.

[10] D. J. Bernstein, "Fast Multiplication and its Applications," In: J. P. Buhler and P. Stevenhagen, Eds., *Algorithmic Number Theory*: *Lattices*, *Number Fields*, *Curves and Cryptography*, MSRI, Cambridge University Press, New York, 2008, pp. 325-384.

[11] D. Coppersmith and A. Shamir, "Lattice Attacks on NTRU," *Advances in Cryptology*, EUROCRYPT 1997, *Lecture Notes in Computer Science*, Vol. 1233, Springer-Verlag, Berlin, 1997, pp. 52-61.

[12] A. K. Lenstra, H. W. Lenstra Jr. and L. Lovasz, "Factoring Polynomials with Integer Coefficients," *Mathematische Annalen*, Vol. 261, 1982, pp. 513-534.

[13] E. Jaulmes and A. Joux, "A Chosen Ciphertext Attack against NTRU," *Advances in Cryptology*, CRYPTO 2000, *Lecture Notes in Computer Science*, Vol. 1880, Springer-Verlag, Berlin, 2000, pp. 20-35.

[14] N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. Silverman, A. Singer and W. Whyte, "The Im-

pact of Decryption Failures On The Security of NTRU Encryption," *Advances in Cryptology*, CRYPTO 2003, *Lecture Notes in Computer Science*, Vol. 2729, Springer-Verlag, Berlin, 2003, pp. 226-246.

[15] J. Hoffstein, J. Pipher and J. Silverman, "NSS: An NTRU Lattice-Based Signature Scheme," *Advances in Cryptology*, EUROCRYPT 2001, *Lecture Notes in Computer Science*, Vol. 2045, Springer-Verlag, Berlin, 2001, pp. 211-228

[16] C. Gentry, J. Jonsson, M. Szydlo and J. Stern, "Cryptanalysis of the NTRU Signature Scheme (NSS) from Eurocrypt 2001," *Advances in Cryptology*, ASIACRYPT 2001, *Lecture Notes in Computer Science*, Vol. 2248, Springer-Verlag, Berlin, 2001, pp. 1-20.

[17] C. Gentry and M. Szydlo, "Analysis of the Revised NTRU Signature Scheme R-NSS," *Advances in Cryptology*, EUROCRYPT 2002, *Lecture Notes in Computer* Science, Vol. 2332, Springer-Verlag, Berlin, 2002, pp. 299-320.

