

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
9 July 2009 (09.07.2009)

PCT

(10) International Publication Number
WO 2009/084752 A1

(51) International Patent Classification:
H04L 9/30 (2006.01)

(74) Agent: KIM, Keon Woo; Davinci Patent & Law Firm, 302
DAERIM OFFICE VALLEY 97-3, Guro-dong, Guro-gu,
Seoul 152-050 (KR).

(21) International Application Number:
PCT/KR2007/006988

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA,
CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE,
EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID,
IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LA, LC, LK, LR,
LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX,
MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO,
RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(22) International Filing Date:
28 December 2007 (28.12.2007)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:
10-2007-0140162
28 December 2007 (28.12.2007) KR

(71) Applicant (for all designated States except US):
INHA-INDUSTRY PARTNERSHIP INSTITUTE
[KR/KR]; Inha Univ. 253 Yonghyeon-dong, Nam-gu,
Incheon 402-751 (KR).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **LEE, Mun-Kyu**
[KR/KR]; 2226-501, Hyosung Apt. Jindalrae Maeul,
Sang-dong, Bucheon si Wonmi-gu, Gyeonggi-do 420-755
(KR).

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL,
PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

(54) Title: A SECURE METHOD FOR CALCULATING A POLYNOMIAL CONVOLUTION OPERATION FOR AN NTRU CRYPTOSYSTEM

[Fig. 3]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.
Output: t the array where $t(X) = a(X) * c(X)$.
1: **for** $0 \leq j < 2N$ **do**
2: Generate a random r_j .
3: **end for**
4: **for** $0 \leq j < 2N$ **do**
5: $t_j \leftarrow r_j$
6: **end for**
7: **for** $0 \leq j < d$ **do**
8: **for** $0 \leq k < N$ **do**
9: $t_{k+|b[j]|} \leftarrow t_{k+|b[j]|} + c_k$
10: **end for**
11: **end for**
12: **for** $0 \leq j < 2N$ **do**
13: $t_j \leftarrow t_j - r_j$
14: **end for**
15: **for** $0 \leq j < N$ **do**
16: $t_j \leftarrow (t_j + t_{j-N}) \bmod q$
17: **end for**

(57) Abstract: A secure method is disclosed which calculates a polynomial convolution operation for an NTRU cryptosystem as a public-key cryptosystem. The secure method calculates a polynomial convolution operation with a first polynomial (which corresponds to a public key or a ciphertext) and a second polynomial arbitrarily selectable in NTRU encryption and decryption. The method comprising: (1) initializing respective elements of an array by an initial value that is not zero; and (2) storing the polynomial convolution operation result of the first and second polynomials in the array initialized by the initial value that is not zero. The secure method provides a secure cryptosystem designed so that a polynomial convolution operation, which is a primary operation of an NTRU cryptosystem, has resistance against power analysis attacks, and a method for blocking power analysis attacks, such as simple power analysis (SPA) and differential power attack (DPA). The secure method can prevent power analysis attacks without large overheads and thus efficiently perform a polynomial convolution operation, thereby maintaining the performance of a security system above a certain level.

WO 2009/084752 A1

Description

A SECURE METHOD FOR CALCULATING A POLYNOMIAL CONVOLUTION OPERATION FOR AN NTRU CRYPTOSYSTEM

Technical Field

- [1] The present invention relates to a method for calculating a polynomial convolution operation for an n -th degree truncated polynomial ring (NTRU) cryptosystem as a public-key cryptosystem. More particularly, this invention relates to a method for configuring a secure security system, through operation unification and randomization, against a power analysis attack that discovers the password information by measuring power consumption patterns in the security system.

Background Art

- [2] In recent years, with the exchange of private important information through wired/wireless networks for internet shopping, internet banking, and electronic funds transferred through bank cards, credit cards, and mobile communication terminals, the importance of protecting information has increased. In line with this, cryptosystems that establishes a theoretical basis for protecting private information have also gradually come under the spotlight. These cryptosystems have already been standardized by a variety of standardization organizations for about 30 years and implemented through a variety of methods, such as hardware and software.
- [3] The n -th degree truncated polynomial ring (NTRU) cryptosystem, recently developed, is a type of public key encryption that can provide a variety of security levels according to the choice of parameters. When the NTRU cryptosystem adjusts the level of the security to the same level as the conventional Rivest-Shamir-Adelman (RSA) and the conventional elliptic curve cryptosystem, it can perform encryption and decryption much faster than the conventional RSA and the conventional elliptic curve cryptosystem. Therefore, the NTRU cryptosystem has drawn attention as a next generation public key encryption. Recently, the IEEE has taken the NTRU cryptosystem into consideration as a next generation standard, i.e., IEEE P1363.1, which complements standard IEEE P1363 related to the conventional RSA and the conventional elliptic curve cryptosystem. The standardization has been substantially progressed and yielded Draft 9, in 2007.
- [4] Since the security of cryptosystem comes from key protection, it is important to safely protect keys in the system and to perform operations directly related to the key when cryptosystem is implemented, so that key-related information cannot be leaked. However, when a cryptosystem is implemented in a relatively small system, for

example, a smart card or an embedded system, etc., external attackers can accurately measure the power consumption patterns of these systems, due to the simplicity of the structure of these systems, and guess the key, which is called a power analysis attack. Although a theoretically secure cryptosystem is designed, the power analysis attack is a very strong attack method that can physically incapacitate the encryption during implementation. For the RSA and the elliptic curve cryptosystem, a variety of methods for protection against power analysis attack have been developed, however, similar methods for the NTRU cryptosystem have not yet been developed.

Disclosure of Invention

Technical Problem

- [5] The present invention solves the above problems, and provides: a secure cryptosystem designed so that a polynomial convolution operation, which is a primary operation of an NTRU cryptosystem, has resistance against power analysis attacks; and a method for blocking power analysis attacks, such as simple power analysis (SPA) and differential power attack (DPA).
- [6] The present invention further provides a method that can prevent power analysis attacks without large overheads and thus efficiently perform a polynomial convolution operation, thereby maintaining the performance of a security system above a certain level.

Technical Solution

- [7] In accordance with an exemplary embodiment of the present invention, the present invention provides a method for calculating a polynomial convolution operation with a first polynomial (which corresponds to a public key or a ciphertext) and a second polynomial arbitrarily selectable in n-th degree truncated polynomial ring (NTRU) encryption and NTRU decryption, the method including: (1) initializing respective elements of an array by an initial value that is not zero; and (2) storing the polynomial convolution operation result of the first and second polynomials in the array initialized by the initial value that is not zero.
- [8] Preferably, the initial value is the same value at all the elements of the array.
- [9] Preferably, the same value is identical to that of an operand in a modulo operation for the NTRU encryption.
- [10] Preferably, the initial value comprises randomly generated values for the respective elements of the array.
- [11] Preferably, the method may further include subtracting the randomly generated value from the polynomial convolution operation result stored in the array.
- [12] Preferably, the second polynomial includes a product-form polynomial.

Advantageous Effects

- [13] As described above, the polynomial convolution operation method, according to the present invention, can provide: a secure cryptosystem designed so that a polynomial convolution operation, which is a primary operation of an NTRU cryptosystem, has resistance against power analysis attacks; and a method for blocking power analysis attacks, such as simple power analysis (SPA) and differential power attack (DPA).
- [14] Also, the polynomial convolution operation method can provide a method that prevents power analysis attacks without large overheads and thus efficiently performs a polynomial convolution operation, thereby maintaining the performance of a security system above a certain level.
- [15] Additionally, the polynomial convolution operation method can provide security to: smart cards, such as credit cards, cash cards, and transportation cards; one time password (OTP), as one of the next generation securing means, with which conventional security cards (a table of random numbers) for internet banking are replaced; universal subscriber identity module (USIM) that must be used in the 3rd generation mobile communication, such as WCDMA, etc.; and embedded systems, etc.
- [16] Furthermore, the polynomial convolution operation method can provide security to all systems other than the foregoing systems listed above, susceptible to power analysis attack.

Brief Description of Drawings

- [17] The features and advantages of the present invention will be more apparent from the following detailed description in conjunction with the accompanying drawings, in which:
- [18] Figure 1 is a view illustrating an example of an algorithm for implementing a polynomial convolution operation, which is a primary operation in an NTRU cryptosystem, according to the present invention;
- [19] Figure 2 is a view illustrating an example of an algorithm for implementing a polynomial convolution method for protection against SPA attack, according to an embodiment of the present invention;
- [20] Figure 3 is a view illustrating an example of an algorithm for implementing a polynomial convolution method for protection against DPA attack, according to an embodiment of the present invention;
- [21] Figures 4 and 5 are graphs illustrating experimental results when the DPA attack is performed using the algorithm of Figure 1, which does not employ protection against power analysis attacks; and
- [22] Figures 6 and 7 are views illustrating examples of an algorithm for implementing a polynomial convolution method for protection against DPA attacks, according to another embodiment of the present invention.

[23]

[24] <Brief Description of Symbols in the Drawings>

[25] $c(X)$: a first polynomial corresponding to a 'public key' or 'ciphertext'[26] $a(X)$: a second polynomial that is arbitrarily selectable[27] $t(X)$: an array for storing polynomial convolution operation results[28] b : an array indicating which of the coefficients of $a(X)$, a binary polynomial having N coefficients, are 1[29] q : an operand in the modulo operation[30] r_j : an initial value generated at random**Best Mode for Carrying out the Invention**

[31] Hereinafter, exemplary embodiments of the present invention are described in detail with reference to the accompanying drawings.

[32] First, a polynomial convolution operation and NTRU public key cryptosystem will be briefly described, following the polynomial convolution operation method according to the present invention.

[33]

[34] 1. Polynomial convolution operation

[35] Let Z be a set of integers. A polynomial ring over Z , denoted $Z[X]$, becomes the set of all polynomials with coefficients in the integers. When quotient ring R is defined as $Z[X]/(X^N-1)$, this means a set of all possible remainder polynomials that may be created when an arbitrary polynomial with integer coefficients is divided by a polynomial X^N-1 . Therefore, an element a in R can be expressed as a polynomial or a vector, as the following equation (1).

$$[36] \quad a(X) = \sum_{i=0}^{N-1} a_i X^i = [a_0, a_1, \dots, a_{N-1}]$$

--- (1)

[37] The convolution product c of elements a and b in R has a coefficient expressed by the following equation (2).

$$[38] \quad c_k = \sum_{i=0}^k a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} = \sum_{i+j \equiv k \pmod{N}} a_i b_j$$

--- (2)

[39] Where $X^N = 1 \pmod{X^N - 1}$.[40] Technically, the operation requires N^2 integer multiplications, and this causes a large number of computations. However, for the polynomial convolution operation used by NTRU, one of elements a and b has small coefficients, so the computation of $a*b$ is very fast.

[41]

[42] 2. NTRU public key cryptosystem

- [43] A variety of NTRU cryptosystems exist, however, the improved version proposed by Hoffstein or Bailey is as follows:
- [44] - NTRU is set up by three public parameters, N , p , and q , where the greatest common divisor (gcd) of p and q is 1, i.e., $\gcd(p, q) = 1$, and $p \ll q$.
- [45] - Coefficients of the polynomial are reduced mod p or q
- [46] - Inverse of a polynomial f over mod q , denoted $f^{-1} \bmod q$, is defined as a polynomial satisfying $f * f^{-1} \equiv 1 \bmod q$, where mod q means that every coefficient in the polynomial is reduced mod q , i.e., every coefficient is calculated by the remainders after every coefficient is divided by q .
- [47] Working draft of standard IEEE P1363.1 proposes a set of typical parameters for an NTRU, one of which is $(N, p, q) = (251, 2, 197)$.
- [48]
- [49] 2.1 Key generation
- [50] Let's choose polynomials, F and g , with small coefficients in R and then calculate $f := 1 + pF$ and $h := pf^{-1} * g \bmod q$. Here, mod q means that every coefficient in the polynomial is reduced mod q , i.e., every coefficient is calculated by the remainders after every coefficient is divided by q . The private key is the polynomial f and the public key is the polynomial h .
- [51]
- [52] 2.2 Encryption
- [53] Let m be a polynomial representing a message. A polynomial r of $N-1$ degree with small coefficients is arbitrarily chosen and then ciphertext e ($e := r * h + m \bmod q$) is calculated.
- [54]
- [55] 2.3 Decryption
- [56] In order to decrypt the ciphertext e , the element a ($a := e * f \bmod q$) is first calculated. The coefficients of a are chosen so as to be $A \leq a_i < A + q$, where A is, as a fixed value, determined by a simple formula depending on the remaining parameters. After that, a plaintext m ($m := a \bmod q$) is recovered.
- [57]
- [58] 2.4 Validity of decryption
- [59] The polynomial a calculated at Section 2.3 satisfies the following equation (3).
- [60]
$$\begin{aligned} a &\equiv e * f \bmod q \\ &\equiv (r * h + m) * f \bmod q && (\text{since } e \equiv r * h + m) \\ &\equiv pr * g + m * f \bmod q && (\text{since } h * f \equiv pg * f^{-1} * f \equiv pg) \\ &\text{--- (3)} \end{aligned}$$
- [61] Let's take the final polynomial $pr * g + m * f$ into consideration. By appropriately

choosing parameters, the coefficients can be adjusted to fall within an interval less than q . Therefore, the polynomial a can be recovered as per the following equation (4).

$$[62] \quad a = pr * g + m * f = pr * g + m * (1 + pF) \\ \text{-- (4)}$$

[63] The m can be established not with respect to mod q but by a precise equality, i.e., $m = a \bmod p$.

[64] Based on the foregoing description, the polynomial convolution operation method of the present invention will be described in detail below.

[65] Figure 1 is a view illustrating an example of an algorithm for implementing a polynomial convolution operation, which is a primary operation in an NTRU cryptosystem, according to the present invention. This algorithm inputs a general polynomial $c(X)$ with integers, which are equal to or greater than 0 and smaller than a natural number q , and a binary polynomial $a(X)$ with coefficients of a binary number '1' or '0,' and then calculates a convolution product of the polynomials $c(X)$ and $a(X)$, $t(X)=a(X)*c(X)$. $c(X)$ is referred to as a first polynomial and $a(X)$ is referred to as a second polynomial. Here, $a(X)$ is replaced with an array b that represents the location of f coefficient '1' and then inputs them to the algorithm. The total number of coefficient of $a(X)$ and $c(X)$ is N (integer). Also, d denotes the number of elements in the array b .

[66] The binary polynomial $a(X)$ corresponds to a private key in decryption or to a randomization polynomial r in encryption. Since the decryption and encryption are an internal operation, they do not allow direct access from the external side. That is, it can be designed so that the external side cannot be aware of the array b that represents which of N coefficients of the binary polynomial $a(X)$ are 1. However, since only the general polynomial $c(X)$ is an open part, it allows the access from the external side and can be arbitrarily altered by attackers. Attackers can input their arbitrarily generated or altered data and enable operations to perform using the input data, so that they can analyze the internal power consumption patterns of the system at the external side and then indirectly derive key information.

[67] The following description is a method for preventing a polynomial convolution operation in an NTRU cryptosystem from attacking, according to a method for protection against simple power analysis (SPA) attack and a method for protection against differential power analysis (DPA) attack.

[68]

[69] (1) Protection against SPA attack

[70] In SPA attack for an NTRU, an attacker can detect a minute difference of power consumption as to the cases where one of x and y is zero and both x and y are not zero when an addition operation $x + y$ is performed in the system. Using this fact, the attacker establishes a general polynomial $c(X)$ whose coefficients are not zero and then

enables the system to perform a polynomial convolution to analyze key information.

[71] In particular, as described in Figure 1, rows 5 to 7 of the algorithm of the polynomial convolution operation for an NTRU are to perform an addition operation of two integers by N times. Since t_j are all initialized to zero, when j is set to zero ($j = \text{zero}$) at row 4, N operations at rows 5 to 7 become all additions whose one operand is zero. However, when j is set to 1 ($j = 1$) at row 4 at the next step, a part of the N operations become additions whose all operands are not zero and the remaining part of the N operations become additions where one of the operands is zero. When measurement is performed as to where the latter case occurs, a relative difference between the first element, $b[0]$, and the second element, $b[1]$ of array b can be obtained. When the attacker repeats this procedure, the relative differences between the adjacent elements of array b can be all acquired. When only the first element $b[0]$ is obtained by a simple exhaustive search, all elements in array b can be obtained to recover a key.

[72] For example, if $N=7$ and $b=[1, 4, 6]$, power consumption patterns for rows 5 to 7 in the algorithm of Figure 1 can be shown in time by order as follows.

[73] $j = 0$: ZN ZN ZN ZN ZN ZN ZN

[74] $j = 1$: NN NN NN NN ZN ZN ZN

[75] $j = 2$: NN NN NN NN NN ZN ZN

[76] Where, ZN denotes an addition operation of adding zero and a non-zero number, and NN denotes an addition operation of adding non-zero numbers.

[77] For $j=1$, three ZN's at the later part represent the relative difference between $b[0]$ and $b[1]$ is 3. For $j=2$, two ZN's at the later part represent the relative difference between $b[1]$ and $b[2]$ is 2. An attacker can guess b using only the power consumption pattern described above, without being aware of the contents of b . When the attacker guesses $b[0]$ as x , $b = [x, x+3, x+5]$ can be obtained. Therefore, although the attacker didn't know each element of b , only if one value, x , is obtained, all the elements of b can be substantially acquired. Therefore, when an exhaustive search is applied to the value x , b can be found out.

[78] It is understood that the attack described above can be applied to all convolution algorithms whose form is similar to that of the algorithm of Figure 1 as well as the algorithm described in Figure 1.

[79] Figure 2 is a view illustrating an example of an algorithm for implementing a polynomial convolution method for protection against SPA attack, according to an embodiment of the present invention. In this algorithm, every element in an array t is initialized not to zero but to q , so that additions with zero cannot be performed and all addition operations have similar power consumption patterns. Therefore, the SPA attack, which is performed based on the determination of the addition for zero, can be prevented. Also, since the algorithm of Figure 2 does not perform any additional

operation except for alteration at the initial step, its performance is not deteriorated at all although SPA prevention function is added thereto.

- [80] Rows 1 to 3 in the algorithm of Figure 2 correspond to a step for initializing every element of the array $t(X)$ to a non-zero initial value (q). The array $t(X)$ stores a polynomial convolution operation result of a first polynomial $c(X)$ corresponding to a 'public key' or a 'ciphertext' and a second polynomial $a(X)$ arbitrarily selectable. Also, rows 4 to 11 correspond to a step for storing the convolution operation result of the first polynomial $c(X)$ and the second polynomial $a(X)$ in the array $t(X)$ that was initialized by the non-zero initial value. In particular, row 2 in the algorithm of Figure 2 performs a step for storing the same initial value (q), not zero, in every element of the array $t(X)$ that will intend to store the polynomial convolution operation result. Also, since row 2 in the algorithm of Figure 2 sets the same initial value, not zero, to the same value as operand q of a modulo operation, it does not need a subtracting step where every element of the array is subtracted to the initial value used at the initializing step after the array stores the polynomial convolution operation result.

[81]

- [82] (2) Protection against DPA attack

- [83] After adding the SPA prevention function, the algorithm can be performed so that the addition of adding zero to a number cannot be distinguished from the addition of adding non-zero numbers, however, it may still generate a minute difference of power consumption by a difference between operands of the addition operation. This difference can be hardly detected by a simple attack method, such as SPA, but can be detected by an enhanced attack method, such as DPA. An attacker using DPA for an NTRU alters and applies a variety of general polynomials $c(X)$ for and to a constant binary polynomial $a(X)$ (which causes a constant array b) to measure the power consumption pattern. After that, the attacker statistically analyzes the measured power consumption pattern and discovers information of the array b . Therefore, since the attacker has an amount of power consumption data greater than that of the SPA, the attacker can accumulate minute differences, which cannot be detected by the power analysis at one time, and cannot therefore perform an attack with a larger accuracy.

- [84] More specifically, to accomplish the aim of discovering the elements in the array b , the attacker can detect a minute difference of power consumption between the case where the result value of $t_{k+b[j]} + c_k$ at row 6 in the algorithm of Figure 1 has a lowest bit of 0 and the case where the result value of $t_{k+b[j]} + c_k$ at row 6 in the algorithm of Figure 1 has a lowest bit of 1. Here, such a difference does not allow for detection by a one power measurement, such as SPA, rather it can be detected by performing operations for a variety of polynomials $c(X)$ and then accumulating and measuring the differences of power consumption. Performing such an attack is based on the rationale that the

addition for the same operand has the same power consumption pattern. Therefore, when the same operand can be calculated so as to always bring about different power consumption patterns, DPA can be prevented.

- [85] Figure 3 is a view illustrating an example of an algorithm for implementing a polynomial convolution method for protection against DPA attack, according to an embodiment of the present invention. This algorithm is designed in such a way that the same input brings about different power consumption patterns whenever the algorithm is performed, under the conditions that every element in an array t is initialized to random numbers. After completing the operation, a corresponding random number is removed to compensate the operation result. Every element in the array t may be initialized by the same value. Therefore, this algorithm can prevent a DPA attack operating on the assumption that the same input brings about the same power consumption.
- [86] Rows 1 to 3 in the algorithm of Figure 3 serve to perform a step for generating initial values, used for the initialization of every element in an array $t(X)$, at random. The array $t(X)$ stores a polynomial convolution operation result. Also, rows 4 to 6 serve to perform a step for initializing every element of the array $t(X)$ using initial values generated at random. Rows 12 to 14 serve to perform a step for subtracting a value that is generated at random from the stored polynomial convolution operation result.
- [87] Figures 4 and 5 are graphs illustrating experimental results when the DPA attack is performed using the algorithm of Figure 1, which does not employ protection against power analysis attack. The horizontal and vertical axes of the graphs denote time and the magnitude of a power signal, respectively. In the experiments, an attacker arbitrarily guesses $b[j]$, the j -th element in an array b , which is fixed but the attacker didn't know and then performs the algorithm of Figure 1 for a plurality of different polynomials $c(X)$. After that, the attacker sorts the result values $t_{k+b[j]} + c_k$ according to whether the lowest bit is 1 or 0 and then amplifies the difference between them. When the guess fails, the amplification signal does not show any particular signal but signals corresponding to random values, whose amplitudes are approximately zero, with respect to the horizontal axis symmetrically, as shown in Figure 4. On the contrary, when the guess is correct, a peak signal corresponding to a peak value appears at a certain portion as shown in Figure 5. The attacker repeatedly guesses possible values for $b[j]$, and sorts and amplifies processes, until a peak value from the guessed value appears.
- [88] When the DPA prevention method according to the present invention is applied to the algorithm, the case where the peak value does not appear as shown in Figure 5. Therefore, although an attacker performs a precise guess, a signal amplification graph similar to that of Figure 4 is obtained, so that the attacker's guessed value cannot be

checked as to whether it is correct or not, and thus the attacker cannot succeed in their attack.

- [89] Figures 6 and 7 are views illustrating examples of an algorithm for implementing a polynomial convolution method for protection against DPA attacks, according to another embodiment of the present invention. Each of these algorithms is designed in such a way that the same input brings about different power consumption patterns whenever the algorithm is performed, under the conditions that every element in an array t is initialized to random numbers. After completing the operation, a corresponding random number is removed to compensate the operation result. Every element in the array t may be initialized by the same value. Therefore, these algorithms can prevent DPA attacks conducted on the assumption that the same input brings about the same power consumption.
- [90] As the SPA prevention algorithm of Figure 2 and the DPA prevention algorithms of Figures 3, 6 and 7 are applied in the case where the polynomial $a(X)$ is a binary polynomial, they can be applied to a product-form polynomial. The product-form polynomial means a type of polynomial, such as $a(X) = a_1(X) * a_2(X)$, $a(X) = (a_1(X) * a_2(X) + a_3(X))$, or $a(X) = a_1(X) * a_2(X) * a_3(X)$, for a binary polynomial, $a_1(X)$, $a_2(X)$, and $a_3(X)$.
- [91] Although the preferred embodiments of the present invention have been disclosed for illustrative purposes, those skilled in the art will appreciate that various modifications, additions and substitutions are possible, without departing from the scope and spirit of the invention as disclosed in the accompanying claims.

Claims

- [1] A method for calculating a polynomial convolution operation with a first polynomial (which corresponds to a public key or a ciphertext) and a second polynomial arbitrarily selectable in n-th degree truncated polynomial ring (NTRU) encryption and NTRU decryption, the method comprising:
 - (1) initializing respective elements of an array by an initial value that is not zero; and
 - (2) storing the polynomial convolution operation result of the first and second polynomials in the array initialized by the initial value that is not zero.
- [2] The method according to claim 1, wherein the initial value is the same value at all the elements of the array.
- [3] The method according to claim 2, wherein the same value is identical to that of an operand in a modulo operation for the NTRU encryption.
- [4] The method according to claim 1, wherein the initial value comprises: randomly generated values for the respective elements of the array.
- [5] The method according to claim 4, further comprising:
subtracting the randomly generated value from the polynomial convolution operation result stored in the array.
- [6] The method according to claim 1, wherein the second polynomial comprises:
a product-form polynomial.

[Fig. 1]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.

Output: t the array where $t(X) = a(X) * c(X)$.

```

1: for  $0 \leq j < 2N$  do
2:    $t_j \leftarrow 0$ 
3: end for
4: for  $0 \leq j < d$  do
5:   for  $0 \leq k < N$  do
6:      $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$ 
7:   end for
8: end for
9: for  $0 \leq j < N$  do
10:   $t_j \leftarrow (t_j + t_{j+N}) \bmod q$ 
11: end for

```

[Fig. 2]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.

Output: t the array where $t(X) = a(X) * c(X)$.

```

1: for  $0 \leq j < 2N$  do
2:    $t_j \leftarrow q$ 
3: end for
4: for  $0 \leq j < d$  do
5:   for  $0 \leq k < N$  do
6:      $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$ 
7:   end for
8: end for
9: for  $0 \leq j < N$  do
10:   $t_j \leftarrow (t_j + t_{j+N}) \bmod q$ 
11: end for

```

[Fig. 3]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.

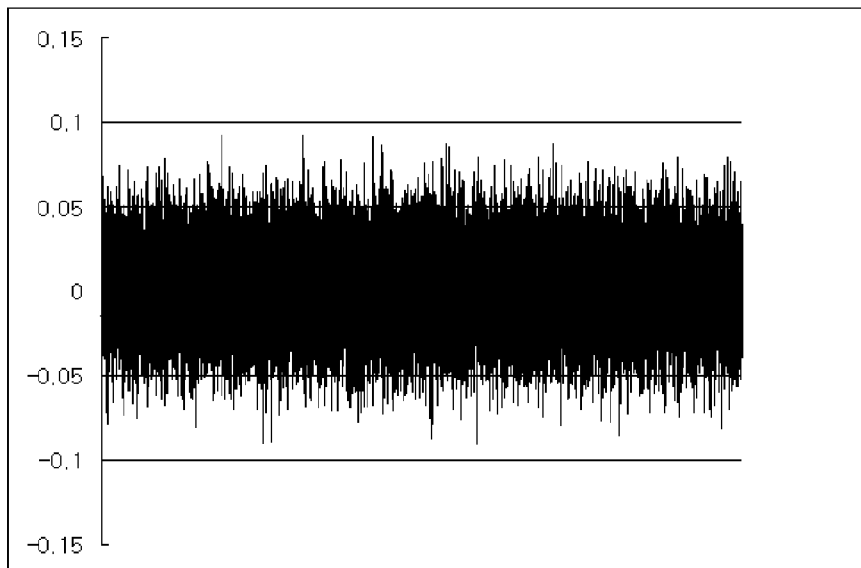
Output: t the array where $t(X) = a(X) * c(X)$.

```

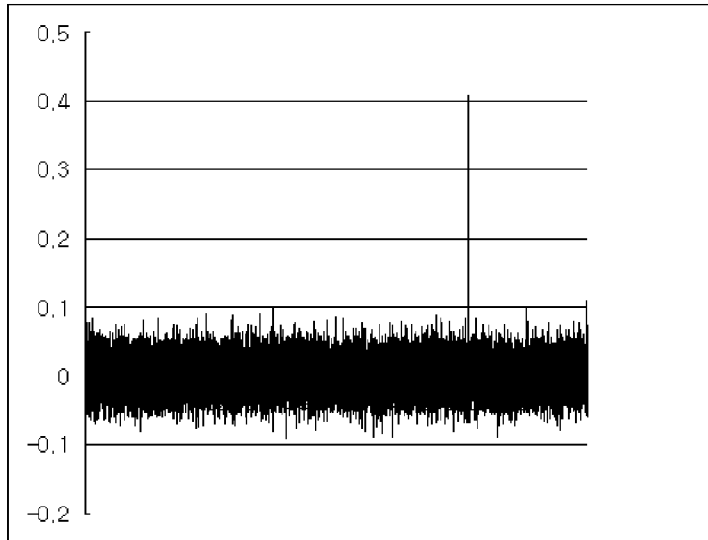
1: for  $0 \leq j < 2N$  do
2:   Generate a random  $r_j$ .
3: end for
4: for  $0 \leq j < 2N$  do
5:    $t_j \leftarrow r_j$ 
6: end for
7: for  $0 \leq j < d$  do
8:   for  $0 \leq k < N$  do
9:      $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$ 
10:  end for
11: end for
12: for  $0 \leq j < 2N$  do
13:    $t_j \leftarrow t_j - r_j$ 
14: end for
15: for  $0 \leq j < N$  do
16:    $t_j \leftarrow (t_j + t_{j+N}) \bmod q$ 
17: end for

```

[Fig. 4]



[Fig. 5]



[Fig. 6]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.

Output: t the array where $t(X) = a(X) * c(X)$.

```

1: for  $0 \leq j < N$  do
2:   Generate a random  $r_j$ .
3: end for
4: for  $0 \leq j < N$  do
5:    $t_j \leftarrow r_j$ 
6: end for
7: for  $0 \leq j < d$  do
8:   for  $0 \leq k < N$  do
9:      $t_{k+b[j] \bmod N} \leftarrow t_{k+b[j] \bmod N} + c_k$ 
10:  end for
11: end for
12: for  $0 \leq j < N$  do
13:    $t_j \leftarrow (t_j - r_j) \bmod q$ 
14: end for

```

[Fig. 7]

Input: b an array of d locations for '1' representing the polynomial $a(X)$;
 $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.

Output: t the array where $t(X) = a(X) * c(X)$.

```
1: for  $0 \leq j < N$  do
2:    $c_{j+N} \leftarrow c_j$ 
3: end for
4: for  $0 \leq j < N$  do
5:   Generate a random  $r_j$ .
6: end for
7: for  $0 \leq j < N$  do
8:    $t_j \leftarrow r_j$ 
9: end for
10: for  $0 \leq j < d$  do
11:   for  $0 \leq k < N$  do
12:      $t_k \leftarrow t_k + c_{k+N-b[j]}$ 
13:   end for
14: end for
15: for  $0 \leq j < N$  do
16:    $t_j \leftarrow (t_j - r_j) \bmod q$ 
17: end for
```

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2007/006988**A. CLASSIFICATION OF SUBJECT MATTER****H04L 9/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 8 H04L 9/30, H04L 9/28

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean Utility Models and applications for Utility Models since 1975

Japanese Utility Models and applications for Utility Models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

e-KIPASS(KIPO internal): "NTRU", "polynomial", "convolution"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 03-050998 A1 (NTRU CRYPTOSYSTEMS, INC.) 19 June 2003 See the abstract, claims 8-25, page 2, lines 10-19, page 7, lines 7-25, page 15, lines 4-25	1-6
A	Colleen Marie O'Rourke, Efficient NTRU Implementations. the paper of Master's Degree submitted to the faculty of the Worcester Polytechnic Institute, April 2002, pp. 12-19.	1-6
A	US6,081,597 A (JEFFREY HOFFSTEIN et al.) 27 June 2000 See the abstract and figure 3.	1-6
A	KR 10-2003-0043448 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 02 June 2003 See the abstract and claim 1.	1-6

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

21 AUGUST 2008 (21.08.2008)

Date of mailing of the international search report

21 AUGUST 2008 (21.08.2008)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

IM, DAE SHIK

Telephone No. 82-42-481-5947



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2007/006988

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 03-050998 A1	19.06.2003	AU 2002-353033 AA CA 2469198 A1 EP 1451967 A1 JP 2005-515659 T2 US 2003-120929 A1	23.06.2003 19.06.2003 01.09.2004 26.05.2005 26.06.2003
US 6081597 A	27.06.2000	US 6298137 B1	02.10.2001
KR 10-2003-0043448 A	02.06.2003	None	