Fast RSA-Type Cryptosystems Using N-Adic Expansion

Tsuyoshi Takagi

NTT Software Laboratories
3-9-11, Midori-cho Musashino-shi, Tokyo 180, Japan
E-mail: ttakagi@slab.ntt.co.jp

Abstract. We propose two RSA-type cryptosystems using n-adic expansion, where n is the public key. These cryptosystems can have more than one block as a plaintext space, and the decrypting process is faster than any other multi-block RSA-type cryptosystem ever reported. Deciphering the entire plaintext of this system is as intractable as breaking the RSA cryptosystem or factoring. Even if a message is several times longer than a public key n, we can encrypt the message fast without repeatedly using the secret key cryptosystem.

1 Introduction

The RSA cryptosystem is one of the most practical public key cryptosystems and is used throughout the world [17]. Let n be a public key, which is the product of two appropriate primes, e be an encryption key, and d be a decryption key. The algorithms of encryption and decryption consist of the e-th and d-th power modulo n, respectively. We can make e small by considering the low exponent attacks [3] [4] [7]. The encryption process uses less computation and is fast. On the other hand, we must keep the decryption key d up to the same size as the public key n to preclude Wiener's attack [21]. Therefore, the cost of the decryption process is dominant for the RSA cryptosystem.

If a cryptosystem has more than one block of plaintexts, where each block is as large as the public-key n, we call it a multi-block cryptosystem. A lot of multi-block RSA-type cryptosystems have been proposed [5] [11] [12] [13] [15] [19]. Their advantage is that they allow us to encrypt data larger than the public-key at a time, and we can prove their security is equivalent to the original RSA cryptosystem or factoring. However, these algorithms are very slow and the attacks against the RSA cryptosystem are also applicable to them (See, for example, [8] [20].). We cannot find significant advantage over using the original RSA cryptosystem for each block.

In this paper, we propose two methods of constructing fast multi-block RSA-type cryptosystems. We express the plaintext as an n-adic expansion, where n is the public key. The features of this method are as follows. We can take an arbitrary number of blocks as a plaintext. To implement the proposed cryptosystems, we use only ordinary and elementary mathematical techniques i.e., the greatest common divisor, so the designer can easily make them. Deciphering the entire

plaintext of the proposed cryptosystems is as hard as breaking the original RSA cryptosystem or factoring. Moreover, the decryption speed is much faster than any previously proposed multi-block RSA-type cryptosystems. Decryption time of the first block is dominant, because we calculate the modular multiplication of the encryption exponent and a greatest common divisor to decrypt blocks after the first one. Even if a message is several times longer than a public-key n, we can encrypt the message fast without repeatedly using the secret key cryptosystem.

Notation: $\mathbf{Z}$ is an integer ring. $\mathbf{Z}_n$ is a residue ring $\mathbf{Z}/n\mathbf{Z}$ and its complete residue class is $\{0, 1, 2, \ldots, n-1\}$. $\mathbf{Z}_n^{\times}$ is a reduced residue group modulo n. $\mathrm{LCM}(m_1, m_2)$ is the least common multiple of m_1 and m_2. $\mathrm{GCD}(m_1, m_2)$ is the greatest common divisor of m_1 and m_2. ${}_lC_m$ is permutation theory notation meaning the number of ways of choosing m from l.

2 The n-adic extension of RSA cryptosystem

In this section, we describe how to extend the RSA cryptosystem using n-adic expansion, and discuss its security and running time.

2.1 Algorithm

1. Generation of the keys: Generate two appropriate primes p, q, and let $n = pq$. Compute $L = \mathrm{LCM}\,(p-1, q-1)$, and find e, d which satisfies $ed \equiv 1 \pmod{L}$, $\mathrm{GCD}\,(e, L) = 1$ and $\mathrm{GCD}\,(e, n) = 1$. Then e, n are public keys, and d is the secret key.
2. Encryption: Let $M_0 \in \mathbf{Z}_n{}^{\times}$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$ be the plaintext. We encrypt the plaintexts by the equation:

$$C \equiv (M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^e \pmod{n^k}. \tag{1}$$

3. Decryption: First, we decrypt the first block M_0 by the secret key d:

$$M_0 \equiv C^d \pmod{n}. \tag{2}$$

This is the same decryption process as in the original RSA. For the remaining blocks $M_1, M_2, \ldots, M_{k-1}$, we can decrypt by solving the linear equation modulo n.

2.2 Details of decryption

Assume that we have already decrypted M_0 by the decryption method of the original RSA cryptosystem, and we write down the process to find $M_1, M_2, \ldots, M_{k-1}$ as follows.

Consider that the encryption function (1) is the polynomial of the variables $X_0, X_1, \ldots, X_{k-1}$ such that

$$E(X_0, X_1, \ldots, X_{k-1}) = (X_0 + nX_1 + \ldots + n^{k-1}X_{k-1})^e.$$

Expand the polynomial $E(X_0, X_1, \ldots, X_{k-1})$ by the polynomial theorem:

$$\sum_{\substack{0 \le s_0, s_1, \ldots, s_{k-1} \le e \\ s_0 + s_1 + \ldots + s_{k-1} = e}} \frac{e!}{s_0! s_1! \ldots s_{k-1}!} X_0^{s_0} (nX_1)^{s_1} \ldots (n^{k-1} X_{k-1})^{s_{k-1}}.$$

And let

$$\Gamma_i := \{(s_0, s_1, \ldots, s_i) \mid s_1 + 2s_2 + \ldots + is_i = i, \\ s_0 + s_1 + \ldots + s_i = e, 0 \le s_0, s_1, \ldots, s_i \le e\},$$

where $(0 \le i \le k-1)$. Let $D_i(X_0, X_1, \ldots, X_i)$ be the coefficient of $n^i (0 \le i \le k-1)$. For $i = 0, 1, \ldots, k-1$, we can find $D_i(X_0, X_1, \ldots, X_i)$ by calculating

$$D_i(X_0, X_1, \ldots, X_i) = \sum_{(s_0, s_1, \ldots, s_i) \in \Gamma_i} \frac{e!}{s_0! s_1! \ldots s_i!} X_0^{s_0} X_1^{s_1} \ldots X_i^{s_i}. \tag{3}$$

Here, we write them down with small i as follows:

$$D_0(X_0) = M_0^e,$$
$$D_1(X_0, X_1) = eM_0^{e-1} M_1,$$
$$D_2(X_0, X_1, X_2) = {}_eC_2 M_0^{e-2} M_1^2 + eM_0^{e-1} M_2,$$
$$D_3(X_0, X_1, X_2, X_3) = {}_eC_3 M_0^{e-3} M_1^3 + 2{}_eC_2 M_0^{e-2} M_1 M_2 + eM_0^{e-1} M_3,$$
$$D_4(X_0, X_1, \ldots, X_4) = {}_eC_4 M_0^{e-4} M_1^4 + 3{}_eC_3 M_0^{e-3} M_1^2 M_2 + {}_eC_2 M_0^{e-2} M_2^2 + eM_0^{e-1} M_4,$$
$$D_5(X_0, X_1, \ldots, X_5) = {}_eC_5 M_0^{e-5} M_1^5 + 4{}_eC_4 M_0^{e-4} M_1^3 M_2 + 3{}_eC_3 M_0^{e-3} M_1 M_2^2$$
$$+2{}_eC_2 M_0^{e-2} M_2 M_3 + 2{}_eC_2 M_0^{e-2} M_1 M_4 + eM_0^{e-1} M_5,$$
$$D_6(X_0, X_1, \ldots, X_6) = {}_eC_6 M_0^{e-6} M_1^6 + 5{}_eC_5 M_0^{e-5} M_1^4 M_2 + 4{}_eC_4 M_0^{e-4} M_1^3 M_3$$
$$+3{}_eC_3 M_0^{e-3} M_1^2 M_4 + {}_eC_3 M_0^{e-3} M_2^3 + {}_eC_2 M_0^{e-2} M_3^2$$
$$+2{}_eC_2 M_0^{e-2} M_2 M_4 + 2{}_eC_2 M_0^{e-2} M_1 M_5 + eM_0^{e-1} M_6,$$
$$\ldots$$
$$D_{k-1}(X_0, X_1, \ldots, X_{k-1}) = \{\text{polynomial of } M_0, M_1, \ldots, M_{k-1}\}.$$

We show the algorithm of decryption. Note that the terms that include X_i do not appear in D_j $(j < i)$, and the only term that includes X_i in D_i is $eX_0^{e-1} X_i$ for $i = 0, 1, \ldots, k-1$. We define

$$D_i'(X_0, X_1, \ldots, X_{i-1}) = D_i(X_0, X_1, \ldots, X_i) - eX_0^{e-1} X_i.$$

Therefore, the terms $D_0, D_1, \ldots, D_{i-1}, D_i'$ are the polynomial of $X_0, X_1, \ldots, X_{i-1} (0 \le i \le k-1)$.

From this relation, we can inductively decrypt M_i after decrypting $M_0, M_1, \ldots, M_{i-1} (0 \le i \le k-1)$. Indeed, $M_1, M_2, \ldots, M_{k-1}$ are calculated as follows. At first, let $i = 1$. The relations $D_1'(X_0) = 0$ and $D_0(X_0) = X_0^e$ hold. So, the solution of the linear equation

$$eM_0^{e-1} x \equiv B_1 \pmod{n}, \quad B_1 = E_1/n, \tag{4}$$

$$E_1 \equiv C - D_0(M_0) \pmod{n^2},$$

is M_1, because M_0 and e are in the reduced residue class modulo n such that $\mathbf{Z}_n{}^{\times}$. Provided that we decrypt $M_1, M_2, \ldots, M_{i-1}$, in the same manner we can uniquely decrypt M_i by solving the linear equation

$$eM_0^{e-1}x \equiv B_i \pmod{n}, \quad B_i = E_i/n^i, \tag{5}$$

$$E_i \equiv C - \sum_{j=0}^{i-1} n^j D_j(M_0, M_1, \ldots, M_j) - n^i D_i'(M_0, M_1, \ldots, M_{i-1}) \pmod{n^{i+1}}.$$

Inductively, we can decrypt all plaintexts $M_1, M_2, \ldots, M_{k-1}$.

Here, we describe the decryption program written in the pidgin ALGOL in the following. For $x \in \mathbf{Z}$ and positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is in $\{0, 1, \ldots, N-1\}$.

procedure **DECRYPTION**:
INPUT: $d, n, C(:= [(M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^e]_{n^k})$
OUTPUT: $M_0, M_1, \ldots, M_{k-1}$

(1) $C_0 := [C]_n$;
$M_0 := [C_0^d]_n$;
(2) $D_0 := [M_0^e]_{n^2}$;
$E_1 := [C - D_0]_{n^2}$;
$B_1 := E_1/n$ in $\mathbf{Z}$;
$A := [(eC_0)^{-1}M_0]_n$;
$M_1 := [AB_1]_n$;
(3) **FOR** $i = 2$ **to** $(k-1)$ **do**
begin
SUM $:= 0$;
FOR $j = 0$ **to** $(i-1)$ **do**
begin
$D_j := [D_j(M_0, M_1, \ldots, M_j)]_{n^{i+1}}$;
SUM $:= [\text{SUM} + n^j D_j]_{n^{i+1}}$
end
$D_i' := [D_i'(M_0, M_1, \ldots, M_{i-1})]_{n^{i+1}}$;
$E_i := [C - \text{SUM} - n^i D_i']_{n^{i+1}}$;
$B_i := E_i/n^i$ in $\mathbf{Z}$;
$M_i := [AB_i]_n$
end

2.3 Permutation

Let S be a finite set, and let $F(x)$ be a function from S to S. The function $F(x)$ is called a permutation function if every pair $x, y \in S$ that satisfies $F(x) = F(y)$ also satisfies $x = y$. If the encryption function $F(x)$ is not a permutation, we cannot uniquely decrypt a ciphertext. It is known that the encryption function of the RSA cryptosystem is a permutation, if and only if the relation $\mathrm{GCD}(e, L) = 1$ holds with the same notation as in section 2.1. In the previous section, we showed that if the conditions $\mathrm{GCD}(e, L) = 1$ and $\mathrm{GCD}(e, n) = 1$ are satisfied, the proposed cryptosystem can be uniquely decrypted i.e., it is a one-to-one function.

Here, the encryption function of the proposed cryptosystem is defined from ${\mathbf{Z}_{n^k}}^{\times}$ to ${\mathbf{Z}_{n^k}}^{\times}$. We can prove this function is a permutation if and only if the conditions $\mathrm{GCD}(e, L) = 1$ and $\mathrm{GCD}(e, n) = 1$ hold.

Actually, the reduced residue group modulo n^k such that ${\mathbf{Z}_{n^k}}^{\times}$ is decomposed into two products such that

$$ {\mathbf{Z}_{n^k}}^{\times} \cong {\mathbf{Z}_{p^k}}^{\times} \times {\mathbf{Z}_{q^k}}^{\times}. \tag{6} $$

Both groups are cyclic groups whose orders are $p^{k-1}(p-1)$ and $q^{k-1}(q-1)$, respectively. Therefore, the order of the group ${\mathbf{Z}_{n^k}}^{\times}$ is $n^{k-1}(p-1)(q-1)$. All elements in ${\mathbf{Z}_{n^k}}^{\times}$ are expressed by n-adic expansion such that

$$ M = M_0 + nM_1 + \ldots + n^{k-1}M_{k-1}, $$

where $M_0 \in {\mathbf{Z}_n}^{\times}$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$. This is the reason that the plaintext must have the form in equation (1).

Let $E(x) \equiv x^e \pmod{n^k}$ be the encryption function of the proposed cryptosystem. Suppose $E(x) \equiv E(y) \pmod{n^k}$, and we get $(x/y)^e \equiv 1 \pmod{n^k}$. By Chinese remainder theorem, we reduce the equation into modulo p^k. Let g be a primitive root of modulo p^k, and let $x/y \equiv g^j \pmod{p^k}$ for some j. We get $g^{je} \equiv 1 \pmod{p^k}$, and

$$ je \equiv 0 \pmod{p^{k-1}(p-1)}. $$

If $E(x)$ is a permutation, this equation must be solvable and all solutions are different, so $\mathrm{GCD}(e, p) = 1$ and $\mathrm{GCD}(e, (p-1)) = 1$ holds. Therefore, we have to choose e such that $\mathrm{GCD}(e, n) = 1$ and $\mathrm{GCD}(e, L) = 1$. The criteria in the key generation of the proposed cryptosystem are necessary.

2.4 Security

Theorem 1. *When plaintexts are uniformly distributed, finding the entire plaintext from the ciphertext for the RSA cryptosystem is as intractable as doing it for the proposed n-adic RSA-type cryptosystem.*

Proof. Using a black-box which can decipher the RSA cryptosystem, we can decipher the first block. Moreover, we can also decrypt blocks after the first one

by using the decryption algorithm in section 2.2, so the entire plaintext is deciphered. Conversely, we are given ciphertext C, which is the result of encrypting a random $M \pmod{n}$ by the RSA cryptosystem. Let C' be a random n-adic ciphertext, whose plaintext M' satisfies $M' \equiv M \pmod{n}$. All the bits of M' are uniquely distributed since M is random, and we can use the black box for the n-adic system to recover M'. Hence, we can decipher the plaintext M.

All the attacks against the RSA cryptosystem (See, for example, [14] [9].) are also applicable to the proposed system, because if we can decipher the first block M_0, then we can recover all the following blocks using relationships (4) or (5).

Here, we wonder whether the proposed cryptosystem has extra flaws in terms of using a non-square modulo n^k. The attacks that might break it are the message concealing [2] and the cycling attacks [22]. In the following two sections, we show these attacks never work against the proposed cryptosystem.

2.5 Message concealing

A function $F(x)$ is called unconcealed when $F(x) = x$ holds for all x. If a function of a cryptosystem is unconcealed, then we cannot encrypt any message by it. G. R. Blakley and I. Borosh showed that the encryption function of the RSA cryptosystem is unconcealed [2]. Let N be the number of residue classes x modulo n^k such that $x^e \equiv x \pmod{n^k}$. They proved

$$N = (1 + \mathrm{GCD}(e-1, p^{k-1}(p-1)))(1 + \mathrm{GCD}(e-1, q^{k-1}(q-1))).$$

If $\mathrm{GCD}(e-1, pq) > 1$ holds, then N becomes very large. We have to choose the system parameters p, q and e described in section 2.1 to preclude this failure. It must be noted that if e is selected smaller than p and q, then $\mathrm{GCD}(e-1, pq) = 1$ holds.

Moreover, they also showed that if e is an odd integer larger than 2, then $N = 9$ if and only if

$$\mathrm{GCD}(e-1, \lambda) = 2, \quad \lambda = \mathrm{LCM}((p-1)p^{k-1}, (q-1)q^{k-1}).$$

For example, the RSA cryptosystem has only 9 unconcealed messages if $\mathrm{GCD}(e-1, L) = 2$. For small e, we have $\mathrm{GCD}(e-1, pq) = 1$, and N for the proposed cryptosystem is equal to that of the RSA cryptosystem.

2.6 Cycling attacks

It is known that the RSA cryptosystem is broken without factoring n when a ciphertext C has a period such that $C^{P(b)} \equiv 1 \pmod{n}$, where $P(t)$ is a polynomial and $t = b$ is an integer. Actually, if the relation holds, the plaintext can be recovered by computing $M \equiv C^Q \pmod{n}$, where Q satisfies $eQ \equiv 1 \pmod{P'}$ and $P' = P(b)/\mathrm{GCD}(e, P(b))$. Moreover, this analysis is true even if the modulo n is changed to n^k. To break the proposed n-adic RSA-type cryptosystem, an

attacker would have to find the polynomial $P(t)$ and the value $t = b$, which have the relation $C^{P(b)} \equiv 1 \pmod{n^k}$. By decomposing of the group $\mathbf{Z}_{n^k}$ like (6), we reduce the relations to

$$P(t) \equiv 0 \pmod{p_i}, \quad P(t) \equiv 0 \pmod{q_i} \quad (i = 1, 2), \tag{7}$$

where $p_1 = p$; $p_2 = q$; and q_i is a large prime such that $q_i | p_i - 1 (i = 1, 2)$. H. C. Williams and B. Schmid [22] showed that the possibility of this polynomial satisfying equation (7) is very small, unless $P(t) = t \pm 1$ and $t = e^m$. Therefore, the designers must make m very large to preclude this attack. One method is to have $q_i - 1$ and $p_i - 1$ be divisible by large primes r_i and r_i' such that $r_i | p_i - 1$ and $r_i' | q_i - 1$; then $r_i | m$ and $r_i' | m$ hold for $i = 1, 2$ and m becomes very large. Since $p_i - 1 (i = 1, 2)$ must be divisible by a large prime to prevent the factoring algorithm called Pollard's $p - 1$ method, we do not need worry about the equations $e^m \equiv \pm 1 \pmod{p_i}$. Consequently, the proposed n-adic RSA-type cryptosystem is secure against this attack according to the same treatment as used for the original RSA cryptosystem.

2.7 Running time

Here, we discuss the running time of the proposed cryptosystem. In the encryption process, we have to compute the e-th power modulo $n^k (k \geq 2)$. As k increases, the running time becomes longer. However, it is possible to make the exponent of the encryption e small, since considering the low exponent attacks [3] [4] [7], the encryption cost is not so expensive.

Next, we consider the decryption process. The first block is decrypted by the same algorithm as in the RSA cryptosystem, and we should make the exponent d as large as the public modulus n to avoid Wiener's attack [21]. Therefore, the decryption of the first block is the most expensive task. After the first block, we have to generate linear equation (4) and maybe also (5), and solve it/them. The ciphertext $C_i (i \geq 1)$ is expressed by the polynomial of $M_i (i \geq 1)$ and the task of computing the polynomial is essentially to calculate M_0^e. Therefore, it costs the same as the encryption process to generate the linear equations. Solving a linear equation is fast, so the decryption time after the first block also becomes as fast as the encryption process. If we choose a very small e, this algorithm becomes very efficient. For example, let the number of blocks be two. We can generate the linear equation to compute equation (4), which are at most $2\lfloor \log_2 e \rfloor$ multiplications modulo n^2 and one division of n^2, and to solve it, which are two multiplications modulo n and one inversion modulo n.

On the other hand, several multi-block RSA-type cryptosystems have been proposed [5] [11] [13]. Their decryption time is l times slower than the original RSA cryptosystem, where l is the number of blocks. Our proposed cryptosystem is much faster than these cryptosystems, as showed by the above analysis. [1]

[1] K. Koyama proposed a two-block cryptosystem having fast decryption by using singular cubic curves. But it only has two blocks [12].

2.8 Effectiveness

As we discussed in the previous sections, the proposed n-adic RSA-type cryptosystem has several effective features. The most significant points are being as hard as breaking the original RSA cryptosystem and providing fast decryption for messages longer than the public key n.

By the way, the RSA cryptosystem is slower than the secret-key cryptosystem, so the RSA cryptosystem is used to encrypt a session key of the secret-key cryptosystem to overcome this disadvantage. However, its theoretical security level must be estimated from the RSA cryptosystem and the secret-key. We do not have to use the secret-key cryptosystem, if the length of the data is shorter than a public-key n.

For a message that is several times longer than the public-key n, our proposed n-adic RSA-type cryptosystem is very efficient. We can encrypt such a message much faster.

Moreover, it is expected that the encryption speed of the RSA cryptosystem will reach 1 Mbits/second within a year or so [18]. The proposed method can contribute to the attainment of the fast encryption speed.

3 The n-adic extension of Rabin cryptosystem

In this section, we describe how to extend the Rabin cryptosystem using n-adic expansion. The discussion is similar to the extension of the RSA cryptosystem.

3.1 Algorithm

1. Generating keys: Generate two appropriate primes p, q, and let $n = pq$. Here, p and q are the secret keys, and n is the public key.
2. Encryption: Let $M_0 \in \mathbf{Z}_n{}^{\times}$ and $M_1, \ldots, M_{k-1} \in \mathbf{Z}_n$ be the plaintext. We encrypt the plaintext by

$$C \equiv (M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^2 \pmod{n^k}. \tag{8}$$

   And we send the ciphertext C.
3. Decryption: We solve the modular quadratic equation

$$x^2 \equiv C \pmod{n^k}. \tag{9}$$

   Then the solutions are just plaintext $M_0, M_1, \ldots, M_{k-1}$.

3.2 Details of decryption

First, we decrypt the first block M_0. We solve the quadratic equation $C \equiv M_0^2$ modulo primes p and q. Here, several algorithms to solve the quadratic equation modulo a prime p are known, and the fastest one can be computed in subquadratic polynomial time [10]. Next, we decrypt the first block of the plaintext M_0 by the Chinese remainder theorem. The degree of ambiguity is 4 for the

decryption modulo n, because we have two solutions of each quadratic equation. And we can eliminate the ambiguousness by adding redundancy bits, and we can get the true plaintext.

Next, we discuss the decryption of the remaining blocks $M_1, M_2, \ldots, M_{k-1}$. The process is similar to the case in the RSA cryptosystem. For M_1, we have the linear equation modulo n^2,

$$M_0^2 + 2nM_0x \equiv C \pmod{n^2}. \tag{10}$$

And this equation is solvable because $2M_0 \in (\mathbf{Z}/n\mathbf{Z})^\times$, and the solution is M_1. Here, assume that we already decrypt $M_0, M_1, \ldots, M_{i-1}$, and we can uniquely decrypt M_i by solving

$$2n^iM_0x \equiv C - \sum_{0 \le l,m \le i-1}^{l+m \le i} n^{l+m}M_lM_m \pmod{n^{i+1}}, \tag{11}$$

Therefore, we can decrypt all plaintext blocks $M_0, M_1, M_2, \ldots, M_{k-1}$.

We describe the decryption program written in the pidgin ALGOL in the following. For $x \in \mathbf{Z}$ and positive integer N, $[x]_N$ will denote the remainder of x modulo N, which is $\{0, 1, \ldots, N-1\}$.

procedure **DECRYPTION**:
INPUT: $p, q, n, C(:= [(M_0 + nM_1 + \ldots + n^{k-1}M_{k-1})^2]_{n^k})$
OUTPUT: $M_0, M_1, \ldots, M_{k-1}$

(1) $C_0 := [C]_n$;
decrypt M_0 using p, q, C_0;
(2) **FOR** $i = 1$ **to** $(k-1)$ **do**
begin
SUM $:= 0$;
FOR $l = 0$ **to** $(i-1)$ **do**
FOR $m = 0$ **to** $(i-1)$ **do**
WHILE $l + m \le i$ **do**
begin
$D := [n^{l+m}M_lM_m]_{n^{i+1}}$;
SUM $:= [\text{SUM} + D]_{n^{i+1}}$
end
$E_i := [C - \text{SUM}]_{n^{i+1}}$;
$B_i := E_i/n^i$ in $\mathbf{Z}$;
$M_i := [(2M_0)^{-1}B_i]_n$
end

3.3 Security

Theorem 2. *Completely breaking the proposed n-adic Rabin-type cryptosystem is as intractable as factoring.*

Proof. Let p, q be primes, and let $n = pq$. The complexities of the following three algorithms only differ by polynomial time.

(I) to factor $n = pq$

(II) to find the solution of the quadratic equation modulo n

(III) to find the solution of the quadratic equation modulo n^k,

where k is an integer greater than 2. (I) and (II) are clearly equivalent because the security of the Rabin cryptosystem is the same as factoring [16]. (III) $\Rightarrow$ (II) is true by reducing the solution in (II) modulo n. (II) $\Rightarrow$ (III) is true because it is just the decryption process after the first block in the previous section, and the algorithm only takes polynomial time to generate and solve linear equations. Here, (III) is just the algorithm deciphering the proposed n-adic system.

The exponent of the Rabin cryptosystem is only 2, so the low exponent attacks are applicable to it [3] [4] [7]. However, we can preclude these attacks by padding a plaintext with random bits.

3.4 Running time and effectiveness

Here, we discuss the running time of the proposed cryptosystem. In the encryption process, we only compute the second power modulo $n^k (k \geq 2)$, which is very fast. For the decryption process, the first block is decrypted by the same decryption method as for the Rabin cryptosystem. The decryption of the first block is the most expensive task. After the first block, we have to generate the linear equation (10) and maybe also (11), and solve it/them. These are computed very fast, and the cost is very small compared with the cost of decrypting the first block. Therefore, the total cost of the decryption is essentially the cost of the first block.

On the other hand, several multi-block Rabin-type cryptosystems have been proposed [15] [19]. We have to solve a polynomial with more than two degrees over the finite field of a prime order. Solving polynomials of higher degree is more expensive than solving a quadratic polynomial, and makes the decryption process ambiguous and restricts the form of the secret primes. These cryptosystems have few advantages.

From the above analysis, our proposed cryptosystem is much faster than these cryptosystems, and easy to implement. Designers do not have to code a complicated algorithm and can use only ordinary mathematical tools such as the greatest common divisor.

As we discussed in section 2.8, for messages that are several times longer than the public-key n, our proposal n-adic Rabin cryptosystem is very efficient. We can encrypt a message with the running time of the first block.

4 Open problems and a partial solution

A plaintext of the proposed n-adic cryptosystem modulo n^k has the form $M \equiv M_0 + nM_1 + \ldots + n^{k-1}M_{k-1}$. Theorems 1 and 2 show that breaking the entire plaintext M is as hard as breaking the RSA cryptosystem or factoring. Here, we mention some problems concerning the security of each block $M_0, M_1, \ldots, M_{k-1}$.

If we have an algorithm that breaks the first block M_0, we can decipher the RSA or Rabin cryptosystem. However, it is an open problem whether you can find the blocks after the first one without deciphering the first block. One strategy for finding such an algorithm is to seek some algebraic relations between a ciphertext and blocks after the first one. Indeed, the most trivial relation is linear equation (4) or (5) whose solutions are the remaining blocks after the first one. But, we have to compute the value M_0^{e-1} to construct them, which is as hard as deciphering the RSA cryptosystem.

W. Alexi et al. showed that we can find the whole plaintext by using an algorithm that deciphers certain bits of the plaintext [1]. This also means that the proposed n-adic system can be broken by an algorithm that deciphers certain bits of the first block of the plaintext. It is an open problem whether there exists an algorithm that can decipher certain bits after first block of the plaintext.

Against the RSA cryptosystem, D. Coppersmith et al. showed that we can recover the original plaintext by algebraic calculation, if we send two ciphertexts whose plaintexts have a polynomial relationship [3]. It might be possible to recover the plaintext of the proposed n-adic system using a variation of this technique. It is an open problem whether you can recover the plaintext if there is a polynomial relationship between some blocks of one plaintext or between blocks of two plaintexts.

4.1 Security of the second block

Theorem 3. *Consider the n-adic RSA-type cryptosystem. Let $\mathcal{O}$ be an oracle which, given a ciphertext $C \equiv (M_0+nM_1+\ldots+n^{k-1}M_{k-1})^e \pmod{n^k}$, answers the second block of the plaintext M_1. The oracle $\mathcal{O}$ can be used to break the entire plaintext $(M_0, M_1, \ldots, M_{k-1})$.*

Proof. If we can decipher the first block M_0, then we can also do all the remaining blocks $M_2, \ldots, M_{k-1}$. Therefore, we can reduce the attack to the case of the two-block cryptosystem with modulo n^2. Let the plaintext $M = M_0 + nM_1$ $(0 \leq M_0, M_1 < n)$, and $C \equiv M^e \pmod{n^2}$ be the ciphertext. For $i = 0, 1, 2, \ldots, h$, expand

$$2^i M \equiv M_0^{(i)} + nM_1^{(i)} \pmod{n^2}, \quad 0 < M_0^{(i)}, M_1^{(i)} < n,$$

where $h = \lfloor \log_2 n \rfloor$. Here, $2^{ie}C \equiv (M_0^{(i)} + nM_1^{(i)})^e \pmod{n^2}$ holds, and we can get each second block $M_1^{(i)} = \mathcal{O}(2^{ie}C)$ by using the oracle $\mathcal{O}$. Here, note that $M_0^{(i)} < n/2$ if and only if $2M_1^{(i)} \pmod{n} = M_1^{(i+1)}$ for $i = 0, 1, 2, \ldots, h$. Hence $2M_1^{(i)} \pmod{n} = \mathcal{O}(2^{(i+1)e}C)$ if and only if $M_0^{(i)} < n/2$ for $i = 0, 1, 2, \ldots, h$. On

the other hand, let $C_0 \equiv C \pmod{n}$, and we have $2^{ie}C_0 \equiv (2^i M_0)^e \equiv (M_0^{(i)})^e \pmod{n}$ for $i = 0, 1, 2, \ldots, h$. This observation means that we can construct the half bit oracle $\mathcal{O}_H$, which computes $\mathcal{O}_H(2^{ie}C_0) = 0$ if $M_0^{(i)} < n/2$ and $\mathcal{O}_H(2^{ie}C_0) = 1$ if $M_0^{(i)} > n/2$. Indeed, define that

$$\mathcal{O}_H(2^{ie}C_0) = \begin{cases} 0, & (2\mathcal{O}(2^{ie}C) \pmod{n}) = \mathcal{O}(2^{(i+1)e}C), \\ 1, & (2\mathcal{O}(2^{ie}C) \pmod{n}) \neq \mathcal{O}(2^{(i+1)e}C), \end{cases}$$

for $i = 0, 1, 2, \ldots, h$. It is well-known this half bit oracle $\mathcal{O}_H$ recovers the plaintext M_0 such that $C_0 \equiv M_0^e \pmod{n}$[6]. Consequently, we can decipher the first block M_0.

5 Conclusion

Our proposed n-adic extensions of the RSA and Rabin cryptosystems perform decryption faster than any other multi-block RSA-type or Rabin-type cryptosystems ever reported. Deciphering the entire plaintext of this system is as intractable as breaking the original RSA cryptosystem or factoring. We also showed that the proposed n-adic RSA-type cryptosystem is a permutation function, and showed the criteria for message concealing and cycling attacks which are applicable to the RSA cryptosystem. Even if a message is several times longer than a public-key n, we can encrypt it fast without repeatedly using the secret-key cryptosystem.

Acknowledgments

I wish to thank S. Naito, M. Nishio, and members of the cryptology seminar in NTT Laboratories for their helpful discussions. I would also like to thank the anonymous referees for their valuable comments.

References

1. W. Alexi, B. Chor, O. Goldreich, C. P. Schnorr; "Rsa and Rabin functions: certain parts are as hard as the whole," SIAM Journal of Computing, 17, (1988), pp.194-209.
2. G. R. Blakley and I. Borosh, "Rivest-Shamir-Adelman public key cryptosystems do not always conceal messages," Comput. & Maths. with Appls., 5, (1979), pp.169-178.
3. D. Coppersmith, M. Franklin, J. Patarin and M. Reiter, "Low-exponent RSA with related messages," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.1-9.
4. D. Coppersmith, "Finding a small root of a univariate modular equation," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.155-165.
5. N. Demytko, "A new elliptic curves based analogue of RSA," Advances in Cryptology – EUROCRYPT '93, LNCS 765, (1994), pp.40-49.

6. S. Goldwasser, S. Micali, and P. Tong, "Why and how to establish a private code on a public network," Proc. of FOCS, (1982), pp.134-144.
7. J. Håstad, "Solving simultaneous modular equations of low degree," SIAM Journal of Computing, 17, (1988), pp.336-341.
8. B. S. Kaliski Jr., "A chosen message attack on Demytko's elliptic curve cryptosystem," Journal of Cryptology, 10, (1997), pp.71-72.
9. B. S. Kaliski Jr. and M. Robshaw, "Secure use of RSA," CRYPTOBYTES, 1 (3), (1995), pp.7-13.
10. E. Kaltofen and V. Shoup, "Subquadratic-time factoring of polynomials over finite fields", Proc. of STOC, (1995), pp.398-406.
11. K. Koyama, U. M. Maurer, T. Okamoto and S. A. Vanstone, "New public-key schemes based on elliptic curves over the ring $\mathbf{Z}_n$," Advances in Cryptology – CRYPTO '91, LNCS 576, (1991), pp.252-266.
12. K. Koyama; "Fast RSA-type schemes based on singular cubic curves," Advances in Cryptology – EUROCRYPT '95, LNCS 921, (1995), pp.329-340.
13. J. H. Loxton, D. S. P. Khoo, G. J. Bird and J. Seberry, "A cubic RSA code equivalent to factorization," Journal of Cryptology, 5, (1992), pp.139-150.
14. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of applied cryptography," CRC Press, (1996).
15. B. Meyer and V. Müller, "A public key cryptosystem based on elliptic curves over $\mathbf{Z}/n\mathbf{Z}$ equivalent to factoring," Advances in Cryptology – EUROCRYPT '96, LNCS 1070 (1996), pp.49-59.
16. M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization", Technical Report No.212, MIT, Laboratory of Computer Science, Cambridge (1979), pp.1-16.
17. R. Rivest, A. Shamir and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21(2), (1978), pp.120-126.
18. RSA Laboratories, "Frequently asked questions about today's cryptography (Version 3.0)," `http://www.rsa.com/rsalabs/`, (1996).
19. J. Schwenk and J. Eisfeld, "Public key encryption and signature schemes based on polynomials over $\mathbf{Z}_n$," Advances in Cryptology – EUROCRYPT '96, LNCS 1070, (1996), pp.60–71.
20. T. Takagi and S. Naito, "The multi-variable modular polynomial and its applications to cryptography," Proc. of ISAAC'96, LNCS 1178, (1996), pp.386-396.
21. M. J. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Transactions on Information Theory, IT-36, (1990), pp.553-558.
22. H. C. Williams and B. Schmid, "Some remarks concerning the M.I.T. public-key cryptosystem," BIT 19, (1979), pp.525-538.