Fast RSA-type Schemes Based on Singular Cubic Curves $y^2 + axy \equiv x^3 \pmod n$

Kenji Koyama

NTT Communication Science Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02 Japan
E-mail: koyama@cslab.kecl.ntt.jp

Abstract

This paper proposes fast RSA-type public-key schemes based on singular cubic curves $y^2 + axy = x^3$ over the ring Z_n. The x and y coordinates of a 2 log n-bit long plaintext/ciphertext are transformed to a log n-bit long shadow plaintext/ciphertext by isomorphic mapping. Decryption is carried out by exponentiating this shorter shadow ciphertext over Z_n. The decryption speed of the proposed schemes is about 2.0 times faster than that of the RSA scheme for a K-bit long message if $\lceil K/\log n \rceil$ is even. We prove that breaking each of the proposed schemes is computationally equivalent to breaking the RSA scheme in one-to-one communication circumstances. We also prove that the proposed schemes have the same security as the RSA scheme against the Hastad attack when linearly related plaintexts are encrypted in broadcast applications.

1 Introduction

In 1991, an RSA-type scheme over elliptic curves, i.e., non-singular cubic curves, was presented by Koyama, Maurer, Okamoto and Vanstone [4]. This scheme, the KMOV scheme for short, is more secure than the RSA scheme [9] against the Hastad attack [2] [6]. The decryption speed of the KMOV scheme, however, is 5.8 times slower than that of the RSA scheme even if rapid computational techniques are used [5].

By changing the base from elliptic curves to singular cubic curves, this paper proposes faster RSA-type schemes based on curves $E_n : y^2 + axy \equiv x^3 \pmod{n}$. The x and y coordinates of a 2 log n-bit long plaintext/ciphertext are transformed to a log n-bit long shadow plaintext/ciphertext by isomorphic mapping. Decryption is carried out by exponentiating this shorter shadow ciphertext over Z_n instead of a sequential addition of the points over singular cubic curves E_n. The decryption speed of the proposed schemes is about 2.0 times faster than that of the RSA scheme for a K-bit long message if $\lceil K/\log n \rceil$ is even. We prove that breaking each of the proposed schemes is computationally equivalent to breaking the RSA scheme. This equivalence in security is guaranteed under usual one-to-one communication circumstances. We also prove that the proposed schemes have the same security as the RSA scheme against the Hastad attack when linearly related plaintexts are encrypted in broadcast applications.

L.C. Guillou and J.-J. Quisquater (Eds.): Advances in Cryptology - EUROCRYPT '95, LNCS 921, pp. 329-340, 1995.

The organization of this paper is as follows. Section 2 mentions singular cubic curves over a finite field and a finite ring. In Section 3, we describe new schemes. The efficiency of the proposed and other schemes is discussed in Section 4. The security of the proposed schemes is discussed in Section 5. Section 6 concludes this paper.

2 Singular Cubic Curves

Let F_p be a finite field with p elements and F_p^* be a multiplicative group of F_p, where p (> 3) is a prime.

Definition 1 ([3][7]) *A non-singular part of a singular cubic curve, denoted by $E_p(a,b)$, is defined as the set of solutions $(x,y) \in F_p \times F_p$ to Eq.(1), excluding a singular point $(0,0)$ and including the point at infinity $\mathcal{O}$.*

$$y^2 + axy = x^3 + bx^2 \ \text{over}\ F_p, \quad a, b \in F_p \tag{1}$$

An addition "$\oplus$" on $E_p(a,b)$ is given by the chord-and-tangent law similar to that for elliptic curves.

The sum (x_3, y_3) of (x_1, y_1) and (x_2, y_2) in F_p is computed as

$$\begin{cases} x_3 = & \lambda^2 + a\lambda - b - x_1 - x_2, \\ y_3 = & \lambda(x_1 - x_3) - y_1, \end{cases} \tag{2}$$

where

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } (x_1, y_1) \neq (x_2, y_2), \\ \dfrac{3x_1^2 + 2bx_1 - ay_1}{2y_1 + ax_1} & \text{if } (x_1, y_1) = (x_2, y_2). \end{cases}$$

Note that $E_p(a,b)$ is a group. Operation $\otimes$ is defined as follows.

$$k \otimes (x,y) = \overbrace{(x,y) \oplus \cdots \oplus (x,y)}^{k \text{ times}} \text{ over } E_p(a,b).$$

A group $E_p(a,b)$ is isomorphic to F_p^*. The isomorphic relationship is generally described in [3] and [7] for curves $(y - \alpha x)(y - \beta x) = x^3$ over F_p^*, where $\alpha, \beta \in F_p^*$, which is equivalent to equation (1) with $a = -\alpha - \beta \bmod p$, $b = -\alpha\beta \bmod p$. When $b = 0$, we can put $\alpha = 0$ and $\beta = -a (\neq 0)$, and the simplified relationship is carried out explicitly in the following theorem.

Theorem 1 *The mapping $\omega : E_p(a,0) \to F_p^*$ defined by*

$$\omega : \mathcal{O} \mapsto 1, \quad (x,y) \mapsto 1 + \frac{ax}{y} = \frac{x^3}{y^2}$$

is a group isomorphism. The group isomorphism mapping $\omega^{-1} : F_p^* \to E_p(a,0)$ is defined by

$$\omega^{-1} : 1 \mapsto \mathcal{O}, \quad v \mapsto \left(\frac{a^2 v}{(v-1)^2}, \frac{a^3 v}{(v-1)^3} \right).$$

Hence, an order of $E_p(a,0)$, denoted by $\#E_p(a,0)$, is $p-1$.

Let $Z_n = \{0, 1, \cdots, n-1\}$ and Z_n^* be a multiplicative group of Z_n. A non-singular part of a singular cubic curve over Z_n is defined as follows.

Definition 2 *Let n be a product of primes p, q (> 3). A non-singular part of a singular cubic curve, denoted by $E_n(a,b)$, is defined as the set of solutions $(x,y) \in Z_n \times Z_n$ to Eq.(3), excluding a singular point $(0,0)$ and including the point at infinity $\mathcal{O}$.*

$$y^2 + axy = x^3 + bx^2 \text{ over } Z_n, \quad a, b \in Z_n. \tag{3}$$

An addition on $E_n(a,b)$ is defined by the chord-and-tangent law. Although the addition is not always defined, the probability for such a case is negligibly small for large p and q. By Theorem 1 and the Chinese Remainder Theorem, the following theorem holds.

Theorem 2 *For (x_i, y_i) and (x_1, y_1) satisfying $(x_i, y_i) = i \otimes (x_1, y_1)$ over $E_n(a,0)$, we have*

$$1 + \frac{ax_i}{y_i} \equiv \left(1 + \frac{ax_1}{y_1}\right)^i \pmod{n},$$

i.e.,

$$\frac{x_i^3}{y_i^2} \equiv \left(\frac{x_1^3}{y_1^2}\right)^i \pmod{n}.$$

The following theorem is a base of a pair of an encryption and a decryption of public-key cryptosystems over $E_n(a,0)$.

Theorem 3 *Let n be a product of primes p, q (> 3) and $N = \mathrm{lcm}(p-1, q-1)$. For any integer k satisfying $k \equiv 1 \pmod{N}$, we have*

$$(x,y) = k \otimes (x,y) \text{ over } E_n(a,0)$$

with the overwhelming probability for large p and q.

3 New RSA-type Schemes Based on $E_n(a,0)$

We can construct RSA-type public-key schemes over singular cubic curves $E_n(a,b)$ with a message-dependent variable a and a fixed constant b. Considering the computational efficiency among variants of instances of these schemes, we put $b = 0$. We propose two new RSA-type schemes over $E_n(a,0)$: scheme 1 and scheme 2. These proposed schemes can be used in both secret communications and digital signatures. For simplicity, we describe protocols of secret communications.

The security of the proposed schemes is based on the difficulty of factoring n, which is a product of large primes p and q. Let a plaintext (m_x, m_y) be an integer pair, where $m_x, m_y \in Z_n^*$ and $m_x^3 \not\equiv m_y^2 \pmod{n}$. A concept of RSA-type schemes based on isomorphism over singular cubic curves is shown in Figure 1. This figure also includes a flow diagram of scheme 1. In scheme 1, the encryption is carried out over $E_n(a,0)$ along the path from plaintext (m_x, m_y) to ciphertext (c_x, c_y). In scheme 2, the encryption is carried out over Z_n^* along the path from plaintext (m_x, m_y) to shadow ciphertext c via shadow plaintext m. Although the decryption of naive cryptosystems based on cubic curves is computed directly from (c_x, c_y) to (m_x, m_y) over E_n $(= E_p \times E_q)$ in the left half of Figure 1, the decryptions for schemes 1 and 2 are carried out over F_p^* and F_q^* because decryption over F_p^* and F_q^* is faster than that over $E_p(a,0)$ and $E_q(a,0)$.

Note that for the original RSA scheme, the encryption and decryption are carried out between (shadow) plaintext m and (shadow) ciphertext c in Z_n^*, more exactly in Z_n, in the right half of Figure 1.

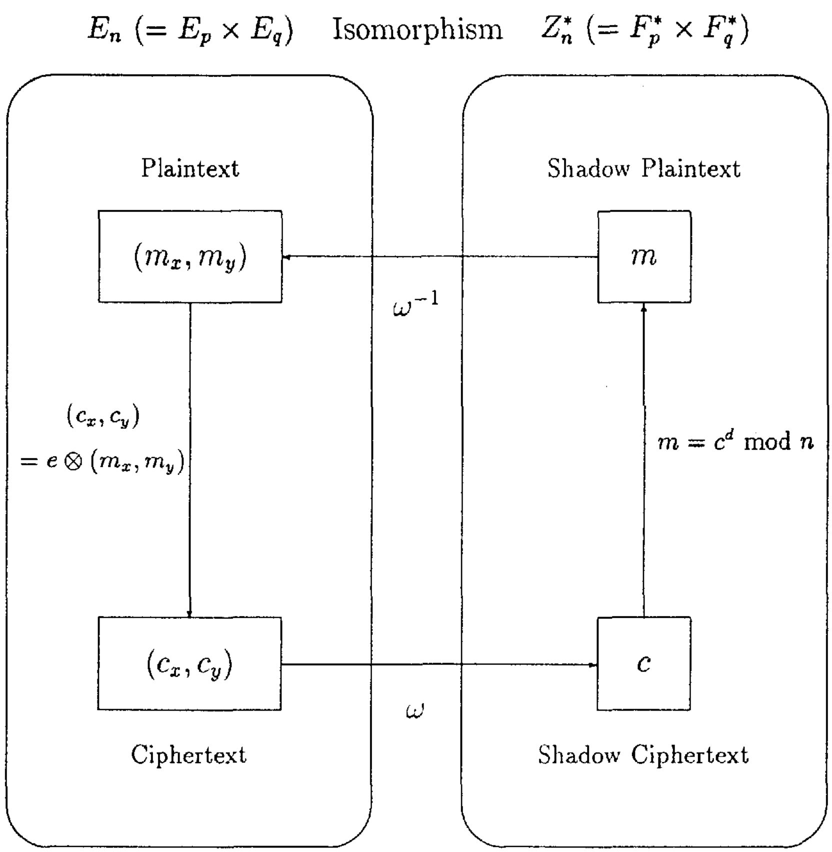

Fig.1 Concept of RSA-type schemes and a flow of scheme 1

3.1 Key Generation of Scheme 1 and Scheme 2

A key generation procedure is common for scheme 1 and scheme 2.

Receiver R chooses two large primes p and q. Let $n = pq$ and $N = \mathrm{lcm}(p-1, q-1)$. R determines an integer e satisfying $\gcd(e, N) = 1$. Decryption keys d_p and d_q are computed from encryption key e as $d_p = \frac{1}{e} \bmod (p-1)$ and $d_q = \frac{1}{e} \bmod (q-1)$, respectively. R's public keys are e and n. R's secret keys are p, q, d_p and d_q.

3.2 Scheme 1

Encryption
Sender S encrypts plaintext (m_x, m_y) with the receiver's public keys e and n as

$$(c_x, c_y) = e \otimes (m_x, m_y) \text{ over } E_n(a, 0),$$

where $a = \dfrac{m_x^3 - m_y^2}{m_x m_y} \bmod n$, and sends a ciphertext (c_x, c_y) to receiver R.
Remark
· Plaintext condition such that $m_x, m_y \in Z_n^*$ and $m_x^3 \not\equiv m_y^2 \pmod n$ holds true with overwhelming probability for large primes p and q and uniformly distributed integers m_x and m_y.

Decryption
Receiver R decrypts ciphertext (c_x, c_y) with secret keys p, q, d_p and d_q. First, R computes $c_{xp} = c_x \bmod p$, $c_{yp} = c_y \bmod p$ and shadow ciphertext $c_p = \dfrac{c_{xp}^3}{c_{yp}^2} \bmod p$ by using the isomorphic mapping ω in Theorem 1. R computes shadow plaintext m_p as

$$m_p = c_p^{d_p} \bmod p = \left(\frac{c_{xp}^3}{c_{yp}^2}\right)^{d_p} \bmod p. \tag{4}$$

R computes $(m_{xp}, m_{yp}) \in E_p(a_p, 0)$ with $a_p = \dfrac{c_{xp}^3 - c_{yp}^2}{c_{xp} c_{yp}} \bmod p$ by using the isomorphic mapping ω^{-1} in Theorem 1 as

$$m_{xp} = \frac{a_p^2 m_p}{(m_p - 1)^2} \bmod p, \quad m_{yp} = \frac{m_{xp} a_p}{(m_p - 1)} \bmod p.$$

R computes $(m_{xq}, m_{yq}) \in E_q(a_q, 0)$ in the same way. Finally, R obtains (m_x, m_y) by combining (m_{xp}, m_{yp}) and (m_{xq}, m_{yq}) via the Chinese Remainder Theorem.
Remarks
· By the isomorphic mappings in Theorem 1, computing $d_p \otimes (c_{xp}, c_{yp})$ over $E_p(a_p, 0)$ corresponds to computing $(c_{xp}^3 / c_{yp}^2)^{d_p}$ over F_p^*. The decryption of scheme 1 corresponds to the path from (c_x, c_y) to (m_x, m_y) via c and m.
· Since $m_{xp}, m_{yp} \in F_p^*$ and $m_{xp}^3 \not\equiv m_{yp}^2 \pmod p$, we have $m_p \neq 1$.

3.3 Scheme 2

Encryption
Sender S encrypts plaintext (m_x, m_y) with the receiver's public keys e and n as

$$c = \left(\frac{m_x^3}{m_y^2}\right)^e \bmod n,$$

$$a = \frac{m_x^3 - m_y^2}{m_x m_y} \bmod n,$$

and sends a pair (c, a) of shadow ciphertext c and the corresponding variable a to receiver R.

Remark

· The length of the transmitted message in scheme 2 is the same as that in scheme 1, which is 2 log n bits.

Decryption

Receiver R decrypts shadow ciphertext (c, a) with secret keys p, q, d_p and d_q. First, R computes $c_p = c \bmod p$ and shadow plaintext m_p from c_p, d_p and p as

$$m_p = c_p^{d_p} \bmod p. \tag{5}$$

R computes $(m_{xp}, m_{yp}) \in E_p(a_p, 0)$ with $a_p = a \bmod p$ by using the isomorphic mapping ω^{-1} in Theorem 1 as

$$m_{xp} = \frac{a_p^2 m_p}{(m_p - 1)^2} \bmod p, \quad m_{yp} = \frac{m_{xp} a_p}{(m_p - 1)} \bmod p.$$

R computes $(m_{xq}, m_{yq}) \in E_q(a_q, 0)$ in the same way. Finally, R obtains (m_x, m_y) by combining (m_{xp}, m_{yp}) and (m_{xq}, m_{yq}) via the Chinese Remainder Theorem.

Remarks

· The decryption of scheme 2 corresponds to the path from c to (m_x, m_y) via m.

· Computations of c_p and a_p in the decryption of scheme 2 need less time than that of scheme 1 because divisions of $c_p = c_{xp}^3 / c_{yp}^2$ and $a_p = (c_{xp}^3 - c_{yp}^2)/c_{xp} c_{yp}$ can be avoided.

4 Efficiency

4.1 Comparison of Proposed Schemes and Other Schemes

Since encryption key e can be set as a small value and decryption keys d_p, d_q are large enough such that $\log d_p \approx \log p$, $\log d_q \approx \log q$, we focus on the decryption procedure. We evaluate the average number of modular multiplications for decryption. Here, we assume $\log p \approx \log q$.

In the proposed schemes, i.e., scheme 1 and scheme 2, the dominant computations involve equations (4) and (5). They require $1.5 \log p$ multiplications modulo p on average. Including the $1.5 \log q$ multiplications modulo q, the decryption of each of the proposed schemes requires about $3 \log p$ modular multiplications.

The block size for the RSA scheme is log n bits, and that for the proposed schemes is 2 log n bits. The number of modular multiplications in the new schemes and previously proposed schemes are shown in Table 1. We define "speed ratio"; the bigger the speed ratio is, the faster the decryption speed is. Let the decryption speed ratio of the RSA scheme be normalized to 1.0. When a K-bit long message is given, the speed ratios for the KMOV scheme and the new schemes are determined as $0.085r$ and r, respectively, where $r = s / \lceil \frac{s}{2} \rceil$ and $s = \lceil K / \log n \rceil$. Note that $1.0 \leq r \leq 2.0$. When integer s is even, the speed ratios for the KMOV scheme and the new schemes are fixed as 0.17 and 2.0, respectively. If message length K is uniformly distributed, the probability that s is even is 1/2. If message length K is predetermined such that $K = 2 \log n$, then integer s is always even. For the Demytko scheme based on elliptic curves [1], its speed ratio is always fixed as 0.14 because the block size is log n. These results are summarized in Table 1. We can observe that the decryption speed of the proposed schemes is about 2.0 times faster than that of the RSA scheme for a K-bit long message if $\lceil K / \log n \rceil$ is even.

Table 1: Efficiency of decryptions

Cryptosystems	Block size	No. of mod. multi.	Speed ratio ($\lceil K/\log n \rceil$ is even)
RSA	$\log n$	$3 \log p$	1.0
KMOV	$2 \log n$	$35 \log p$	0.17
Demytko	$\log n$	$22 \log p$	0.14
New schemes	$2 \log n$	$3 \log p$	2.0

Nowadays, the RSA scheme with 512 bits modulus n (block size) is practically used for key distributions and digital signatures. In this standard RSA scheme, eight DES keys can be distributed in one block. In the new schemes with 1024 bits block size, 16 DES keys can be distributed at the same decryption speed and the same security level.

4.2 Encryption Efficiency of Scheme 1 and Scheme 2

Although the dominant computations involve the decryptions in scheme 1 and scheme 2, we evaluate their encryption efficiency to compare thses schemes. We focus on pure encryption procedures excluding isomorphic mapping procedures. Let $|e|$ be the bit-length of encryption key e. A possible minimum value of e is 3, and $|e| = 2$. It is clear that the encryption of scheme 2 requires $1.5|e|$ multiplications modulo n on average. In scheme 1, computing the multiples of a point on curve E_n can be performed in affine coordinates (2) or homogeneous coordinates. A point (x, y) on the affine plane is equivalent to a point (X, Y, Z) on the projective plane, where $x = X/Z$, $y = Y/Z$. When we put $b = 0$, the addition formula in affine coordinates can be rewritten in homogeneous coordinates as equations (7) and (8) in the Appendix. The revised formulae with minimum number of multiplications are equations (9) and (10) in the Appendix. In the addition formula in homogeneous coordinates, contrary to that in affine coordinates, the divisions in Z_n in each addition over E_n can be avoided. Each elementary addition over E_n is calculated using addition, subtraction, multiplication and division in Z_n. For simplicity, addition, subtraction and special multiplication by a small constant were neglected for the comparison. In affine coordinates, each non-doubling addition requires three multiplications and one division in Z_n, and each doubling requires six multiplications and one division in Z_n. In homogeneous coordinates, each non-doubling addition requires 26 multiplications in Z_n, and each doubling requires 26 multiplications in Z_n. Let ℓ be the ratio of the computation amount of division in Z_n to that of multiplication in Z_n. Consequently, the encryption of scheme 1 based on affine coordinates requires $(7.5 + 1.5\ell)|e|$ multiplications in Z_n on average. That based on homogeneous coordinates requires $39|e| + \ell$ multiplications in Z_n on average. Since $1.5|e| < (7.5 + 1.5\ell)|e|$ and $1.5|e| < 39|e| + \ell$, the encryption of scheme 2 is faster than that of scheme 1. In particular, encryption efficiency of scheme 1 differs by the implemented coordinates. For example, when $e = 3$ and $e = 21$, the encryption in homogeneous coordinates is faster than that in affine coordinates if and only if $\ell > 31.5$ and $\ell > 24.2$, respectively.

5 Security

5.1 Security in One-to-one Communication

We show a theorem about the security relationship between the proposed schemes and the RSA scheme.

Theorem 4 *Breaking each of the proposed schemes is computationally equivalent to breaking the RSA scheme. That is, the following sentences are equivalent.*

(i) There is an efficient algorithm $A1$ *such that for all* $c_x, c_y \in Z_n^*$, $(c_x, c_y) \in E_n(a,0)$, *if* $(c_x, c_y) = e \otimes (m_x, m_y)$ *over* $E_n(a,0)$, *then* $A1(c_x, c_y, e, n) = (m_x, m_y)$.

(ii) There is an efficient algorithm $A2$ *such that for all* $c, a \in Z_n^*$, $(m_x, m_y) \in E_n(a,0)$, *if* $c = \left(\frac{m_x^3}{m_y^2}\right)^e \bmod n$, *then* $A2(c, a, e, n) = (m_x, m_y)$.

(iii) There is an efficient algorithm B *such that for all* $c \in Z_n^*$, *if* $c = m^e \bmod n$, *then* $B(c, e, n) = m$.

Proof: First, the equivalence between (i) and (iii) is shown as follows.
(i) $\Rightarrow$ (iii)
Assuming algorithm $A1$ is given, algorithm B is defined as follows.
Input: c, e, n
Step 1: Choose $a \in Z_n^*$ randomly.
Step 2: Compute $(c_x, c_y) \in E_n(a,0)$ from c, a and n by using isomorphic mapping, without knowing factors of n as

$$c_x = \frac{a^2\,c}{(c-1)^2} \bmod n, \quad c_y = \frac{a^3\,c}{(c-1)^3} \bmod n.$$

Step 3: Compute $(m_x, m_y) = A1(c_x, c_y, e, n)$.
Step 4: Compute $m = 1 + \frac{am_x}{m_y} \bmod n$
Output: m
If algorithm $A1$ requires $O(T)$ bit-operations, then algorithm B requires $O(T + (\log n)^3)$ bit-operations, and is polynomially reducible from algorithm $A1$.

(iii) $\Rightarrow$ (i)
Assuming algorithm B is given, algorithm $A1$ is defined as follows.
Input: $(c_x, c_y), e, n$
Step 1: Compute $a = \frac{c_x^3 - c_y^2}{c_x c_y} \bmod n$.
Step 2: Compute $c = 1 + \frac{ac_x}{c_y}$.
Step 3: Compute $m = B(c, e, n)$.
Step 4: Compute $(m_x, m_y) \in E_n(a,0)$ from m, a and n by using isomorphic mapping, without knowing factors of n as

$$m_x = \frac{a^2\,m}{(m-1)^2} \bmod n, \quad m_y = \frac{a^3\,m}{(m-1)^3} \bmod n.$$

Output: (m_x, m_y)
If algorithm B requires $O(T)$ bit-operations, then algorithm $A1$ requires $O(T + (\log n)^3)$ bit-operations, and is polynomially reducible from algorithm B.

Next, the equivalence between (ii) and (iii) is shown as follows.
(ii) $\Rightarrow$ (iii)
Assuming algorithm $A2$ is given, algorithm B is defined as follows.
Input: c, e, n
Step 1: Choose $a \in Z_n^*$ randomly.
Step 2: Compute $(m_x, m_y) = A2(c, a, e, n)$.
Step 3: Compute $m = 1 + \frac{am_x}{m_y} \bmod n$
Output: m
If algorithm $A2$ requires $O(T)$ bit-operations, then algorithm B requires $O(T + (\log n)^3)$ bit-operations, and is polynomially reducible from algorithm $A2$.

(iii) $\Rightarrow$ (ii)
Assuming algorithm B is given, algorithm $A2$ is defined as follows.
Input: c, a, e, n
Step 1: Compute $m = B(c, e, n)$.
Step 2: Compute $(m_x, m_y) \in E_n(a, 0)$ from m, a and n, without knowing factors of n as

$$m_x = \frac{a^2 m}{(m-1)^2} \bmod n, \quad m_y = \frac{a^3 m}{(m-1)^3} \bmod n.$$

Output: (m_x, m_y)
If algorithm B requires $O(T)$ bit-operations, then algorithm $A2$ requires $O(T + (\log n)^3)$ bit-operations, and is polynomially reducible from algorithm B. ∎

The above theorem is concerning on usual passive attacks. Consider possibility of active known-plaintext attacks. Assume that an attacker knows a value of m_y in addition to the values of c_x, c_y, e and n. The attacker aims at obtaining m_x by solving cubic congruence $m_x^3 - m_y^2 \equiv am_x m_y \bmod n$ with known m_y and $a = \frac{c_x^3 - c_y^2}{c_x c_y} \bmod n$. However, it seems difficult to obtain m_x if breaking the RSA scheme is difficult. On the other hand, assume that an attacker knows a value of m_x in addition to the values of c_x, c_y, e and n. The attacker aims at obtaining m_y by solving quadratic congruence $m_x^3 - m_y^2 \equiv am_x m_y \bmod n$ with known m_x and $a = \frac{c_x^3 - c_y^2}{c_x c_y} \bmod n$. However, it seems difficult to obtain m_y if breaking the Rabin scheme [8] is difficult. Note that breaking the Rabin scheme (i.e., factoring n) is more difficult than the breaking the RSA scheme in a usual sense. Thus, additive information on m_x or m_y seems useless for cryptanalysis.

5.2 Security in Broadcast Applications

In broadcast applications, the original RSA scheme is not secure if encryption key e is small. Let e and n_i be public keys of the original RSA scheme for a receiver R_i $(1 \leq i \leq k)$. The common plaintext m is encrypted as $c_i = m^e \bmod n_i$ $(1 \leq i \leq k)$ for k receivers. If $k \geq e$, then the system of congruences $c_i \equiv m^e (\bmod n_i)$ $(1 \leq i \leq e)$ can be transformed into the equation $c = m^e$, where c is the combined ciphertext from c_i via the Chinese Remainder Theorem. Hence, the plaintext m can be computed as $m = c^{1/e}$ over the real field. Even if known terms like "user ID" are included in the plaintexts such that $m_i = \alpha_i m + \beta_i$, where α_i and β_i are publicly known, Hastad [2] showed that similar attacks aimed at obtaining m can be successful by solving a set of k congruences of polynomials $\sum_{j=0}^{u} t_{ij} m^j \equiv 0 \bmod n_i$. The inequality condition for a successful attack is given by

$$\prod_{i=1}^{k} n_i > n_s^{u(u+1)/2}(k+u+1)^{(k+u+1)/2} 2^{(k+u+1)^2/2}(u+1)^{u+1},$$

where n_s =min(n_i). This condition is the most sensitive to the degree u of the obtained set of congruences of polynomials. In the RSA scheme with the linearly related plaintexts $m_i = \alpha_i m + \beta_i$, the system of congruences in m with degree e can be obtained in broadcast applications. In the KMOV scheme over elliptic curves, the system of congruences in m_x with degree e^2 can be obtained in broadcast applications. Thus, it was shown in [6] that the KMOV scheme is more secure than the original RSA scheme against the Hastad attack.

We evaluate the security of the new schemes (i.e., scheme 1 and scheme 2) in broadcast applications, in which the plaintext is purely common or linearly related. First, consider scheme 1. There is a recursive formula for computing x_i such that $(x_i, y_i) = i \otimes (x_1, y_1)$ over $E_n(a, 0)$, where $(x_1, y_1) \in E_n(a, 0)$ is the initial point:

$$x_{2i} = \frac{x_i^2}{4x_i + a^2} \bmod n, \quad x_{2i+1} = \frac{x_{i+1}^2 x_i^2}{x_1(x_{i+1} - x_i)^2} \bmod n. \qquad (6)$$

Using Eq. (6), ciphertext c_x in scheme 1 is expressed by

$$c_x = \frac{m_x^e}{h_e(m_x)} \bmod n,$$

where m_x is a plaintext and $h_i(m_x)$ is recursively defined as

$$h_1(m_x) = 1,$$
$$h_{2i}(m_x) = 4m_x^i h_i(m_x) + a^2(h_i(m_x))^2 \bmod n \ (i \geq 1),$$
$$h_{2i+1}(m_x) = (h_{i+1}(m_x) - m_x h_i(m_x))^2 \bmod n \ (i \geq 1).$$

Since the degree of $h_i(m_x)$ is $i-1$, and m_x^i and $h_i(m_x)$ are relatively prime polynomials, the system of congruences in m_x with degree e can be obtained as $m_x^e - c_x h_e(m_x) \equiv 0 \pmod{n}$. Thus, it is shown that scheme 1 has the same security as the RSA scheme when linearly related plaintexts are encrypted in broadcast applications. It is also shown that scheme 2 has the same security as the RSA scheme when linearly related plaintexts are encrypted in broadcast applications. Note that the RSA scheme with a purely common plaintext generates a simpler monomial m^e than a set of polynomials with degree e. Thus, the new schemes are more secure than the RSA scheme when purely common plaintexts are encrypted in broadcast applications.

We show numerical examples. When modulus n_i is 512 bits long and $e = 5$, the Hastad attack is applicable if more than 16 ciphertexts are obtained for the new schemes and the RSA scheme with linearly related plaintexts. When modulus n_i is 512 bits long and $e = 19$, the Hastad attack is applicable if more than 282 ciphertexts are obtained for the new schemes and the RSA scheme with linearly related plaintexts. When modulus n_i is 512 bits long and $e \geq 21$, the Hastad attack is *not* applicable for the new schemes and the RSA scheme with linearly related plaintexts. Note that when modulus n_i is 512 bits long and $e \geq 5$, the Hastad attack is *not* applicable for the KMOV scheme.

6 Conclusion

We have proposed fast RSA-type schemes over $E_n(a, 0)$. For a $2 \log n$-bit long message, the decryption speed of the proposed schemes is about 2.0 times faster than that of the RSA scheme. We have proved that breaking the proposed scheme is equivalent to breaking the RSA scheme.

Acknowledgement

We thank Hidenori Kuwakado and Yukio Tsuruoka for their valuable discussions.

References

[1] N. Demytko: "A new elliptic curves based analogue of RSA", Advances in Cryptology - Eurocrypt'93, LNCS **765** pp. 40-49 (1992).

[2] J. Hastad: "On using RSA with low exponent in a public key network", Advances in Cryptology - Crypto'85, LNCS **218** pp. 403–408 (1985).

[3] D. Husemöller: "Elliptic Curves", Springer-Verlag (1987).

[4] K. Koyama, U. M. Maurer, T. Okamoto and S. A. Vanstone: "New public-key schemes based on elliptic curves over the ring Z_n", Advances in Cryptology - Crypto'91, LNCS **576** pp. 252-266 (1991).

[5] K. Koyama and Y. Tsuruoka: "A signed binary window method for fast computing over elliptic curves", Advances in Cryptology - Crypto'92, LNCS **740** pp. 345-357 (1992).

[6] H. Kuwakado and K. Koyama: "On the security of RSA-type cryptosystems over elliptic curves against the Hastad attack", IEE Electronics Letters, Vol.30, No.22, pp. 1843-1844 (1994).

[7] A. J. Menezes: "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers (1993).

[8] M. Rabin: "Digital signatures and public-key cryptosystems", MIT/LCS/TR-21, (1979).

[9] R. L. Rivest, A. Shamir and L. Adleman: "A method for obtaining digital signatures and public-key cryptosystems", Comm. of the ACM, **21**, 2, pp. 120–126 (1978).

Appendix: Addition Formula for Singular Cubic Curves

For singular cubic curves $y^2 + axy = x^3$, the addition: $(x_3, y_3) = (x_1, y_1) \oplus (x_2, y_2)$ is given by chord-and-tangent law. The addition formula in affine coordinates is shown in equation (2). The addition formula in homogeneous coordinates is as follows.

Non-doubling Addition Formula for $(X_1, Y_1, Z_1) \neq (X_2, Y_2, Z_2)$

$$\left\{\begin{array}{ll} X_3 = & X_2^4 Z_1^4 + 2X_1X_2^3Z_1^3Z_2 + aX_2^2Y_2Z_1^4Z_2 + X_2Y_2^2Z_1^4Z_2 \\ & -aX_2^2Y_1Z_1^3Z_2^2 - 2aX_1X_2Y_2Z_1^3Z_2^2 - 2X_2Y_1Y_2Z_1^3Z_2^2 \\ & -X_1Y_2^2Z_1^3Z_2^2 - 2X_1^3X_2Z_1Z_2^3 + 2aX_1X_2Y_1Z_1^2Z_2^3 \\ & +X_2Y_1^2Z_1^2Z_2^3 + aX_1^2Y_2Z_1^2Z_2^3 + 2X_1Y_1Y_2Z_1^2Z_2^3 \\ & +X_1^4Z_2^4 - aX_1^2Y_1Z_1Z_2^4 - X_1Y_1^2Z_1Z_2^4 \\ Y_3 = & X_2^3Y_2Z_1^4 - 2X_2^3Y1Z_1^3Z_2 - aX_2Y_2^2Z_1^4Z_2 - Y_2^3Z_1^4Z_2 \\ & +3X_1X_2^2Y_1Z_1^2Z_2^2 - 3X_1^2X_2Y_2Z_1^2Z_2^2 + 2aX_2Y_1Y_2Z_1^3Z_2^2 \\ & +aX_1Y_2^2Z_1^3Z_2^2 + 3Y_1Y_2^2Z_1^3Z_2^2 + 2X_1^3Y_2Z_1Z_2^3 \\ & -aX_2Y_1^2Z_1^2Z_2^3 - 2aX_1Y_1Y_2Z_1^2Z_2^3 - 3Y_1^2Y_2Z_1^2Z_2^3 \\ & -X_1^3Y_1Z_2^4 + aX_1Y_1^2Z_1Z_2^4 + Y_1^3Z_1Z_2^4 \\ Z_3 = & X_2^3Z_1^4Z_2 - 3X_1X_2^2Z_1^3Z_2^2 + 3X_1^2X_2Z_1^2Z_2^3 - X_1^3Z_1Z_2^4 \end{array}\right. \tag{7}$$

Doubling Formula for $(X_1, Y_1, Z_1) = (X_2, Y_2, Z_2)$

$$\left\{\begin{array}{ll} X_3 = & 9aX_1^5Z_1 + 18X_1^4Y_1Z_1 + a^3X_1^4Z_1^2 - 6a^2X_1^3Y_1Z_1^2 \\ & -24aX_1^2Y_1^2Z_1^2 - 16X_1Y_1^3Z_1^2 - a^4X_1^2Y_1Z_1^3 \\ & -3a^3X_1Y_1^2Z_1^3 - 2a^2Y_1^3Z_1^3 \\ Y_3 = & -27X_1^6 + 45aX_1^4Y_1Z_1 + 36X_1^3Y_1^2Z_1 + 2a^3X_1^3Y_1Z_1^2 \\ & -15a^2X_1^2Y_1^2Z_1^2 - 24aX_1Y_1^3Z_1^2 - 8Y_1^4Z_1^2 \\ & -a^4X_1Y_1^2Z_1^3 - a^3Y_1^3Z_1^3 \\ Z_3 = & a^3X_1^3Z_1^3 + 6a^2X_1^2Y_1Z_1^3 + 12aX_1Y_1^2Z_1^3 + 8Y_1^3Z_1^3 \end{array}\right. \tag{8}$$

By introducing moderate intermediate variables, addition formulae (7) and (8) can be revised to minimize the number of multiplications:

Revised Non-doubling Addition Formula

$$\left\{\begin{array}{ll} X_3 = & H\{Z_1Z_2(T + X_1X_2K) - M - Q\}, \\ Y_3 = & L(M - Q) - Z_1Z_2\{GT + 3H(X_2^2Y_1Z_1 + X_1^2Y_2Z_2)\}, \\ Z_3 = & Z_1Z_2H^3, \end{array}\right. \tag{9}$$

where $H = X_2Z_1 - X_1Z_2$, $G = Y_2Z_1 - Y_1Z_2$, $K = X_2Z_1 + X_1Z_2$, $L = Y_2Z_1 + Y_1Z_2$, $M = X_2^3Z_1^3$, $Q = X_1^3Z_2^3$, $T = G(aH + G)$.

Revised Doubling Formula

$$\left\{\begin{array}{ll} X_3 = & Z_1A[X_1\{9V + Z_1(a^2I - 8Y_1J)\} - a^2C^2], \\ Y_3 = & -27V^2 + C[9VD + Z_1\{B^3 + a(X_1EF - a^2CJ)\}], \\ Z_3 = & Z_1^3A^3, \end{array}\right. \tag{10}$$

where $A = aX_1 + 2Y_1$, $B = aX_1 - 2Y_1$, $C = Y_1Z_1$, $D = 5aX_1 + 4Y_1$, $E = aX_1 - 12Y_1$, $F = aX_1 + 3Y_1$, $I = X_1^2 - aC$, $J = aX_1 + Y_1$, $V = X_1^3$.