Equivalence of Counting the Number of Points on Elliptic Curve over the Ring Z_n and Factoring n

Noboru Kunihiro and Kenji Koyama

NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
E-mail: {kunihiro, koyama}@cslab.kecl.ntt.co.jp

Abstract. For composite n, we prove that counting the number of points on elliptic curves over the ring Z_n is randomly computationally equivalent to factoring n. That is, we prove that if we can count it, we can easily factor n. Furthermore, we also prove that if we can solve the elliptic curve discrete logarithm problem modulo n, we can easily factor n.

1 Introduction

Elliptic curves can be applied to public-key cryptosystems, and as such several schemes have been proposed [3, 4, 5, 6, 9, 11]. There are two typical elliptic curve cryptosystems: ElGamal-type scheme [4, 11] and RSA-type schemes [3, 5, 6]. The security of the ElGamal-type elliptic curve cryptosystem is based on the difficulty of solving a discrete logarithm over elliptic curve modulo a prime. However, the security of an RSA-type elliptic curve cryptosystem is based on the difficulty of factoring a large composite. It has been conjectured that completely breaking the original RSA is computationally equivalent to the factoring the used composite, although this has NOT been proved yet. In a certain RSA-type elliptic curve (or cubic curve) cryptosystem proposed in [6], however, this equivalence between the two problems was proved. In general RSA-type elliptic curve cryptosystems, including RSA-type cubic curve cryptosystems, the equivalence has not been proved. As the order $\phi(n)$ of Z_n^* for a composite n have played a significant role in analyzing the security of the original RSA scheme, it is important to evaluate the complexity of counting the number of points on an elliptic curve over the ring Z_n for RSA-type elliptic curve cryptosystems.

We are interested in reductions of factoring to other problems in elliptic curve theory over Z_n. In this paper, we will consider the following problems.

FCT(n) : Given composite n, find the complete prime factorization of n.

COMP$(\phi(n))$: Given composite n, compute the Euler phi function $\phi(n) = |Z_n^*|$, which is the number of integers in the interval $[1, n]$, each of which are relatively prime to n.

COMP$(\#E_n(a, b))$: Given composite n and integers a and b, compute $\#E_n(a, b)$, which is the number of points over an elliptic curve $E_n : y^2 \equiv x^3 + ax + b \pmod{n}$.

COMP($\#E_n(a,b)$ mod d) : Given composite n, an elliptic curve $E_n(a,b)$ and prime d (= O($\log n$)), compute $\#E_n(a,b)$ mod d.

EDLP mod p (Elliptic curve Discrete Logarithm Problem mod p): Given *a prime* p, an elliptic curve E_p and two points G and A over E_p, find the positive integer α such that $\alpha G = A$.

EDLP mod n (Elliptic curve Discrete Logarithm Problem mod n): Given *a composite* n, an elliptic curve E_n and two points G and A over E_n, find the positive integer α such that $\alpha G = A$.

We would like to emphasize that for a prime p (not a composite), COMP($\#E_p(a,b)$) and COMP($\#E_p(a,b)$ mod d) are computable in polynomial time [13]. Conversely, it is not known whether an algorithm exists for solving an EDLP mod p in polynomial time (or even in sub-exponential time).

The purpose of this paper is to show possible reductions between the factoring problem and some problems in elliptic curve theory over Z_n. We prove the equivalence of FCT(n) and COMP($\#E_n(a,b)$) in Sect. 3 and the equivalence of FCT(n) and COMP($\#E_n(a,b)$ mod d) in Sect. 4. Finally, we prove that FCT(n) is randomly polynomial time reducible to EDLP mod n in Sect. 5.

2 Preliminaries

First, we define the notation used in the computational relationships and then describe some of the previous work relating to the original RSA–scheme. Following this, we briefly describe elliptic curves over Z_n and explain the elliptic curve factoring method [7, 14].

2.1 Notations of Computational Relationship

Let A, B and C be computational problems. We can define computational relationship among these problems in the following terms: $A \leq_P B$, $A \leq_{RP} B$, $A =_P B$, $A =_{RP} B$ and $A \leq_P B \oplus C$.

Definition 1 We say that problem A is *polynomial time reducible* to problem B, written as $A \leq_P B$, if there is an algorithm that solves A which uses *an oracle* (or a subroutine) for B, and the algorithm runs in polynomial time.

Definition 2 We say that problem A is *randomly polynomial time reducible* to problem B, written as $A \leq_{RP} B$, if there is an algorithm that solves A which uses *an oracle* (or a subroutine) for B, and the algorithm runs in randomly polynomial time.

Roughly speaking, $A \leq_P B$ or $A \leq_{RP} B$ implies that if solving B is easy, then solving A is also easy.

Definition 3 If $A \leq_P B$ and $B \leq_P A$, then we say that A and B are *computationally equivalent*, written as $A =_P B$.

Definition 4 If $\{A \leq_P B$ and $B \leq_{RP} A\}$, or $\{A \leq_{RP} B$ and $B \leq_P A\}$, or $\{A \leq_{RP} B$ and $B \leq_{RP} A\}$, then we say that A and B are *randomly computationally equivalent*, written as $A =_{RP} B$.

Definition 5 We say that A is *polynomial time reducible* to B and C, written as $A \leq_P B \oplus C$, if there is an algorithm that solves A which uses both *an oracle* for B and *an oracle* for C, and the algorithm runs in polynomial time.

2.2 Previous Work

For COMP($\phi(n)$) and FCT(n), the following two facts are widely known.

Fact 1 Let n be a composite that is a product of distinct odd primes. On the assumption that *Extended Riemann Hypothesis* (ERH) is true, it holds that COMP($\phi(n)$) $=_P$ FCT(n) [10]. Without the ERH assumption, it holds that COMP($\phi(n)$) $\leq_P$ FCT(n) and FCT(n) $\leq_{RP}$ COMP($\phi(n)$), i.e. FCT(n) $=_{RP}$ COMP($\phi(n)$) [8].

Fact 2 For composite n' that is a product of two distinct odd primes p and q, it holds that COMP($\phi(n')$) $=_P$ FCT(n').

Proof: Using an oracle for FCT(n'), we can obtain p and q. Therefore, we can obtain $\phi(n')$ by computing $(p-1)(q-1)$. Hence, it follows that COMP($\phi(n')$) $\leq_P$ FCT(n'). Conversely, using an oracle for COMP($\phi(n')$), we can obtain $p+q$ by using the fact that $p+q = n'+1-\phi(n')$. Thus, p and q can be determined by solving the quadratic equation: $x^2-(n'+1-\phi(n'))x+n' = 0$ over a real field R. This leads to $x = \frac{n'+1-\phi(n')\pm\sqrt{(n'+1-\phi(n'))^2-4n'}}{2}$, from which it follows that FCT(n') $\leq_P$ COMP($\phi(n')$). Thus, COMP($\phi(n')$) and FCT(n') are computationally equivalent, COMP($\phi(n')$) $=_P$ FCT(n'). □

Woll [15] studied the reductions between numerous number theoretic problems, including FCT(n) and COMP($\phi(n)$).

2.3 Elliptic Curve Modulo Composite n [9]

For simplicity, we assume that n is a composite that is a product of two distinct odd (unknown) primes p and q. The equation of an elliptic curve E is given as $E: y^2 = x^3+ax+b$. Let E_n, E_p and E_q be the elliptic curve E over Z_n, F_p and F_q, respectively. A point P over E_p is expressed by x and y coordinates modulo p as $P = (x, y)$ including infinity point $\mathcal{O}_p$. This infinity point is a zero element of group E_p, which we refer to as the zero point in this paper. The zero point over E_q is similarly denoted as $\mathcal{O}_q$. The set of all points over $E_p(, E_q)$ and the point $\mathcal{O}_p, (\mathcal{O}_q)$ forms an Abelian group under a certain addition, *tangent-and-chord* operation.

A group E_n (i.e. a set of points on E_n) can be defined as the direct sum of two groups: $E_n = E_p \oplus E_q$. Hence, every element P over E_n can be represented by the pair $P_1 \in E_p$ and $P_2 \in E_q$, which we denote as $P = \langle P_1, P_2\rangle$. In this group

E_n, the point $\langle O_p, O_q \rangle$ is the zero point O of E_n. The points $\langle O_p, P_2(\neq O_q)\rangle$ and $\langle P_1(\neq O_p), O_q\rangle$ are semi-zero points. Finally, ordinary points are any point other than the zero point or semi-zero points. The number of elements of E_n is given by $\#E_n = \#E_p \cdot \#E_q$. The group E_n consists of $((\#E_p - 1)(\#E_q - 1))$ ordinary points and $(\#E_p + \#E_q - 2)$ semi-zero points and *one* zero point O.

The case of general composites $n = \prod_{i=1}^{k} p_i$ should also be mentioned. A group E_n can be defined as a direct sum of k groups: $E_n = \oplus_{i=1}^{k} E_{p_i}$. Let O_{p_i} be the zero point of E_{p_i}. Hence, every element P over E_n can be represented by k-tuples of $P_1 \in E_{p_1}$, $P_2 \in E_{p_2} \cdots$ and $P_k \in E_{p_k}$, letting us denote $P = \langle P_1, P_2, \cdots, P_k\rangle$. Note that the point with all zero point in k-tuples is zero point O and the point with at least 1 zero point in k-tuples is a semi-zero point, except zero point of E_n.

2.4 Elliptic Curve Factoring Method [7]

Suppose that $P \in E_n$ and MP is a M times point of P as $MP = \underbrace{P + P + \cdots P}_{M \text{ times}} \equiv \langle P_1', P_2' \rangle$. We compute MP using the *tangent-and-chord* operation modulo n without knowing the prime factors p and q. Suppose that MP is calculated successively in a binary method as $M_1P(= P), M_2P, \ldots, M_lP(= MP)$. If MP is a semi-zero point as $MP = \langle O_p, MP_2(\neq O_q)\rangle$, then at least one M_iP $(1 \leq i \leq l)$ is a semi-zero point. In this case, in the process to compute MP, we cannot obtain M_iP using the tangent-and-chord operation modulo n, which includes a calculation of an multiplicative inverse of the number not prime to n. However, we can find a prime factor p of n. Hence, if $P_1' = O_p$ and $P_2' \neq O_q$, we can find a prime factor p.

Denoting the order of point P over E_p by $\Phi_p(P)$, we can rewrite the above condition as follows. That is, if $\Phi_p(P) | M$ and $\Phi_q(P) \nmid M$, we can find a prime factor p.

3 FCT(n) and COMP($\#E_n(a, b)$)

In this section, we examine FCT(n) and COMP($\#E_n(a, b)$). We shall prove Theorem 1 (described below). The following is well known.

Fact 3 For composite n that is a product of distinct odd primes, it holds that COMP($\#E_n(a, b)$) $\leq_P$ FCT(n).

Proof: Using an oracle for FCT(n), we can obtain prime factorization of n as $n = p_1 p_2 \cdots p_k$. We can compute $\#E_{p_i}$ for each p_i by Schoof algorithm [13] and obtain $\#E_n = \prod_{i=1}^{k} \#E_{p_i}$. Since k is less than $\log_2 n$, we can obtain $\#E_n$ in polynomial time. □

We will now prove the converse of Fact 3.

Theorem 1 For composite n that is a product of district odd primes, it holds that FCT(n) $\leq_{RP}$ COMP($\#E_n(a, b)$) and FCT(n) $=_{RP}$ COMP($\#E_n(a, b)$).

Proof: The following algorithm can factorize n in randomly polynomial time using an oracle for COMP($\#E_n(a,b)$).

```
Factoring Algorithm using an oracle for COMP($\#E_n(a,b)$)
Input: Composite $n(=\prod_{i=1}^{k} p_i)$
Output: Prime factors $p_1, p_2, \cdots, p_k$
Step 1 Set a parameter $S$ and set elliptic curve $E_n(a,b) : y^2 \equiv x^3 + ax + b \pmod{n}$ satisfying GCD$(n, 4a^3 + 27b^2) = 1$ and a point $P$ over $E_n$.
    1.1 Set $S$ that is the largest prime less than $\lfloor \log n \rfloor$.
    1.2 Set the point $P = (x_0, y_0)$ randomly and set $a$ randomly.
    1.3 Calculate $b = {y_0}^2 - {x_0}^3 - ax_0$.
Step 2 Using an oracle for COMP($\#E_n(a,b)$), obtain $\#E_n(a,b)$.
Step 3 Check the divisibility of $\#E_n(a,b)$ by $S$. If $S | \#E_n$ and $S^2 \not| \#E_n$, proceed to step 4. Otherwise, return to step 1.2.
Step 4 Set $M \equiv \frac{\#E_n(a,b)}{S}$ and try to compute $MP$ over $E_n(a,b)$. Suppose that $MP$ is calculated as $M_1P(=P), M_2P, \ldots, M_lP(=MP)$.
    - If at least one $M_iP$ is a *semi-zero point*, then we can find a prime factor $p_i$. When $n/p_i$ is a prime, factoring is completed. When $n/p_i$ is a composite, set $n = n/p_i$, return to step 1.
    - If $MP$ is a *zero point*, then we cannot find a prime factor. Return to step 1.2.
```

The following analysis leads us to the conclusion that the above algorithm runs in randomly polynomial time $O((\log n)^5)$.

We use properties of *elliptic curve* and *elliptic curve factoring method* [7] described in Sect. 2. Let $\Phi_{p_i}(P)$ be the order of point P over E_{p_i}. In the elliptic curve method, the order of points plays an important role. Note that, in the above algorithm, we only assumed an oracle to compute the number of points: $\#E_n$, instead of an oracle to compute the order of a point.

First, we will prove that the point MP in step 4 always becomes a zero point or a semi-zero point. In general, mP is a zero point if $\Phi_{p_i}(P)|m$ for all i, and mP is a semi-zero point if $\Phi_{p_i}(P)|m$ for at least one i and mP is not a zero point. Note that $\Phi_{p_i}(P)|\#E_{p_i}$ and $\#E_{p_i}|\#E_n$ and $\Phi_{p_i}(P)|\#E_n$ for all i and all P over E_n. Let p_1 be the prime factor that satisfies $S|\#E_{p_i}$. Since $S|\#E_n$ and $S^2 \not| \#E_n$, such p_1 uniquely exists. Since $S|\#E_{p_1}$ and $S \not| \#E_{p_i}$ for $2 \le i \le k$, M is denoted by $M = \frac{\#E_{p_1}}{S} \prod_{i=2}^{k} \#E_{p_i}$ and then $\#E_{p_i}|M$ for $2 \le i \le k$. Note that $\#E_{p_1} \not| M$ since $\#E_{p_1}$ is a multiple of S and M is not a multiple of S. Letting $P = \langle P_1, P_2, \ldots, P_k \rangle$, MP is denoted as $MP = \langle MP_1, MP_2, \ldots, MP_k \rangle$. Since $\Phi_{p_i}(P)|\#E_{p_i}$, then $\Phi_{p_i}(P)|M$ and $MP_i = \mathcal{O}_{p_i}$ for $2 \le i \le k$ and all P over E_n. Hence MP is a zero point or a semi-zero point because $\Phi_{p_i}(P)|M$ for at least one i.

This classification of the above two cases depends on whether MP_1 is a zero point $\mathcal{O}_{p_1}$ or an ordinary point. This dependency corresponds to the divisibility of $\Phi_{p_1}(P)$ by S as follows. If $S|\Phi_{p_1}(P)$, we can write $M = \frac{\Phi_{p_1}(P)}{S} \cdot C$, where C is not a multiple of S. Since M is not a multiple of $\Phi_{p_1}(P)$, MP_1 is not a zero point. Hence, MP is a semi-zero point if $S|\Phi_{p_1}(P)$. If $S\nmid\Phi_{p_1}(P)$, we can write $M=\Phi_{p_1}(P)\cdot C$, where C is not a multiple of S. Since M is a multiple of $\Phi_{p_1}(P)$, MP_1 is a zero point $\mathcal{O}_{p_1}$. Hence, MP is a zero point if $S\nmid\Phi_{p_1}(P)$.

Next, we will evaluate the probability of passing step 3. If R is a uniformly distributed random number, then the probability that $S^j|R$ is $\frac{1}{S^j}$, where j is a small integer and S is a prime. For prime p and randomly chosen integers a and b, the value of $\#E_p(a,b)$ behaves as a pseudo random number. Strictly speaking, $\#E_p(a,b)$ is not uniformly distributed in the range $p-2\sqrt{p}+1\leq\#E_p(a,b)\leq p+2\sqrt{p}+1$. However, we put the assumption: $\Pr\{S^j|\#E_p(a,b)\}=\frac{1}{S^j}$. The probability that $S|\#E_n$ and $S^2\nmid\#E_n$ is equal to the probability that there exists p_i such that $S|\#E_{p_i}$ and $S^2\nmid\#E_{p_i}$ and $S\nmid\#E_{p_j}$ for all $j(\neq i)$. Since $\Pr\{S|\#E_{p_i}$ and $S^2\nmid\#E_{p_i}\}$ is equal to $\frac{1}{S}(1-\frac{1}{S})$ and $\Pr\{S\nmid\#E_{p_j}\}$ is equal to $(1-\frac{1}{S})$ for each j on the above assumption, $\Pr\{S|\#E_n$ and $S^2\nmid\#E_n\}=\frac{1}{S}(1-\frac{1}{S})\cdot(1-\frac{1}{S})^{k-1}\cdot k=(1-\frac{1}{S})^k\frac{k}{S}$.

Next, we will evaluate the probability of finding a prime factor in step 4. From the previous analysis, this probability is equal to the probability that a random point P over E_{p_1} satisfies that $S|\Phi_{p_1}(P)$. This probability is given as $1-\frac{1}{S}$.

Hence, the probability $Q(k,S)$ of finding a prime factor per curve is given by

$$Q(k,S)\equiv(1-\frac{1}{S})^{k+1}\frac{k}{S}. \tag{1}$$

Note that the average number of trial curves is given by $1/Q(k,S)$.

By theoretical analysis, we find that $Q(k,S)$ is monotonically increasing in $k\leq S$, for a given S. From this property and that $k\leq\log n\approx S$, an integer which minimizes $Q(k,S)$ is $k=2$. In this case the probability $Q(k,\log n)\geq Q(2,\log n)=(1-\frac{1}{\log n})^3\frac{2}{\log n}=\mathrm{O}(\frac{1}{\log n})$. Hence, the average number of trial curves needed to find one prime factor is less than $\mathrm{O}(\log n)$. Since the number of prime factors is less than $\log n$, the average number of total trial curves to find the complete prime factors is $\mathrm{O}((\log n)^2)$.

Next, we will determine the computation amount per curve. Computing MP needs $\mathrm{O}(\log M)$ group operations. Since $M\leq\#E_n\approx n$, the number of group operations is $\mathrm{O}(\log n)$. It is known that the computation amount per group operation is $\mathrm{O}((\log n)^2)$. These results lead to the conclusion that the above algorithm runs in randomly polynomial time $\mathrm{O}((\log n)^5)$. Thus, $\mathrm{FCT}(n)\leq_{RP}\mathrm{COMP}(\#E_n(a,b))$ has been proven.

This property and Fact 3 ($\mathrm{COMP}(\#E_n(a,b))\leq_P\mathrm{FCT}(n)$) imply
$\mathrm{FCT}(n)=_{RP}\mathrm{COMP}(\#E_n(a,b))$. □

We show a simple example.

Example: Let $n=22657$. When we set $S=5$, $a=22651$ and $b=9310$, we have $\#E_n=28688$ and $5\nmid\#E_n$. This case fails at step 3. When we set $S=5$, $a=20837$ and $b=8047$, we have $\#E_n=26400$ and $5^2|\#E_n$. This case also fails at step 3. Next, we show an example of success. In step 1, we set $S=5$ and $a=18405, b=18024$ and $P=(3926,16206)$. In step

2, we obtain $\#E_n(a,b) = 22080$ by an oracle. In step 3, we confirm that $5|\#E_n$ and $5^2 \not| \#E_n$. In step 4, since $M \equiv \frac{\#E_n}{5} = 4416$, we compute MP over $E_{22657}(18405, 18024)$ by using a binary method, which is calculated successively as $P, 2P, 4P, 8P, 16P, 17P, 34P, 68P, 69P = (15499, 8896)$. However, $138P$ cannot be calculated by tangent and chord operation because $138P$ is a *semi-zero* point (note that 8896 is not relatively prime to 22657). Since $\mathrm{GCD}(8896, n(= 22657)) = 139$, factoring is thus completed.

As a result $n = 22657$ is factored by $n = 22657 = pq = 139 \times 163$. Note that $\Phi_p(P) = 138, \Phi_q(P) = 10$ and $M = 4416$ imply $\Phi_p(P)|M$ and $\Phi_q(P)\not| M$, and that factoring is successful (refer to Sect. 2.4). The reason why factoring is done in computing $138P$ is that "$138P$" occurs while in the process of computing $4416P$ and "138" happens to satisfy $\Phi_p(P)|138$ and $\Phi_q(P)\not|138$. We may note that $\#E_n = \#E_p \times \#E_q = 138 \times 160 = 2^6 \times 3 \times 5 \times 23$.

Remarks:

(1) It may be interesting to compare Fact 1 and Theorem 1. If ERH is true, then $\mathrm{FCT}(n) \leq_P \mathrm{COMP}(\phi(n))$ and $\mathrm{FCT}(n) \leq_{RP} \mathrm{COMP}(\#E_n(a,b))$. Without the assumption that ERH is true, $\mathrm{FCT}(n) \leq_{RP} \mathrm{COMP}(\phi(n))$ and $\mathrm{FCT}(n) \leq_{RP} \mathrm{COMP}(\#E_n(a,b))$.

(2) We check whether $S|\#E_n$ and $S^2 \not| \#E_n$ in step 3 of the factoring algorithm in the proof of Theorem 1. We can construct an algorithm to rule out the condition: $S^2 \not| \#E_n$ in step 3. Since the success probability of reconstructed algorithm is larger than that of the previous algorithm, this algorithm runs in randomly polynomial time. Note that the factoring algorithm in the proof of Theorem 1, in which analysis of success probability is easier, also runs in randomly polynomial time.

(3) In the factoring algorithm in the proof of Theorem 1, we set one S and use *several* trial curves to factorize. Moreover, we can construct a dual algorithm with *one* curve by using *several* primes S, ranging from 2 to B. We define "partially B–smooth" number as the integer whose *smallest*[1] prime factor is less than B. Step 3 and step 4 of the dual algorithm are as follows.

Step 3 If $\#E_n(a,b)$ is *partially B-smooth*, find prime factors less than B of $\#E_n$ by trial division. Otherwise, return to step 1.2. Let's denote $\#E_n = {q_1}^{e_1}{q_2}^{e_2}\cdots{q_s}^{e_s} \times R$, where $q_1, q_2, \cdots, q_s$ are all primes less than B with $e_i \geq 0$ and R is not partially B–smooth.

Step 4 Let $\widetilde{M_{i,j}} = \#E_n(a,b)/{q_i}^j$ for $1 \leq i \leq s$ and $1 \leq j \leq e_i$. Compute $\widetilde{M_{i,j}}P$. If there exists $\widetilde{M_{i,j}}P$ which is a semi-zero point, factoring is successful. Otherwise (i.e. all $\widetilde{M_{i,j}}P$ are zero points), factoring fails.

We can approximately estimate the probability of success. The probability T that $\#E_n(a,b)$ becomes partially B–smooth is denoted by $T = 1 - \prod_{i=1}^{s}(1 - \frac{1}{q_i})$ if $B \ll \#E_n$. From Mertens's Theorem [12], T is approximated as $1 - \frac{e^{-\gamma}}{\log B}$, where γ is Euler's constant. If B is sufficiently large, we have $T \approx 1$. The probability that there exists q_i with $e_i = 1$ $(1 \leq i \leq s)$ is almost 1. Let q^* be the largest

[1] B–smooth number is an integer whose *biggest* prime factor is less than B.

of the primes. The probability U to succeed in factoring in step 4 is more than $1 - \frac{1}{q^*}$. When B is sufficiently large, $U \approx 1$. Hence, the number of expected trial curves is almost 1. Let $|\widetilde{M_{i,j}}|$ be the number of combination of suffix of $\widetilde{M_{i,j}}$, where $|\widetilde{M_{i,j}}| = \sum_{i=1}^{s} e_i$. Since the average of e_i is given as $\frac{1}{q_i - 1}$, the average of $|\widetilde{M_{i,j}}|$, denoted by $\overline{|\widetilde{M_{i,j}}|}$, is given by the $\overline{|\widetilde{M_{i,j}}|} = \sum_{i=1}^{s} \frac{1}{q_i - 1} \approx \log\log B$ [12]. From $B \ll n$, we have $\overline{|\widetilde{M_{i,j}}|} < \log\log n$. Hence, the computing all $\widetilde{M_{i,j}}P$ is completed in polynomial time. Since $T \approx 1$ and $U \approx 1$, this reconstructed dual algorithm also runs in randomly polynomial time.

In several variants of RSA-type cryptosystems, the multiple of $\phi(n)$, instead of $\phi(n)$ itself, can be easily known from public information. These schemes are insecure because Miller [10] proved that if the multiple of $\phi(n)$ is known, n can be easily factored. In relation to this fact, we prove the following.

Lemma 1 Let a value $r\#E_n(a,b)$ be the multiple of $\#E_n(a,b)$, where r is randomly distributed and independent of n, a and b. For composite n that is a product of odd primes, it holds that $\mathrm{FCT}(n) \leq_{RP} \mathrm{COMP}(r\#E_n(a,b))$.

Proof: We can prove this lemma by revising the previous "Factoring Algorithm". This is achieved by replacing an oracle for $\mathrm{COMP}(\#E_n(a,b))$ with an oracle for $\mathrm{COMP}(r\#E_n(a,b))$ in step 2 of the proof of Theorem 1. Note that the above assumption about r implies $\Pr\{S^j|r\} = \frac{1}{S^j}$. From this property and $\Pr\{S^j|\#E_{p_i}\} = \frac{1}{S^j}$ for each i and j from the assumption described above, the probability that $S|r\#E_n$ and $S^2 \not| r\#E_n$ is given as $(1 - \frac{1}{S})^{k+1}\frac{k+1}{S}$. The probability of succeeding of finding a prime factor in step 4 is $1 - \frac{1}{S}$ as well as $\mathrm{COMP}(\#E_n(a,b))$. Hence, the probability of successful factoring is $(1 - \frac{1}{S})^{k+2}\frac{k+1}{S}$. As well as the analysis of $\mathrm{COMP}(\#E_n(a,b))$, this revised algorithm finds complete prime factors in randomly polynomial time $O((\log n)^5)$. □

4 FCT(n) and COMP($\#E_n(a,b)$ mod d)

In this section, we present reductions between two problems, $\mathrm{FCT}(n)$ and $\mathrm{COMP}(\#E_n(a,b) \bmod d)$.

The computation amount of the problem $\mathrm{COMP}(\#E_p(a,b) \bmod d)$ is known to be equal $O((\log p)^3 d^5)$, where p is a prime (not a composite)[13]. Hence, if $d = O(\log p)$, then $\mathrm{COMP}(\#E_p(a,b) \bmod d)$ is computable in polynomial time. The Schoof algorithm[2] for calculating $\#E_p(a,b)$ runs in polynomial time $O((\log p)^8)$ by using $O(\log p)$ times of $\mathrm{COMP}(\#E_p(a,b) \bmod d)$. However, Theorem 2 (described below) states that $\mathrm{COMP}(\#E_n(a,b) \bmod d)$ is not as easy as $\mathrm{FCT}(n)$. Consequently, before proving Theorem 2, we must establish the following lemma.

[2] Charlap, Coley and Robbins [2] showed an algorithm which calculates $\#E_p(a,b)$ in $O((\log p)^6)$ by revising Schoof algorithm.

Lemma 2 For composite n that is a product of distinct odd primes, it holds that $\mathrm{COMP}(\#E_n(a,b)) \le_P \mathrm{COMP}(\#E_n(a,b) \bmod d)$.

Proof: Using the following algorithm, $\#E_n(a,b)$ is computable in polynomial time.

Counting the number of points ($\#E_n(a,b)$) Algorithm using an oracle for $\mathrm{COMP}(\#E_n(a,b) \bmod d)$

Input: Composite $n(=\prod_{i=1}^{k} p_i)$, elliptic curve E_n: $y^2 \equiv x^3 + ax + b \pmod n$

Output: $\#E_n(a,b)$

Step 1 Choose a number L such that $(n+1)^2 \le \prod l$, where the product ranges over all primes l between 3 and L.

Step 2 Compute $r_l = \#E_n(a,b) \bmod l$ by using an oracle for $\mathrm{COMP}(\#E_n(a,b) \bmod l)$ for $l = 3, 5, 7, 11, \cdots, L$, respectively.

Step 3 Solve the system of the congruences: $\#E_n \equiv r_l \pmod l$ by the Chinese Remainder Theorem, where l is all the primes between 3 and L.

The following analysis leads to the conclusion that this algorithm runs in polynomial time $\mathrm{O}((\log n)^2)$.

Since $L \approx \log(n+1)^2$, we have $L = \mathrm{O}(\log n)$. Therefore, obtaining each r_l is executable by an oracle for $\mathrm{COMP}(\#E_n(a,b) \bmod l)$. Since step 2 consists of at most L iterations to obtain r_l, step 2 is completed in polynomial time. Since step 3 is computable in polynomial time $\mathrm{O}((\log n)^2)$, it follows from the above discussion that this algorithm completes in polynomial time to compute $\#E_n(a,b)$. □

Theorem 2 For composite n that is a product of distinct odd primes, it holds that $\mathrm{FCT}(n) \le_{RP} \mathrm{COMP}(\#E_n(a,b) \bmod d)$. In addition, it holds that $\mathrm{FCT}(n) =_{RP} \mathrm{COMP}(\#E_n(a,b) \bmod d)$.

Proof: We have the former part of Theorem 2 from Theorem 1 and Lemma 2 because $\mathrm{FCT}(n) \le_{RP} \mathrm{COMP}(\#E_n(a,b)) \le_P \mathrm{COMP}(\#E_n(a,b) \bmod d)$. In addition, $\mathrm{COMP}(\#E_n(a,b) \bmod d) \le_P \mathrm{FCT}(n)$ (this is trivially proven) and $\mathrm{FCT}(n) \le_{RP} \mathrm{COMP}(\#E_n(a,b) \bmod d)$ imply the latter part of Theorem 2, $\mathrm{FCT}(n) =_{RP} \mathrm{COMP}(\#E_n(a,b) \bmod d)$. □

5 FCT(n) and EDLP mod n

In this section, we will present the reductions between $\mathrm{FCT}(n)$ and the elliptic curve discrete logarithm problem modulo n, EDLP $\bmod n$. EDLP $\bmod n$ is analogously related with DLPmod n, the ordinary discrete logarithm problem modulo n. Letting DLP $\bmod p$ be an ordinary discrete logarithm problem modulo prime p, it is known that $\mathrm{FCT}(n) \le_{RP}$ DLP $\bmod n$ and DLP $\bmod n \le_P \mathrm{FCT}(n) \oplus$ DLP $\bmod p$ [1]. Similar reductions also hold in $\mathrm{FCT}(n)$ and EDLP mod n.

Theorem 3 For composite n that is a product of distinct odd primes, it holds that $\mathrm{FCT}(n) \leq_{RP}$ EDLP mod n.

Proof: We will present a factoring algorithm using an oracle for EDLP mod n. This algorithm is constructed by revising the factoring algorithm described in the proof of Theorem 1.

```
Factoring Algorithm using an oracle for EDLP mod $n$
Input: Composite $n(= \prod_{i=1}^{k} p_i)$
Output: $k$ prime factors $p_1, p_2, \dots, p_k$
Step 1 Set a parameter $S$ and set elliptic curve $E_n(a,b) : y^2 \equiv x^3 + ax + b \pmod{n}$ and a point $P$ over $E_n$.
Step 2 Compute the smallest positive integer $M'$ satisfying $M'P = \mathcal{O}$ using an oracle for EDLP mod $n$.
Step 3 If $S|M'$ and $S^2 \not| M'$, proceed to the next step. Otherwise, return to step 1.
Step 4 Set $M \equiv \frac{M'}{S}$ and compute $MP$.
    – If $MP$ is an ordinary point, then we fail to find a prime factor. Return to step 1.
    – Otherwise, since $MP$ is a semi-zero point, then we can find a prime factor $p_i$. When $n/p_i$ is a prime, factoring is completed. When $n/p_i$ is a composite, set $n = n/p_i$ and return to step 1.
```

The following analysis leads to the conclusion that this algorithm runs in randomly polynomial time $\mathrm{O}((\log n)^5)$.

Note that $M^{'}$ is the point order of P. Since $M^{'}$ is the smallest positive integer satisfying $M^{'}P = \mathcal{O}$, MP is not a zero point for $M < M^{'}$. Note that either $\{S \not| \Phi_{p_i}(P)\}$ or, $\{S|\Phi_{p_i}(P)$ and $S^2 \not| \Phi_{p_i}(P)\}$ satisfies for each i since $M^{'} =$ lcm $(\Phi_{p_1}(P), \Phi_{p_2}(P), \dots, \Phi_{p_k}(P))$. If $S|\Phi_{p_i}(P)$ and $S^2 \not| \Phi_{p_i}(P)$ for all i, then $\Phi_{p_i}(P) \not| M$ for all i since M is not a multiple of S and $\Phi_{p_i}(P)$ is a multiple of S. Hence, each MP_i is not a zero point $\mathcal{O}_{p_i}$. Hence in this case, MP is an ordinary point. Otherwise, (i.e. if there exists i that satisfies $S \not| \Phi_{p_i}(P)$), MP is a semi-zero point.

The probability that MP is a semi-zero point is bigger than $Q(k,S) = (1 - \frac{1}{S})^{k+1}\frac{k}{S}$ in Eq. (1), since that $S|M^{'}$ and $S^2 \not| M^{'}$ implies that $S|\#E_n$ and $S^2 \not| \#E_n$ and $S|\Phi_{p_i}(P)$.

Hence, similar analysis as in Sect. 3 leads us to the conclusion that this algorithm runs in randomly polynomial time $\mathrm{O}((\log n)^5)$. □

Next, we will prove that if both $\mathrm{FCT}(n)$ and EDLP mod p are tractable, EDLP mod n is also tractable.

Theorem 4 For composite n that is a product of distinct odd primes, it holds that EDLP mod $n \leq_P \mathrm{FCT}(n) \oplus$ EDLP mod p.

Proof: The following algorithm solves EDLP mod n in polynomial time.

Solving EDLP mod n algorithm using an oracle for FCT(n) and an oracle for EDLP mod p

Input: composite $n(=\prod_{i=1}^{k} p_i)$, elliptic curve $E_n : y^2 \equiv x^3+ax+b \pmod{n}$ and two points G and A over E_n.

Output: Integer α such that $\alpha G = A$ over E_n.

Step 1 Using an oracle for FCT(n), find k prime factors of n as $p_1, p_2, \cdots, p_k$.

Step 2 Using an oracle for EDLP mod p_i, solve $\alpha_i G = A$ over E_{p_i} for each i.

Step 3 Calculate $\#E_{p_i}(a,b)$ for each i.

Step 4 Obtain α which satisfies the system of the congruences: $\{\alpha = \alpha_i \bmod \#E_{p_i}\}_{i=1}^{k}$ by using the Chinese Remainder Theorem.

This algorithm is completed in polynomial time, $O((\log n)^6)$. Structures of the direct sum of G and A are denoted by $G = \langle G_1, G_2, \ldots, G_k\rangle$ and $A = \langle A_1, A_2, \ldots, A_k\rangle$, respectively, or simply as $G = \langle G_i\rangle_{i=1}^{k}$ and $A = \langle A_i\rangle_{i=1}^{k}$. It holds that $\alpha_i G_i = A_i$ for each i from step 2. It follows that $\alpha G = \alpha\langle G_i\rangle_{i=1}^{k} = \langle \alpha G_i\rangle_{i=1}^{k} = \langle(\alpha \bmod \#E_{p_i})G_i\rangle_{i=1}^{k} = \langle \alpha_i G_i\rangle_{i=1}^{k} = \langle A_i\rangle_{i=1}^{k} = A$. Hence, α obtained in step 4 is the solution to the equation, $\alpha G = A$. □

6 Conclusion

We investigated the factoring problem and some problems in elliptic curve theory over Z_n. We proved that if we know $\#E_n$, we can easily factor n. We also proved that if we know $\#E_n \bmod d$ for all primes d less than $2\log n$, instead of $\#E_n$ itself, we can easily factor n. Finally, we also proved that if we can solve the elliptic curve discrete logarithm problem modulo n, we can easily factor n.

References

1. E. Bach. Discrete logarithm and factoring. Technical Report UCB/CSD 84/186, Computer Science Division (EECS), University of California, Berkley, Calfornia, 1984.
2. L. Charlap, R. Coley and D. Robbins. Enumerate of rational points on elliptic curves over finite fields. preprint, 1991.
3. N. Demytko. A new elliptic curve based analogue of RSA. *Proc. of EUROCRYPT'93*, LNCS765: pp.40–49, 1994.
4. N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48: pp.203–209, 1987.
5. K. Koyama, U. Mauer, T. Okamoto and S. Vanstone. New public-key schemes based on elliptic curves over the ring Z_n. *Proc. of CRYPTO'91*, LNCS576: pp.345–357, 1992.
6. K. Koyama. Fast RSA-type scheme based on singular curves $y^2 + axy \equiv x^3 \pmod{n}$. *Proc. of EUROCRYPT'95*, LNCS921: pp.329–340, 1995.
7. H. Lenstra. Factoring integer with elliptic curves. *Ann. of Math.*, 126: pp.649–673, 1987.

8. D. L. Long. Random equivalence of factorization and computation of orders. Technical Report 284, Princeton University, Department of Electrical Engineering and Computer Science, April 1981.
9. A. Menezes. *Elliptic curve public key cryptosystems.* Kluwer academic publishers, 1993.
10. G. L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13: pp.300–317, 1976.
11. V. S. Miller. Use of elliptic curves in cryptography. *Proc. of CRYPTO'85*, LNCS218: pp.417–426, 1986.
12. P. Riebenboim. *The little book of big primes.* Springer–Verlag, 1991.
13. R. Schoof. Elliptic curve over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44: pp.483–494, 1985.
14. J. Silverman. *The Arithmetic of Elliptic Curves.* Springer–Verlag, New York, 1986. GTM 106.
15. H. Woll. Reductions among number theoretical problems. *Information and Computation*, 72: pp.167–179, 1987.