Regular paper http://dx.doi.org/10.13067/JKIECS.2015.10.10.1093

유한체위에서의 타원곡선을 이용한 고속 소인수분해법에 관한 연구

김용태*

Fast Factorization Methods based on Elliptic Curves over Finite Fields

Yong-Tae Kim*

요 약

RSA 암호법의 안전성은, 덫 문으로 사용되는 큰 정수 N을 소인수분해하는 일이 매우 어렵다는 사실에 기반을 두고 있기 때문에, RSA 암호법을 이용하여 암호문을 전달할 때와 그 암호문을 공격할 때에는 합성수를 소인수분해하는 방법이 매우 중요한 문제이다. 100자리 이상의 큰 정수 N을 소인수분해하는 지금까지 알려진 가장 빠른 알고리즘은 일반 수체 체(General Number Field Sieve, GNFS) 알고리즘이지만, 현대의 공개키 암호법에서 자주 사용되는 20~25 자리의 수(64.~83 비트)정도의 소인수를 찾아내는 가장 빠른 알고리즘은 Lenstra의 타원곡선법이다. 그러나 Lenstra의 방법은 실행시간의 대부분을$M \bullet P \bmod N$ 을 계산하는 과정에서 소비하게 되었기 때문에, Montgomery와 Koyama는 $M \bullet P \bmod N$ 을 고속으로 계산하는 방법을 제안하였다. 본 논문에서는 Montgomery와 Koyama의 방법을 분하여, 최적의 매개변수를 선택하고 곱셈횟수를 줄여서 구축한 효율적인 $M \bullet P \bmod N$ 계산 알고리즘을 제안한다. 분석결과, Montgomery와 Koyama의 알고리즘보다 제안한 알고리즘이 H/W에서의 구현시간을 약 20% 단축하였다.

ABSTRACT

Since the security of RSA cryptosystem depends on the difficulty of factoring integers, it is the most important problem to factor large integers in RSA cryptosystem. The Lenstra elliptic curve factorization method(ECM) is considered a special purpose factoring algorithm as it is still the best algorithm for divisors not greatly exceeding 20 to 25 digits(64 to 83 bits or so). ECM, however, wastes most time to calculate $M \bullet P \bmod N$ and so Montgomery and Koyama both give fast methods for implementing $M \bullet P \bmod N$. We, in this paper, further analyze Montgomery and Koyama's methods and propose an efficient algorithm which choose the optimal parameters and reduces the number of multiplications of Montgomery and Koyama's methods. Consequently, the run time of our algorithm is reduced by 20% or so than that of Montgomery and Koyama's methods.

키워드

RSA Cryptosystem, Elliptic Curve, Factorization, Addition of Points
RSA 암호법, 타원 곡선, 소인수 분해, 점의 덧셈

Ⅰ. 서 론

1977년에 Rivest, Shamir와 Adleman의 연구에 의해 체계화된 공개키 암호시스템으로, 메시지의 암호화

* 교신저자 : 광주교육대학교 수학교육과
• 접 수 일 : 2015. 08. 07
• 수정완료일 : 2015. 10. 13
• 게재확정일 : 2015. 10. 23
• Received : Aug 07, 2015, Revised : Oct 13, 2015, Accepted : Oct 23, 2015
• Corresponding Author : Yong-Tea Kim
Dept. of Mathematics Education, Gwangju National University of Education,
Email : ytkim@gnue.ac.kr

뿐만 아니라 전자서명이 가능한 최초의 알고리즘인 RSA 암호법[1]은 공개키를 기반으로 하는 암호법과 전자서명 인증을 요구하는 전자상거래 등에서 광범위하게 활용되고 있다. RSA 암호법의 안전성은, 덫 문(trapdoor)으로 사용되는 큰 정수 N을 소인수분해하는 일이 매우 어렵다는 사실에 기반을 두고 있으며, 초등 정수론의 지식만으로도 누구나 그의 안전성을 쉽게 이해할 수 있기 때문에 모든 국가에서 우호적으로 사용되고 있다. 따라서 암호학 자체의 발전과 RSA 암호법을 이용하는 암호문을 공격하기 위해서 세계 각국의 국가기관, 기업은 물론 관심 있는 수학자와 암호학자들이 합성수의 소인수분해법 연구에 지대한 노력을 경주하고 있다. 일반적으로 두 소수의 곱인 정수 N을 선택할 때, 2^t-1 형태의 합성수를 자주 사용하는데 이러한 형태의 이진수의 특성에 관한 연구[2-3]도 활발하게 진행되고 있다. 어떤 양의 실수 k에 대해서 b-bit 정수를 다항식 시간(polynomial time) 즉, $O(b^k)$ 안에 소인수분해 할 수 있는 알고리즘이 알려지지 않고 있다. 양자 컴퓨터를 이용하지 않는다는 전제하에서, 100자리 이상의 큰 정수 N을 소인수분해하는 지금까지 알려진 가장 빠른 알고리즘은 일반 수체 체(General Number Field Sieve, GNFS)알고리즘이며, GNFS를 이용하여 b-bit 정수 N을 소인수분해하는 데에 가장 빠른 실행시간은 $O(\exp((\frac{64}{9}b)^{\frac{1}{3}}(\log b)^{\frac{2}{3}}))$ 이다. 두 번째로 빠른 알고리즘은 다중다항식 이차 체(Multiple Polynomial Quadratic Sieve)이며, 세 번째가 1985년에 Lenstra[4]가 제안한 유한체 위에서의 타원곡선법이다. 그런데 현대의 공개키 암호법에서 사용하는 암호문은 유효기간이 짧거나 전자서명을 자주 바꾸는 등의 이유로, 응용분야에서는 공개키를 100자리 이하의 정수 N을 로 사용하는 경우가 많다. 따라서 공개키 암호의 사용자 또는 공격자는 100자리 이하의 정수 N의 소인수분해방법을 알아내는 일이 매우 중요하기 때문에 20~25 자리의 수(64.~83 비트)정도의 소인수를 찾아내는 가장 빠른 알고리즘인 Lenstra의 타원곡선법을 사용하고 있다. 그러나 Lenstra의 방법은 실행시간의 대부분을 $M \bullet P \bmod N$ 을 계산하는 과정에서 소비하게 되었기 때문에, Montgomery[5]와 Koyama[6]는 $M \bullet P \bmod N$ 을 고속으로 계산하는 방법을 고안하였다. 타원곡선법을 이용하여 현재까지 찾아낸 가장 큰 약수는 2013년에 R. Propper가 찾아낸 83자리 수[7]이다. 그러나 그의 방법은 타원곡선의 개수를 충분히 크게 하여 그 중에서 우연히 약수를 얻는 확률적인 방법으로, 약수의 자릿수가 증가함에 따라 필요한 타원곡선의 개수가 비선형적(non-linear)으로 증가하기 때문에 일반화하기에는 적절하지 않은 방법으로 간주되고 있다. 본 논문에서는 합성수를 소인수 분해하거나 또는 합성수의 약수를 구하는, Montgomery와 Koyama가 개발한 $M \bullet P \bmod N$ 의 고속계산법의 곱셈 횟수를 줄이고 매개변수를 효율적으로 선택하는 방법을 적용한 고속 소인수분해 알고리즘을 제안한다.

II. 타원곡선을 이용한 소인수분해법

이 장에서는 유한체위에서의 타원곡선을 이용하여 합성수를 소인수분해하는 방법을 알아보기로 한다. Lenstra[4]가 제안한 유한체 위에서의 타원곡선법은 복잡한 사영(projective)좌표계를 이용한 방법이므로, 유한체위에서의 타원곡선에 대한 일반적인 이론, 기호와 타원곡선위의 점들의 덧셈법은 이해하기 쉬운 직교좌표계를 이용한 Silverman[8]을 참고하였으며, Lenstra의 유한체위에서의 타원곡선을 이용한 소인수분해법(Elliptic Curve Factorization Method)을 관례에 따라 ECM으로 표기하기로 한다.

2.1 ECM의 개요

Lenstra의 ECM[4]은 다음과 같은 과정으로 주어진 자연수 N의 인수를 구하게 된다.

2.1.1 Setup

N이 큰 정수일 때 유한체 $GF(N)$ 위에서의 타원곡선

$$\mathrm{E}: By^2 = x^3 + Ax^2 + x \mod N, A, B \in GF(N) \quad (1)$$

과 그 위의 한 점 $P(x_0, y_0)$ 를 택한다.

2.1.2 타원곡선위의 덧셈

타원곡선 E 위의 두 점 P, Q의 덧셈은 Silverman[8]과 같이 정의하며, 그 덧셈을 $P \oplus Q$ 로 표기하면, 타원곡선 E 위의 모든 점들은

연산 $\oplus$ 하에서 군(group)을 이룬다. 그 유한군의 위수(order) S는 상수 A,B에 따라 결정되며, 이 유한군의 임의의 점 P에 대하여 $k \bullet P = P \oplus \cdots \oplus P$($k$번)로 표기하기로 하면, $S \bullet P = O$ (무한원점) 이 된다. 실제로, 타원곡선 위의 무한원점 O이 연산 $\oplus$ 하에서의 항등원이다.

2.1.3 $k! \bullet P$ 의 계산법

a. $2 \bullet P$ 를 계산한다.

b. $3 \bullet (2 \bullet P)$ 를 계산한다.

c. $4 \bullet (3! \bullet P)$ 를 계산한다. …

2.1.4 N의 약수 구하기

기하학적으로, 타원곡선위의 두 점 P,Q의 덧셈 $P \oplus Q$은 두 점을 잇는 현의 기울기($\bmod N$)이므로 N을 법으로 하여 확장된 유클리드 알고리즘으로 계산하는 두 잉여류의 나눗셈이 된다. 특히, 임의의 정수 u를 어떤 정수 $v(\bmod N)$로 나누는 것은 두 정수 v,N의 최대공약수 $\gcd(v,N)$을 구하는 것과 같다. 따라서 다음의 결과를 얻게 된다.

a. $\gcd(u,N)=1$ 인 경우

타원곡선위의 두 점 P,Q을 잇는 현의 기울기가 u/v형태이면, $v=0$이 된다. 즉, $P \oplus Q = O$이 된다.

b. $\gcd(u,N) \neq 1, N$ 인 경우

$P \oplus Q$ 의 값은 존재하지 않으므로, 타원곡선위의 점들의 집합은 덧셈$\oplus(\bmod N)$에 대하여 군이 아니다. 따라서 $1 < \gcd(v,N) < N$ 즉, N의 자명하지 않은 약수를 구하게 된다.

2.1.5 타원곡선위의 모든 점의 덧셈의 값이 존재하지 않는 경우에는 다른 곡선을 택하여 1 단계로 돌아간다.

2.2 $M \bullet P \bmod N$ 의 고속화

Montgomery와 Koyama는 Lenstra의 ECM의 계산속도를 더욱 빠르게 하기 위하여 다음과 같이 $M \bullet P \bmod N$ 의 계산방법을 수정하였다[5-6].

m,n을 서로 다른 정수라고 하자 그러면 타원곡선 E : $By^2 = x^3 + Ax^2 + x \bmod N, A, B \in GF(p)$ 위의 점 $m \bullet P$, $(m+n) \bullet P$, $(m-n) \bullet P$ 과 $2n \bullet P$ 의 x성분 간에 다음과 같은 관계식이 성립한다[3-4].

$$x_{m+n} = (x_m x_n - 1)^2 / x_{m-n}(x_m - x_n)^2 \quad (2)$$

$$x_{2n} = (x_n^2 - 1)^2 / 4x_n(x_n^2 + Ax_n + 1) \quad (3)$$

자연수 N을 소인수분해하기 위해서는 타원곡선 E 위의 점의 y성분은 필요하지 않다. 이제 타원곡선 E 위의 점 $r \bullet P$ 의 x성분 x_r을 유리수 c_r/d_r로 표기한다면 식(2)는

$$c_{m+n} = d_{m-n}(c_m c_n - d_m d_n)^2 \quad (4)$$

$$d_{m+n} = c_{m-n}(c_m d_n - d_m c_n)^2 \quad (5)$$

이 되며, 식(3)은

$$c_{2n} = (c_n^2 - d_n^2)^2 \quad (6)$$

$$d_{2n} = 4c_n d_n (c_n^2 + Ac_n d_n + d_n^2) \quad (7)$$

이 된다.

따라서 타원곡선 E위의 점의 x성분을 나눗셈을 하지 않고 구할 수 있게 된다. 식(4)와 (5)의 계산에서는 8회의 곱셈과 2회의 덧셈이 필요하며, 식(6)과 (7)의 계산에서는 7회의 곱셈과 3회의 덧셈이 필요함을 알 수 있다.

III. 제안하는 고속 PECM

이제, Montgomery와 Koyama의 $M \bullet P \bmod N$ 의 계산방법을 개선하여 계산 속도를 높이는 우리의 알고리즘(PECM)을 제안하기로 한다.

3.1 x성분의 변환

Montgomery와 Koyama이 표현한 식(4)와 (5)를 H/W 실행시에 연산의 횟수를 줄이기 위하여 다음과 같이 변환한다.

$$c_{m+n} = d_{m-n}[(c_m - d_m)(c_n + d_n) + (c_m + d_m)(c_n - d_n)]^2 \quad (8)$$

$$d_{m+n} = c_{m-n}[(c_m - d_m)(c_n + d_n) - (c_m + d_m)(c_n - d_n)]^2 \quad (9)$$

따라서 $M \bullet P \bmod N$ 의 x성분의 계산에 필요한 식(8)과 (9)는 6회의 곱셈과 4회의 덧셈만으로 실행되게 된다. 비슷한 방법으로 식(6)과 (7)을 다음과 같이 변환할 수 있다.

$$c_{2n} = (c_n + d_n)^2 (c_n - d_n)^2 \tag{10}$$

$$d_{2n} = [(c_n + d_n)^2 - (c_n - d_n)^2][(c_n - d_n)^2 + \{(c_n + d_n)^2 - (c_n - d_n)^2\}(A+2)/4] \tag{11}$$

따라서 만일 나타나는 상수$(A+2)/4$ 를 미리 계산해 두면, $2n \bullet P$ 의 x성분의 계산에 필요한 식(10)과 (11)은 5회의 곱셈과 4회의 덧셈만으로 실행되게 된다.

3.2 PECM의 고속 알고리즘 1단계

PECM에서는 임의의 합성수 N의 소인수 q를 찾아내기 위하여 $M \bullet P \bmod N$ 을 계산한다. 만일 M이 유한군의 위수 S의 배수인 경우에는 반드시 적어도 하나의 소인수 q를 찾을 수 있다. S는 소인수가 Lenstra[2] 알고리즘의 최적화 상수인 $b = e^{\sqrt{1/2 \log N \log\log N}}$ 이하 즉, b-smooth 로 설정하고 M역시 b-smooth로 설정하기로 한다. PECM의 고속 알고리즘 1단계 알고리즘의 개요는 다음과 같다.

PECM 1

Step 1. 소인수분해 할 합성수 N과 b-smooth 정수 M 을 정한다.

Step 2. 정수 $A (\bmod N)$를 임의로 선택하여 $GF(p)$ 위의 타원곡선 E를 정한다. 또한 두 정수 c_1과 d_1을 임의로 선택하여 타원곡선 E상의 한 점 $P(c_1, d_1, c_1 + d_1, c_1 - d_1)$ 을 정한다.

Step 3. 식(8)~(11)을 이용하여 $M \bullet P \bmod N$ 을 계산한다.

Step 4. $M \bullet P \bmod N$ 의 x성분의 분모 d_M과 N의 최대공약수 $a = \gcd(d_M, N)$ 을 계산한다.

Step 5. 만일 $a \neq 1$ 이라면 a가 N의 약수이다. 만일 $a = 1$ 이면 Step 2로 돌아가서 다른 타원곡선을 선택하여 위의 과정을 반복 실행한다.

3.3. PECM의 고속 알고리즘 2단계

만약 유한군의 위수 S는 어떤 자연수 b_1 이하인 소인수 이외에 특별히 큰 소인수 $a (> b_1)$ 를 한 개만 포함하는 경우에는 2단계의 PECM알고리즘을 적용한다. 2단계 알고리즘의 개요는 다음과 같다.

PECM 2

Step 1. M은 b_1이하의 모든 소수의 곱으로 정한다. 즉, $M = \Pi_{p_i \le b_i} p_i$ 로 정한다.

Step 2. S의 특별히 큰 소인수 a가 포함되는 구간 $[b_1, b_2]$ 를 찾아서, 이 구간에 속하는 소수를 차례로 $q_1, q_2, \cdots, q_L$ 라고 하자. 그러면 PECM 1.의 최종점인 $M \bullet P \bmod N$ 을 P^* 라고 놓고, N을 법으로 $q_1 \bullet P^*, q_2 \bullet P^*, \cdots, q_L \bullet P^*$ 를 차례로 계산한다.

Step 3. $q_1 \bullet P^*, q_2 \bullet P^*, \cdots, q_L \bullet P^*$ 의 x성분의 분모와 N과의 최대공약수를 계산하여 N의 소인수를 차례로 구한다. 또는 위의 각 점의 x성분의 분모를 모두 곱한 수와 N과의 최대공약수를 계산한다.

PECM의 최적인 매개변수 M, A 등의 선택에 관한 문제는 Smart[9]와 Cosset[10]의 분석결과를 적용하였다.

IV. 수치실험 및 프로그램

이장에서는 제안하는 두 알고리즘 PECM 1과 PECM 2에 따라서, Pentium 166 MHZ CPU에서 Mathematica 4.0[11]을 이용한 프로그램을 사용하여 얻은 수치 실험결과와 프로그램은 다음과 같다. 단, NN은 목적 수(target numner), M은 b_1-smooth number 이고 c_1, d_1, AP는 각각 PECM 1의 Step 2의 정수, Result는 NN의 약수이다.

4.1 수치실험의 예

1) $NN = 2^{105} - 1$, c_1=12, d_1=13, AP=14, M=30!
Result=696990232001

2) $NN = 2^{105} - 1$, c_1=35432, d_1=663, AP=28,

M=762048000
Result=2068220273

3) NN=$(2^{431}-1)/836\times3449$, c_1=12, d_1=55, AP=99, M=2×10^4-smooth number
Result=2785266783178489=36238481×76859369

4) NN=$(2^{431}-1)/836\times3449$, c_1=12, d_1=23, AP=67, M=2×10^4-smooth number
Result=168223563758872497=4642152737×36238481

5) NN=5636963037465601×9860942209386451, c_1=78, d_1=33, AP=90, M=2×10^5-smooth number
Result=5636963037465601 (P16)

4.2 결과 분석

수치실험의 예 1),2)는 합성수 $2^{105}-1$의 약수를 타원곡선을 바꿔가면서 구한 결과이고, 3),4)는 합성수 $2^{431}-1$에 Fibonacci-형의 나눈 후 공격법(divide and conquer method)을 적용하여 먼저 $2^{431}-1$을 836×3449로 나눈 수를 타원곡선을 바꿔가면서 그의 약수를 구한 것이다. 5)는 16자리 소수(P16) 5636963037465601의 배수에 우리의 알고리즘을 적용하여 그 소수를 찾아낸 결과이다. $M \bullet P \bmod N$ 계산과정에서, 제안하는 고속 PECM은 Montgomery와 Koyama의 알고리즘에 비해서, 덧셈은 3회 증가했으며 곱셈은 4회 줄었다. 그러나 유한체 $GF(N)$ 위에서의 연산에서는, 곱셈에 필요한 AND gate에서 소모되는 시간이 덧셈에 필요한 XOR gate에서 소모되는 시간보다 훨씬 더 비중이 크며 최적의 매개변수 M, A를 선택하였기 때문에, PECM 1,2 알고리즘이 Montgomery와 Koyama의 알고리즘보다 실행시간이 약 20% 단축된 것을 알 수 있다.

4.3 프로그램

```
(*b:=Exp[Sqrt[(1/2)n[Log[p*Log[Log[p]]]]]]:
k:=PrimePi[9999]:
m=(50!)*Product[Prime[ii],{ii,k}:*)
NECM1[NN_Integer, C1_, D1_, AP_Integer]:=
  Block[{}, c1=C1:d1=D1:n=NN:ap=AP:m=M:
    hh=Round[0.6175291m]:If{Mod[hh,2]==0,
      hh=hh+1}:
        r11=Mod[(c1+d1)^2,n]:dummy=
          {c1+d1)^2-(c1-d1)^2:
       r12=Mod[dummy*((c1-d1)^2+dummy*
      ap),n]:cn=c1:dn=d1:cm=c1:dm=d1:
    ms=1:ns=1:gs=hh:fs=m-hh:
  Label[1]:
  If[fs>gs, fs=fs-gs:ns=ms+ns:cmn=cn:
    dmn=dn:  cn=r11:dn=r12,gs=gs-fs:cmn=cm:
      dmn=dm:cm=r11:dm=r12:ms=ms+ns:
        If[gs===0,  dummy=GCD[dm,n]:
       Return[{cm,  dm,  dummy}]]]:
r1=(cm-dm)(cn+dn):r2=(cm+dm)(cn-dn):
        r11=Mod[dmn(r1-r2)^2,n]:r12=
      Mod[cmn(r1-r2)^2,n]:
    Goto[1]
  ]:

NECM2[NN_Integer, C1_, D1_, AP_Integer,B1_
  Integer, B2_Integer]:=Block[{c1,d1},
   b1=PrimePi[B1]:b2=PrimePi[B2]:1=b2-b1:
     n=NN:
      dum=NECM1[n, C1, D1, AP,
        M]:c1=dum[[1]]:d1=dum[[2]]:
        dum=NECM1[NN, c1, d1, AP, Prime
         [b1+1]]:cs=dum[[1]]:ds=dum[[2]]:t=ds:
       c2=(c1+d1)^2*(c1-d1)^2:dum=(c1+d1)^2;
     d2=dum*(dum*AP+(c1-d1)^2):
    cs2=dum[[1]]:ds2=dum[[2]]:
For[i=2, i<=1, i++, j:=(Prime[b1+i]-Prime[
  b1+i-1])/2:s1=cs:s2=ds:r1=cs2:r2=ds2:
    For[k=1, k<=j, k++, dum1=s1:dum2=s2:
     s1=Mod[r1((s1-s2)(c2+d2)+(s1+s2)
       (c2-d2)),n]:
       s2=Mod[r2((s1-s2)(c2+d2)+(s1+s2)
         (c2-d2)),n]:r1=dum1:r2=dum2
        ]:
```

```
        s=s2:t=Mod[t*s,n]
       ]:
      a=GCD[t,n]:
     Return[{a, s1, s2}]
]:
```

V. 결 론

RSA 암호법을 전자상거래 등의 일상생활에 사용하는 경우에는 공개키를 100자리 이하의 정수 N을 로 사용하는 경우가 많다. 따라서 공개키 암호의 사용자 또는 공격자는 100자리 이하의 정수 N의 소인수분해 방법을 알아내는 일이 매우 중요하기 때문에 20~25 자리의 수(64.~83 비트)정도의 소인수를 찾아내는 가장 빠른 알고리즘인 Lenstra의 타원곡선법을 사용하고 있다. 그러나 Lenstra의 방법은 실행시간의 대부분을 $M \bullet P \bmod N$ 을 계산하는 과정에서 소비하게 되었기 때문에, Montgomery[5]와 Koyama[6]는 $M \bullet P \bmod N$을 고속으로 계산하는 방법을 고안하였다. 본 논문에서는 일반적으로 두 소수의 곱인 정수 N을 선택할 때, 2^t-1 형태의 합성수를 자주 사용하는데 이러한 형태의 이진수의 특성에 관한 연구[12]를 참조하여, 합성수를 소인수 분해하거나 또는 합성수의 약수를 구하는 Montgomery와 Koyama가 개발한 $M \bullet P \bmod N$의 고속계산법의 곱셈 횟수를 줄이고 매개변수를 효율적으로 선택하는 방법을 적용한 고속 소인수분해 알고리즘을 제안하였다. 분석결과, H/W에서의 구현시간을 약 20% 단축하였다.

감사의 글

본 논문은 광주교육대학교 2015년도 학술연구비 지원에 의한 것임

References

[1] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM* vol. 21, no. 2, Feb. 1978, pp. 120 - 126.

[2] U. Choi and S. Cho, "Design of Binary Sequence with optimal Cross-correlation Values," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 6, no. 4, 2011, pp. 539-544.

[3] S. Cho, J. Kim, U. Choi, and S. Kim, "Cross-correlation of linear and nonlinear GMW-sequences generated by the same primitive polynomial on $GF(2^p)$," *The Korea Institute of Electronic Communication Sciences 2011 Spring Conf.* Busan, Korea, vol. 5 no. 1, June 2011, pp.155-158.

[4] H. W. Lenstra Jr, "Elliptic curve factorization and primality testing," *Proc. Advances in Cryptology-CRYPTO '85*, Springer-Verlag, London, UK., Aug., 1985, pp. 409-416.

[5] P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Mathematics Computation,* vol. 48, no.3, 1987, pp. 243-264.

[6] K. Koyama, "Factoring using a fast elliptic curve method," *J. of Japanese Institute of Electronic Information Communication, D,* vol. 70, no. 12, 1987, pp. 2730-2738.

[7] M. W. Baesagade and S. Meshram, "Overview of History of Elliptic Curves and its use in cryptography," *Int. J. of Scientific & Engineering Research,* vol. 5, no. 4, Apr. 2014, pp. 466-469.

[8] J. H. Silverman, *The Arithmetic of Elliptic Curves.* Graduate Texts in Mathematics, no. 106, New York: Springer-Verlag. 1986.

[9] N. P. Smart, "How Secure Are Elliptic Curves over Composite Extension Fields?," *Advances in Cryptology-EUROCRYPT'2001 Proc.,* LNCS 2045, Springer, Innsbruck, Austria, May 2001.

[10] R. Cosset, "Factorization with genus 2 curves," *Mathematics of Computation,* vol. 79, no. 2, 2010, pp. 1191 - 1208.

[11] S. Wolfram, *Mathematica, 4th Ed.* Wolfram Champaign: Research, Inc., 1999.

[12] H. Kim, S. Cho, M. Kwon, and H. An, " A study on the cross sequences," *J. of the Korea Institute of Electronic Communication Sciences,* vol. 7, no. 1, 2012, pp. 61-67.

저자 소개

김용태(Yong-tae Kim)

1976년 : 공주사범대학 수학교육과(이학사)
1986년 : 고려대학교 대학원 수학과 (이학석사)
1991년 : 고려대학교대학원 수학과(이학박사)
2000년 : 서울대학교 대학원 수학교육과(교육학석사)
2008년 : 서울대학교 대학원 수학교육과(박사과정 수료)
1992년 ~ 현재 : 광주교육대학교 수학교육과 교수
※ 관심분야 : ECC, 정수론적 암호학, 공개키암호학