情報保護學會論文誌
第11券 第6號, 2001. 12

타원곡선상의 고속 곱셈연산을 위한 새로운 분해 알고리즘*

박 영 호**, 김 용 호**, 임 종 인***, 김 창 한***, 김 용 태****

A new decomposition algorithm of integer for fast scalar multiplication on certain elliptic curves

Young-Ho Park**, Yong Ho Kim**, Jongin Lim**, Chang Han Kim***, Yong Tae Kim****

요 약

최근에 Gallant, Lambert, Vanstone은 소수체 위에 정의된 타원곡선이 효율적으로 계산 가능한 자기준동형을 가질 때 스칼라 곱을 가속화하는 방법을 제안하였다. 이 방법은 실제로 자기준동형의 특성다항식의 고유치를 사용하여 스칼라를 분해하는데 기반을 두고 있다. 본 논문에서는 그러한 타원곡선의 자기준동형 환의 원소를 이용하여 스칼라를 분해하는 개선된 알고리즘을 제안한다. 이 알고리즘은 Gallant 등의 알고리즘보다 속도면에서 효율적이며 분해 성분들의 구체적인 상한 값을 줄 수 있다.

ABSTRACT

Recently, Gallant, Lambert and Vanstone introduced a method for speeding up the scalar multiplication on a family of elliptic curves over prime fields that have efficiently-computable endomorphisms. It really depends on decomposing an integral scalar in terms of an integer eigenvalue of the characteristic polynomial of such an endomorphism. In this paper, by using an element in the endomorphism ring of such an elliptic curve, we present an alternate method for decomposing a scalar. The proposed algorithm is more efficient than that of Gallant's and an upper bound on the lengths of the components is explicitly given.

keyword : *Elliptic Curve, Scalar Multiplication, Endomorphism.*

Ⅰ. 서 론

타원곡선 암호시스템(ECC)에 바탕을 둔 모든 프로토콜의 구현 속도는 타원곡선 상의 점 P에 스칼라 k배의 계산속도에 가장 많은 영향을 받는다. 따라서 스칼라 곱의 고속연산 방법은 타원곡선 자체 또는 바탕체의 특성에 따라 다양하게 연구되었다[2]. 특히, Koblitz[5]와 Solinas[14,15]는 유한체 GF(2) 위에 정의된 타원곡선(Koblitz curve)에서 제곱연산 대신에 Frobenius 자기준동형(endomorhohism)을 사용하여 스칼라 곱을 빠르게 하는 방법을 제안하였다. 그리고 Meier와 Staffelbach[7], Müller[6] 등은

* 본 연구는 정보통신부 대학정보통신연구센터 육성지원사업 지원 및 소프트웨어진흥원 관리로 수행하였습니다.
** 고려대학교 정보보호기술연구센터(CIST)({youngho,kyh,jilim}@cist.korea.ac.kr)
*** 세명대학교 컴퓨터수리정보학과(CHKIM235@chollian.net)
**** 광주교육대학교 수학교육학과(ytkim@gnue.ac.kr)

이러한 방법을 확장하여 표수가 2인 작은 유한체 위의 확장체에서 정의된 타원곡선에 대해서 적용하였다. 나아가 Smart는 표수가 홀수인 경우에도 적용가능 하다는 것을 보였다[13]. 하지만 이들의 방법은 확장체 위에서 정의된 어떤 타원곡선에서만 적용될 뿐 큰 소수체 위에서 정의된 타원곡선에서는 적용할 수 없다.

최근에 Gallant, Lambert, Vanstone[4]은 큰 소수체 위에서 정의된 타원곡선이 효율적인 계산이 가능한 자기준동형을 갖는 경우, 이 자기준동형을 사용하여 스칼라 곱의 고속연산이 가능한 효율적인 방법을 제안하였다.

이 방법은 자기준동형에 대한 특성 방정식의 고유치 λ를 이용하여, 타원곡선의 점의 개수가 큰 소인수 n를 가질 때, 임의의 정수 $k \in [1, n-1]$를 $k = k_1 + k_2\lambda \bmod n$ ($k_1, k_2 \in [0, \lfloor \sqrt{n} \rfloor]$)로 효율적인 분해가 가능하다면, k를 이렇게 분해한 후 windowed simultaneous multiple exponentiation 방법을 사용하여 지금까지 제안된 가장 빠른 방법보다도 대략 50% 더 빠르게 연산속도를 향상시킬 수 있음을 보였다. 이 방법을 사용하기 위하여 k를 분해하는 효율적인 알고리즘의 존재가 선행되어야 한다. [4]에서 제안된 효율적인 분해 알고리즘은 분해성분인 k_1, k_2의 크기의 상한 값을 구체적으로 제시하지 못하였다. 분해성분의 상한 값을 구체적으로 아는 것은 소프트웨어 구현시 메모리의 효율적인 사용의 관점에서 중요할 수 있다.

따라서 본 논문에서는 n을 노름(norm)으로 갖는 자기준동형 환의 원소를 이용하여 스칼라를 분해하는 효율적인 알고리즘을 제시한다. 이 알고리즘은 Gallant의 알고리즘보다 더 효율적이며, 복소 이차체의 order에서 노름의 계산에 의하여 분해된 각 구성값 k_1, k_2의 크기의 상한 값도 분명하게 주어진다.

이 논문의 구성은 다음과 같다. 2, 3절에서는 타원곡선의 기본성질과 Gallant의 방법을 간략하게 소개한다. 4절에서는 새로운 k의 분해방법을 설명하고 알고리즘을 제시한다. 그리고 5절에서는 [4]에서 제시한 타원곡선들에 대한 분해성분들의 상한값을 구체적으로주고, 6절에서 실제 구현한 분해 결과를 바탕으로 7절에서 결론을 맺는다.

II. 자기준동형 환(endomorphism ring)

본 장에서는 타원곡선에 관한 기본적인 성질을 살펴본다. 유한체 F_q위에 주어진 타원곡선 E가 다음과 같이 주어졌다 하자.

$$E(F_q) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$a_i \in F_q$. $E(F_q)$는 F_q의 유리점들과 무한점(the point at infinity) O으로 이루어진 집합이고, E의 F_q 자기준동형 환(endomorphism ring) $End(E)$는 F_q위에서 정의되고 O을 보존하는 사상(mapping) $f : E \rightarrow E$이다. Frobenius 자기준동형 $\Phi \in End(E)$는 $\Phi(x, y) = (x^q, y^q)$으로 주어지는 함수이다. 그리고 $End(E)$에서 $\Phi^2 - t\Phi + q = 0$과 특히 $|E(F_p)| = q + 1 - t$를 만족한다.

[정리 1] (Hasse의 정리)
F_q 위에 정의된 타원곡선 $E(F_q)$의 점의 개수를 n이라 하면

$$|t| = |q + 1 - n| \leq 2\sqrt{q}$$

이다.[12]

주어진 타원곡선의 점의 개수를 구하는 것은 위의 정리로부터 Frobenius 자기준동형의 자취(trace)를 구하는 것과 동치임을 알 수 있다. 일반적으로 타원곡선의 점의 개수를 구하는 것은 그리 쉽지 않으며 Schoof[10], Schoof-Elkies-Atkin[3], Satoh[9] 알고리즘 등을 사용하여 구해야 한다.

암호분야의 응용에서 supersingular 타원곡선은 MOV[8]의 공격에 취약하므로 non-supersingular인 타원곡선을 이용한다. Non-supersingular 타원곡선은 complex multiplication을 갖으며 이 경우에 E의 자기준동형 환은 복소이차체 $Q(\sqrt{t^2 - 4q})$의 order이다. 그러므로 $End(E) \subset Q(\sqrt{t^2 - 4q})$이다.

만약 음수인 $-D$를 order의 discriminant라 하면 $End(E) \cong Z[\delta] = Z + Z[\delta]$을 만족한다. 여기서 $-D$가 짝수이면 $\delta = \frac{\sqrt{-D}}{2}$이고 홀수이면 $\delta = \frac{1+\sqrt{-D}}{2}$이다.

예로 j-invariant가 0이거나 1,728인 경우 자기준동형 환은 최대(maximal) order와 동형임을 알 수 있다.

Ⅲ. 효율적인 자기준동형을 이용한 스칼라 곱

p를 큰 소수, E를 유한체 F_p위에 정의된 타원곡선이라하고 η는 $End(E)$에 있는 효율적으로 계산가능한 자기준동형 이라 하자. 암호학적으로 타원곡선 $E(F_p)$의 점의 개수는 큰 소인수 n(160비트 이상)를 가져야 한다. P를 큰 소수 n를 점의 위수로 갖는 $E(F_p)$의 점이라 하자. 그러면 η는 λ의 곱으로 $E(F_p)$에 작용한다. 여기서 λ는 η modulo n의 특성 다항식의 근이다. Gallant 등[4]은 스칼라 곱의 속도를 개선하기 위하여 η를 사용하였으며 그 방법은 다음과 같다.

정수 $k\in[1, n-1]$를 Extented Euclidean Algorithm을 사용하여 $k=k_1+k_2\lambda \bmod n$으로 분해한다. 여기서 $k_1,\ k_2\approx\sqrt{n}$이다. 그런 다음에

$$\begin{aligned} kP &= (k_1+k_2\lambda)P \\ &= k_1P+k_2\lambda P \\ &= k_1P+k_2\eta(P) \end{aligned}$$

를 이용하여 계산한다. $\eta(P)$가 쉽게 계산되므로 $k_1P+k_2\eta(P)$을 계산하는데 windowed simultaneous multiple exponentiation 방법을 적용하면 일반적인 방법에 비하여 속도를 60% 정도 개선할 수 있다. 또한 160 비트 크기의 n에 대해서 window의 크기를 w=3로 놓아 계산한 결과, 지금까지의 가장 빠른 일반적인 방법보다도 대략 50% 빠르다는 것을 제시하였다[4].

이 방법을 사용하는데 있어 중요한 문제점은 임의의 정수 k를 어떻게 효과적으로 λ를 이용하여 분해하느냐 하는 것이다. 먼저 Gallant등이 제안한 주어진 정수 n, λ에 대하여 임의의 정수 k를 분해하는 방법을 살펴보자[4]. 이 방법은 크게 두 단계로 나눌 수 있다.

첫 번째, 준동형사상

$$\begin{aligned} f:\ Z\times Z &\to Z_n \\ (i,j) &\mapsto (i+j\lambda) \bmod n \end{aligned}$$

과 Extended Euclidean Algorithm을 이용하여 $f(v_1)=f(v_2)=0$을 만족하는 일차독립인 크기가 작은 벡터(short vector) $v_1,\ v_2\in Z\times Z$를 구한다. 이 과정은 선행계산 과정에 의해 k와 관계없이 독립적으로 계산할 수 있다.

두 번째, 선형대수를 이용하여 $(k,0)$에 근접한 Zv_1+Zv_2에 있는 벡터를 찾는다. 그러면 k_1, k_2가 다음 식에 의하여 주어진다.

$$(k_1,k_2)=(k,0)-(\lfloor b_1\rceil v_1+\lfloor b_2\rceil v_2),$$

여기서 $b_1, b_2\in Q$, $(k,0)=b_1v_1+b_2v_2$이고 $\lfloor b\rceil$는 b에 가장 가까운 정수이다.

[알고리즘 1] ((k_1,k_2) 찾기)

Input : $k\approx n$, the short vectors $v_1=(x_1,y_1), v_2=(x_2,y_2)$ Output : $k \equiv k_1+k_2\lambda \pmod n$
1) $d=x_1y_2-x_2y_1,\ a_1=y_2k,\ a_2=-y_1k.$ 2) $z_i=\lfloor a_i/d\rceil$ (for $i=1,2$). 3) $k_1=k-(z_1x_1+z_2x_2),$ $k_2=z_1y_1+z_2y_2$

이 알고리즘은 2번의 $\lfloor\cdot\rceil$ 연산과 8번의 큰 정수곱셈연산이 필요하다. 또한 [4]에서 제시했듯이, 알고리즘 1을 통해 구한 분해성분의 크기는

$$\|(k_1,k_2)\|\le\max\{\|v_1\|,\|v_2\|\}$$

이다. 따라서 v_1, v_2의 상한 값이 구체적으로 주어지지 않으므로 (k_1,k_2) 분해 성분인 k_1,k_2의 상한 값을 주지 못함을 알 수 있다.

Ⅳ. 새로운 k의 분해 방법

본 장에서 우리는 대수적 수론 관점에서 k를 분해하는 새로운 방법을 제시한다. 먼저 non-supersingular 타원곡선 E의 자기준동형 환 $End(E)$는 복소 이차체 $K=Q(\sqrt{-D})$ $(D>0)$의 최대 order O_K에 포함됨을 기억하자. 또한 $Z[\eta]\subset End(E)\subset O_K$이다. 일반적으로 효율적인 계산이 가능한 자명하지 않은 η는 정수가 아니므로 다음의 방정식을 만족한다.

$$\eta^2 - t_\eta \eta + n_\eta = 0 \quad (1)$$

그리고 η의 discriminant는 적당한 정수 s에 대해 $D_\eta = t_\eta^2 - 4n_\eta = D \cdot s^2$로 표현된다. 왜냐하면 $\eta \in Q(\sqrt{D_\eta}) = Q(\sqrt{-D}) = K$를 만족해야하기 때문이다. 만일 어떤 원소 $\alpha = a + b\eta \in Z[\eta]$가 존재하여

$$N_{Z[\eta]/Z}(\alpha) = n \text{와 } (\alpha)P = O \quad (2)$$

를 만족한다고 가정하자. 그러면 정수 k를 $Z[\eta]$의 원소로 보고, $Z[\eta]$가 적당한 실수 μ에 대한 μ-Euclidean domain임을 고려하여 (2)를 만족하는 원소 α로 나눈다. 따라서

$$k = \beta\alpha + \rho \quad \beta, \rho \in Z[\eta],$$
$$N_{Z[\eta]/Z}(\rho) < \mu N_{Z[\eta]/Z}(\alpha)$$

와 같이 표현할 수 있다. 마침내 (2)를 이용하여

$kP = (\beta\alpha + \rho)P = \beta(\alpha(P)) + \rho(P) = \rho(P)$가 된다. 그리고 $\rho \in Z[\eta]$ 이므로 $\rho = k_1 + k_2\eta$라 놓을 때

$$kP = k_1 P + k_2 \eta(P)$$

로 계산할 수 있고 따라서 Gallant 등[4]과 같은 방법을 적용할 수 있다.

이 분해 방법은 Gallant[4]의 방법보다 약간의 계산상 효율성을 가지며, 특히, 분해성분인 k_1, k_2의 상한 값을 구체적으로 구할 수 있다. 이를 얻기 위하여 다음의 정리가 필요하다.

[정리 3]

임의의 원소 $x \in Z[\eta]$에 대하여 x를 α로 나눈 몫을 β, 나머지 ρ라 할 때 다음을 만족한다.

$$x = \beta\alpha + \rho \quad \beta, \rho \in Z[\eta],$$
$$N_{Z[\eta]/Z}(\rho) < \mu N_{Z[\eta]/Z}(\alpha) \text{ with}$$
$$0 < \mu \le \begin{cases} (9+4n_\eta)/16 & \text{if } t_\eta \text{ is odd,} \\ (1+n_\eta)/4 & \text{if } t_\eta \text{ is even.} \end{cases}$$

[증명]

먼저 $\eta^2 - t_\eta\eta + n_\eta = 0$을 만족하므로 $\eta = (t_\eta + \sqrt{D_\eta})/2$, $D_\eta = t_\eta^2 - 4n_\eta$라 하자.

상수 $c = -\lfloor t_\eta/2 \rfloor$, $\eta' = \eta + c$라 놓자. 그러면 $\eta' = \begin{cases} (1+\sqrt{D_\eta})/2 & \text{if } t_\eta \text{ is odd,} \\ \sqrt{D_\eta}/2 & \text{if } t_\eta \text{ is even.} \end{cases}$ 이다.

$\alpha = a_1 + b_1\eta'$로 표현하여 $Z[\eta'] = Z[\eta]$에서 x를 α로 나누면, $\gamma = x/\alpha = x\bar{\alpha}/\alpha\bar{\alpha} = \frac{c_1 + c_2\eta}{n}$, $c_1, c_2 \in Z$로 쓸 수 있다. $b_1 = \lfloor c_1/n \rceil$, $b_2 = \lfloor c_2/n \rceil$라 놓고 $\beta = b_1 + b_2\eta$로 놓으면 $\rho = \alpha(\gamma - \beta) \in Z[\eta]$을 얻는다. 또한

$$N_{Z[\eta]/Z}(\rho)/N_{Z[\eta]/Z}(\alpha) = N_{Q[\eta]/Q}(\gamma - \beta)$$
$$\le N_{Z[\eta]/Z}(\tfrac{1}{2} + \tfrac{1}{2}\eta') = 1/4 N_{Z[\eta]/Z}(1+\eta')$$
$$= \begin{cases} 1/4 N_{Z[\eta]/Z}((3+\sqrt{D_\eta})/2) & \text{if } t_\eta \text{ is odd,} \\ 1/4 N_{Z[\eta]/Z}((2+\sqrt{D_\eta})/2) & \text{if } t_\eta \text{ is even.} \end{cases}$$
$$= \begin{cases} (9-D_\eta)/16 \le (9+4n_\eta)/16 & \text{if } t_\eta \text{ is odd.} \\ (4-D_\eta)/16 \le (1+n_\eta)/4 & \text{if } t_\eta \text{ is even.} \end{cases} \square$$

이제 [정리 3]의 증명과정을 이용하여 k와 $\alpha = a + b\eta$로부터 나머지 $\rho = k_1 + k_2\eta$를 구하는 알고리즘을 제안하자. 이 알고리즘은 [4]에서와 같이 선행계산과정과 본 과정 두 가지 과정으로 나누어진다. 먼저 선행과정에서는 다음 1), 2), 3)을 계산한다. 본 알고리즘 2는 정리 3의 증명과정에서 쉽게 유도된다.

1) $n = N_{Z[\eta]/Z}(\alpha)$, $t_\eta = Tr_{Z[\eta]/Z}(\eta)$, $c = -\lfloor t_\eta/2 \rfloor$
2) $\eta' = \eta + c$, $N = N_{Z[\eta]/Z}(\eta')$, $T = Tr_{Z[\eta]/Z}(\eta')$
3) $a_1 = a - bc$, $b_1 = b$ ($a = a_1 + b_1\eta'$로 표현).

[알고리즘 2] (k를 $\alpha = a + b\eta$로 나누기)

Input : $(k \approx n)$, n, N, T, c, a_1, b_1 Output : $\rho = k_1 + k_2\eta$ such that $N_{Z[\eta]/Z}(\rho) < \mu N_{Z[\eta]/Z}(\alpha)$
1) $x_1 = k(a_1 + b_1 T)$, $x_2 = -kb_1$. 2) $y_i = \lfloor x_i/n \rceil$ (for $i = 1, 2$). 3) $k'_1 = k - (a_1 y_1 - N b_1 y_2)$, $k'_2 = -(a_1 y_2 + b_1 y_1 + T b_1 y_2)$, 4) $k_1 = k'_1 + k'_2 c$, $k_2 = k'_2$.

본 알고리즘 2는 알고리즘 1과 마찬가지로 2번의 $\lfloor \cdot \rceil$ 연산과 8번의 큰 정수 곱셈연산이 필요하다. (상수 $T = \begin{cases} 1 & \text{if } t_\eta \text{ is odd,} \\ 0 & \text{if } t_\eta \text{ is even.} \end{cases}$ 임을 주의.)

그러나 효율적인 계산이 가능한 자기준동형 η의

t_η, c와 n_η의 크기가 작을 경우(일반적으로 알려진 경우 크기가 작음 ([4] 또는 5절 참고)) 6번의 큰 정수곱셈연산 만이 필요하다. 따라서 알고리즘 2가 알고리즘 1에 비해 좀더 효율적임을 알 수 있다. 이 두 알고리즘을 비교하기 위해 크기가 160비트인 n에 대하여 PentiumIII에서 구현한 running time을 제시한다.

[표 1] 분해 알고리즘 1, 2의 속도비교(ms)

	$t_\eta=0$ $n_\eta=1$	$t_\eta=-1$ $n_\eta=1$	$t_\eta=1$ $n_\eta=2$	$t_\eta=0$ $n_\eta=2$
Algorithm 1	0.072	0.069	0.071	0.069
Algorithm 2	0.053	0.054	0.053	0.054

V. 분해성분들의 상한 값

본 절에서는 [4]에서 제안한 소수체 F_p 위의 간단한 연산에 의하여 효율적으로 계산 가능한 자기준동형을 갖는 non-supersingular 타원곡선에 대한 구체적인 상한 값을 구하고 제안된 분해방법과 [4]의 방법을 비교한다.

[예제 1]

$p\equiv 1 \bmod 4$인 소수이고 타원곡선 E_1가 유한체 F_p 위에서 다음과 같다 하자:

$$E_1/F_p : y^2=x^3+ax$$

β를 F_p에서 위수가 4인 원소라 놓으면 함수 $\eta: E_1\to E_1$, $\eta(x,y)=(-x,\beta y)$, $O\to O$는 F_p위에서 E_1의 자기준동형이 된다. 또한 η의 이차방정식은 $\eta^2+1=0$이고 $t_\eta=0$, $n_\eta=1$을 만족한다. 자기준동형 환 $End(E_1)$는 $Z[\eta]$와 동형이고 $Q(\sqrt{-1})$의 maximal order이다.

[예제 2]

$p\equiv 1 \bmod 3$인 소수이고 타원곡선 E_2가 유한체 F_p 위에서 다음과 같다 하자:

$$E_2/F_p: y^2=x^3+b$$

γ를 F_p에서 위수가 3인 원소라 놓으면 함수 $\eta: E_2\to E_2$, $\eta(x,y)=(\gamma x,y)$, $O\to O$는 F_p위에서 E_2의 자기준동형이 된다.

또한 η의 이차방정식은 $\eta^2+\eta+1=0$이고, $t_\eta=-1$, $n_\eta=1$을 만족한다. 자기준동형환 $End(E_2)$는 $Z[\eta]$와 동형이고 $Q(\sqrt{-3})$의 maximal order이다.

위 예제 1, 2에서 주어진 자기준동형 η는 유한체 F_p에서 한번의 상수 배로 구할 수 있다.

[예제 3]

$p>3$인 소수이고 -7이 F_p에서 완전제곱수일 때, $\omega=(1+\sqrt{-7})/2$와 $a=(\omega-3)/4$라 놓고 타원곡선 E_3가 유한체 F_p위에서 다음과 같다 하자:

$$E_3/F_p: y^2=x^3-3/4x^2-2x-1$$

그러면 함수 $\eta: E_3\to E_3$, $\eta(x,y)=(\omega^{-2}\frac{x^2-\omega}{(x-a)}, \omega^{-3}y\frac{x^2-2ax+\omega}{(x-a)^2})$, $O\to O$는 F_p 위에서 E_3의 자기 준동형이 된다.

또한 η의 이차방정식은 $\eta^2-\eta+2=0$이고 $t_\eta=1$, $n_\eta=2$을 만족한다. 자기준동형 환 $End(E_3)$는 $Z[\eta]$와 동형이고 $Q(\sqrt{-7})$의 maximal order이다.

[예제 4]

$p>3$인 소수이고 -2이 F_p에서 완전제곱수일 때, 타원곡선 E_4가 유한체 F_p위에서 다음과 같다 하자:

$$E_4/F_p: y^2=4x^3-30x^2-28$$

그러면 함수 $\eta: E_4\to E_4$, $\eta(x,y)=\left(-\frac{2x^2+4x+9}{4(x+2)}, -y\frac{2x^2+8x-1}{4\sqrt{-2}(x+2)^2}\right)$, $O\to O$는 F_p 위에서 E_4의 자기준동형이 된다. 또한 η의 이차방정식은 $\eta^2+2=0$이고 $t_\eta=0$, $n_\eta=2$을 만족한다. 자기준동형 환 $End(E_4)$는 $Z[\eta]$와 동형이고 $Q(\sqrt{-2})$의 maximal order이다.

이제 위 [예제 1-4]에 주어진 타원곡선들에 대한 스칼라곱에 관하여 살펴보자. 먼저 P를 큰 소수 위수 n를 갖는 $E(F_p)$의 점이라 하자.

위 예제들에서 알 수 있듯이 자기준동형 환 $Z[\eta]$는 모두 $Q(\sqrt{-D})$의 maximal order이고 이들 모두 principal domain이므로[16], 원소 $\alpha=a+b\eta\in Z[\eta]$가 존재하여 $N_{Z[\eta]/Z}(\alpha)=n$와 $(\alpha)P=O$를 만족한다. 여기서 이러한 α를 구하는 방법으로는 Shanks의 알고리즘[11], lattice reduction 방법[17], Cornacchia 알고리즘[1]등이 사용될 수 있다.

[보조정리 4]

α가 $N_{Z[\eta]/Z}(\alpha)=n$를 만족한다 하자. 임의의 원소 $x\in Z[\eta]$에 대하여 x를 α로 나눌 때 다음을 만족하는 δ와 ρ가 $Z[\eta]$에 존재한다.

$$x=\beta\alpha+\rho,$$
$$N_{Z[\eta]/Z}(\rho)\le\begin{cases}n/2 & \text{for } E_1,\\ 3n/4 & \text{for } E_2,\\ n & \text{for } E_3,\\ 3n/4 & \text{for } E_4.\end{cases}$$

[증명]

정리 3에 의해 쉽게 유도됨 □

[정리 5]

임의의 정수 k에 대하여 k를 α로 나눈 나머지를 $\rho=k_1+k_2\eta\in Z[\eta]$라 하면 다음을 만족한다.

$$\max\{|k_1|,\ |k_2|\}\le\begin{cases}\sqrt{n/2} & \text{for } E_1,\\ \sqrt{n} & \text{for } E_2,\\ \sqrt{8n/7} & \text{for } E_3,\\ \sqrt{3n}/2 & \text{for } E_4,\end{cases}$$

[증명]

먼저 E_1인 경우에, $N_{Z[\eta]/Z}(\rho)=N_{Z[\eta]/Z}(k_1+k_2\eta)=k_1^2+k_2^2$이고 보조정리 4에 의하여 $N_{Z[\eta]/Z}(\rho)=k_1^2+k_2^2\le n/2$인 것을 알 수 있다. 따라서 $\max\{|k_1|,\ |k_2|\}\le\sqrt{n/2}$을 만족한다.

E_2인 경우 $N_{Z[\eta]/Z}(\rho)=k_1^2+k_2^2-k_1k_2$이다. 만약 $k_1k_2\le 0$이면 [보조정리 4]로부터

$$\max\{|k_1|,\ |k_2|\}\le\sqrt{3n/4}$$

을 얻는다. 또한 $k_1k_2>0$일 경우 $|k_2|\ge|k_1|>0$라 가정하자. [보조정리 4]을 이용하여

$$k_1^2+k_2^2-k_1k_2=|k_1|^2+|k_2|^2-|k_1||k_2|$$
$$=(|k_1|-\frac{1}{2}|k_2|)^2+\frac{3}{4}|k_2|^2\le 3n/4$$

인 것을 알 수 있다.

따라서 $|k_2|^2\le n$와 $\max\{|k_1|,\ |k_2|\}=|k_2|\le\sqrt{n}$을 얻는다.

다른 E_3, E_4의 경우에도 동일한 방법으로 위 결과를 쉽게 얻을 수 있다. □

Ⅵ. 두 가지 분해방법의 구현결과

본 절에서는 5절에서 제시된 타원곡선들에 대해서 [4]에서 제안된 알고리즘 1과 본 논문에서 제안된 새로운 알고리즘 2를 사용하여 각각 분해결과를 비교할 것이다. 이것은 이 두 알고리즘이 거의 같은 분해결과를 도출한다는 것을 보여줄 것이다.

암호학적 응용의 관점에서, 타원곡선 $E(F_p)$의 점의 개수는 160비트 이상의 큰 소인수 n를 가져야 한다. 이러한 곡선을 찾기 위해 먼저 주어진 타원곡선의 점의 개수를 구해야 되는데 이 문제는 일반적으로 Schoof[10] 또는 Schoof-Elkies-Atkin[3] 알고리즘들을 사용하여 구하게 되며 이는 그리 쉬운 것은 아니다. 따라서 본 절에서는 자기준동형 환의 어떤 원소 π가 존재하여 작은 cofactor h에 대해

$$N_{Z[\eta]/Z}(\pi)=p\text{와 }N_{Z[\eta]/Z}(\pi-1)=hn \qquad (3)$$

을 만족하는 p, n(160 비트 이상의 소수)의 경우에 대하여 구현하였다. 위 (3)의 조건은 만일 타원곡선이 존재하여 그 점의 개수가 $\#E(F_p)=hn$을 갖는다면 반드시 만족해야만 한다. 하지만 반대로 (3)의 조건을 만족한다고 해서 이러한 타원곡선이 존재한다는 보장은 없다는 것에 주의하자.

다음 표에서는 예제 1-4에서 주어진 자기준동형 η에 대해 각각 20개의 서로 다른 (p,n,λ)에 대하여 10^5개의 난수 $k(<n)$를 구현한 결과를 보여준다. 이 표에서는 두 알고리즘의 분해정도를 측정할 수 있는 값들이 주어져있다. 임의의 정수 k에 대한 분해 값을 알고리즘 1을 사용한 경우 $k=k_1+k_2\lambda\pmod n$, [알고리즘 2]를 사용한 경우에는 $\rho=k'_1+k'_2\eta$로 표시하자.

표에서 ratio는 $\max\{|k_1|,|k_2|\}$과 $\max\{|k'_1|,|k'_2|\}$ 사이의 큰 값을 작은 값으로 나눈 비 중 최대 값의 올림 값을 나타낸다. 많은 n, p에 대한 구현결과 ratio값이 3이하였고 이 사실은 두 분해성분들의 비트수 차이가 많아야 2비트 정도임을 알 수 있다. 또한

$\max_1 = \max\{|k_1|, |k_2|\}$, $\max_2 = \max\{|k'_1|, |k'_2|\}$로 표시할 때 $\max_1/\sqrt{n}$과 $\max_2/\sqrt{n}$값을 계산하여 정리 5의 타당함을 보였다. 또한 그 결과 예제 1-4의 경우 분해성분의 크기가 $\sqrt{n}$을 넘지 않음을 알았다.

〔표 2〕 예제 1의 구현 결과

예제 1 : $\eta^2+1=0$	
분해성분이 같은 (p,n,λ)의 갯수	20
분해성분이 다른 (p,n,λ)의 갯수	0
$\max_1/\sqrt{n}$ 의 최대값	0.706

〔표 3〕 예제 2의 구현 결과표

예제 2 : $\eta^2+\eta+1=0$		
분해성분이 같은 (p,n,λ)의 갯수		10
분해성분이 다른 (p,n,λ)의 갯수		10
동일한 분해	서로 다른 분해	ratio
74.9%	25.1%	3
$\max_1/\sqrt{n}$ 의 최대값		0.844
$\max_2/\sqrt{n}$ 의 최대값		0.995

〔표 4〕 예제 3의 구현 결과

예제 3 : $\eta^2-\eta+2=0$		
분해성분이 같은 (p,n,λ)의 갯수		10
분해성분이 다른 (p,n,λ)의 갯수		10
동일한 분해	서로 다른 분해	ratio
75.0%	25.0%	3
$\max_1/\sqrt{n}$ 의 최대값		0.852
$\max_2/\sqrt{n}$ 의 최대값		0.997

〔표 5〕 예제 4의 구현 결과

예제 4 : $\eta^2+2=0$		
분해성분이 같은 (p,n,λ)의 갯수		12
분해성분이 다른 (p,n,λ)의 갯수		8
동일한 분해	서로 다른 분해	ratio
75.0%	25.0%	2
$\max_1/\sqrt{n}$ 의 최대값		0.865
$\max_2/\sqrt{n}$ 의 최대값		0.865

특별히, 〔예제 1〕의 경우에는 두 알고리즘 모두가 동일한 분해성분을 얻을 수 있으며, 본 구현 결과 91.2% 정도의 동일한 분해성분을 얻었다.

Ⅶ. 결 론

본 논문에서 우리는 효과적으로 계산 가능한 자기준동형 η를 갖는 소수체위의 타원곡선에서, Gallant[4]가 제시한 η의 특성방정식의 근 λ를 사용하여 정수 k를 분해하는 대신, 타원곡선의 자기준동형 환의 원소 α를 사용하여 k를 분해하는 새로운 알고리즘을 제안하였다. 본 알고리즘은 〔4〕에서 제안한 알고리즘보다 속도면에서 효율적이었으며 또한 정수 k의 분해성분들의 상한 값을 구체적으로 줄 수 있었다. 그러나 두 방법 모두 구현하여 실험한 결과 평균 91.2% 정도 동일한 분해결과를 얻을 수 있었고 많은 경우 n, p, λ에 대해 100% 동일한 결과를 얻을 수 있었다. 서로 다른 분해성분을 얻는 경우에도 두 분해성분의 크기의 차이가 많아야 2비트 정도 밖에 나지 않음을 알았다.

참 고 문 헌

〔1〕 A. O. L. Atkin and F. Morain, "Elliptic curves and Primality Proving" Math. of Comp. 61(1993), No. 203, pp. 29~68.

〔2〕 Ian Blake, Gadiel Seroussi and Nigel Smart, "Elliptic Curves in Cryptography". London Mathematical Society Lecture Note Series. 265, Cambridge University Press, 1999.

〔3〕 N. Elkies, "Ellpitic and modular curves over finite fields and related computational issues", Computaional Perspectives on Number theory, pp. 21~76, 1998.

〔4〕 R. Gallant, R. Lambert and S. Vanstone, "Faster Point Multiplication on Elliptic Curves", Advances in Cryptology-Crypto '2001, pp. 190~200.

〔5〕 N. Koblitz, "CM-curves with good cryptographic properties", Advances in Cryptology-Crypto '91, 1992, pp. 279~287.

〔6〕 V. Müller, "Fast multiplication in elliptic

curves over small fields of characteristic two", Journal of Cryptology, 1998, pp. 219~234.

[7] W. Meier, O. Staffelbach, "Efficient multiplication on certain non-supersingular elliptic curves", Advances in Cryptology-Crypto'92, 1992, pp. 333~344.

[8] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curves logarithms to logarithms in a finite field", IEEE Trans. Info. Theory, 39, pp. 1639~1646, 1993.

[9] T. Satoh, "The canonical lift of an ordinary elliptic curve over a finite field and its point counting", J. Ramanujan Math. Soc., 15, pp. 247~270, 2000.

[10] R. Schoof, "Counting points on elliptic curves over finite fields", J. Théorie des Nombres de Bordeaux,7, pp. 219~254,1995.

[11] D. Shanks, "Five number theoretic algorithms" In Proc. 2nd Manitoba Conference on Numerical Mathematics, 1972, pp. 51~70.

[12] J. H. Silverman, 'Advanced Topics in the Arithmetic of Elliptic Curves', Springer-Verlag, New York, 1994.

[13] N. Smart, "Elliptic curve cryptosystems over small fields of odd characteristic", Journal of Cryptology, 1999, pp. 141~145.

[14] J. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves", Advances in Cryptology-Crypto '97, 1997, pp. 357~3711.

[15] J. Solinas, "Efficient arithmetic on Koblitz curves", Design, Codes and Cryptography, 2000, pp. 195~249.

[16] I. Stewart, D. Tall, "Algebraic Number Theory", Chapman and Hall, Halsted Press, 1979.

[17] B. Vallee, Une approche geometrique des algorithmes de reduction des reseaux en petite dimension, 1986, These, Universite de Caen.

〈著者紹介〉

박 영 호 (Young-Ho Park) 정회원
1990년 2월 : 고려대학교 수학과 학사
1993년 2월 : 고려대학교 수학과 석사
1997년 2월 : 고려대학교 수학과 박사
2001년 ~현재 : 고려대 정보보호기술연구센터 객원조교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜

김 용 호 (Yong-Ho Kim) 정회원
2000년 2월 : 고려대학교 수학과 학사
2000년 3월~현재 : 고려대학교 수학과 석사 과정
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜

임 종 인 (Jong-in Lim) 정회원
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1999년 2월~현재 : 고려대학교 자연과학대학 정교수, 한국통신정보보호학회 편집위원장
고려대학교 정보보호대학원 원장, 고려대학교 정보보호기술연구센터 센터장
〈관심분야〉 블록 암호 및 스트림 암호의 분석 및 설계, 암호 프로토콜, 공개키 암호 알고리즘의 분석

김 창 한 (Chang-Han Kim) 정회원
1985년 2월 : 고려대학교 수학과 학사
1987년 2월 : 고려대학교 수학과 석사
1992년 2월 : 고려대학교 수학과 박사
2000년 ~현재 : 세명대학교 컴퓨터수리정보학과 부교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜

김 용 태 (Yong Tae Kim)
1976년 2월 공주사대 수학교육과 학사
1991년 2월 고려대학교 수학과 박사
1992년 ~ 현재 광주교육대학교 수학교육과 부교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜