通信情報保護學會論文誌
第5卷 第1號, 1995. 3

효율적 반복 연산을 위한 타원 곡선*

이 은 정**, 최 영 주**

Elliptic Curves for Efficient Repeated Additions

Eun Jeong Lee and Young Ju Choie

요 약

타원 곡선을 사용한 암호 시스템은 안전도가 높고 smart card에 응용할 수 있지만 타원 곡선에서의 연산이 유한체에서의 연산보다 느리기 때문에 실용화를 위해서는 타원 곡선위에서 고속 연산 기법, 고속 반복 연산 기법이 개발되어야 한다. 1991년 Koblitz는 Frobenious map의 trace $Tr(\varphi)$가 1인 anomalous 타원 곡선을 제안하였고, 이 곡선의 사용으로 타원 곡선위의 한 점 P를 반복 더하는 mP를 효과적으로 계산할 수 있었다. 본 논문에서는 사전 계산을 할 경우 Koblitz의 F_2 위에서의 anomalous 타원 곡선과 같이 보통의 반복 연산 방법(repeated-doubling method)보다 3배 빨리 mP를 계산할 수 있는 유한체 F_4 위에서 정의된 타원 곡선을 제안한다. 사전 계산을 하지 않는 경우 제안된 타원곡선 위에서는 mP 계산시 가장 많은 더하기 횟수는 $\frac{3}{2}\log_2 m + 1$번이다.

Abstract

In spite of the good security of the cryptosystem on an elliptic curve defined over finite field, the cryptosystem on an elliptic curve is slower than that on a finite field. To be practical, we need a better method to improve a speed of the cryptosystem on an elliptic curve defined over a finite field. In 1991, Koblitz suggested to use an anomalous curve over F_2, which is an elliptic curve with Frobenious map whose trace is 1, and reduced a speed of computation of mP. In this paper, we consider an elliptic curve defined over F_4 with Frobenious map whose trace is 3 and suggest an efficient algorithm to compute mP. On the proposed elliptic curve, we can compute multiples mP with $\frac{3}{2}\log_2 m + 1$ additions in worst case.

---

* 이 논문은 1994년도 산업과학기술 연구소(RIST)의 순수기초 연구비(R94004), 기초 과학 연구소(N94123-3)의 지원으로 연구되었음.

** 포항공과대학교 수학과

1. 서 론

Diffie와 Hellman이 공용키이 암호 시스템을 제안한 이후 여러 가환 군이 암호 시스템에 사용되었고 그 위에서의 이산 대수 문제가 시스템의 안전도와 깊이 연관되었다. ElGamal은 유한체 F_p^*(p는 소수) 위에서 암호 알고리듬을 제안했고, 컴퓨터 실행상 편리함을 위해 $F_{2^n}^*$(n은 자연수)가 널리 사용되고 있다. 이러한 암호 시스템을 공격하기 위해 여러가지 알고리듬이 개발되었는데 그 중 index calculus 알고리듬을 사용하여 유한체 위에서의 이산 대수 문제를 subexponential time하에 풀 수 있게 되었다. 따라서, 유한체 위에서 만들어진 암호시스템의 안전도를 위해서 p나 2^n의 크기가 커져야 한다. 그러나, smart card와 같이 저장 용량이 크지 않을 경우도 있으므로 유한체의 크기가 작으면서도 안전한 암호 시스템이 요구되었다. 그 후, Koblitz[1]와 Miller[7]는 유한체 위에서 정의된 타원 곡선을 이용할 것을 제안했는데, 현재까지 타원 곡선 위에서의 이산 대수 문제를 subexponential time하에 풀 수 있는 알고리듬이 개발되지 않았으므로 작은 유한체 위에서 정의된 타원 곡선을 사용하여 안전한 암호 시스템을 만들 수 있다.

이산 대수 문제를 풀기 위한 일반적인 알고리듬으로 군의 형태의 상관없이 적용되는 Shanks의 baby step giant step 알고리듬은 군의 order를 나누는 가장 큰 소수의 이중 근(square root)정도로 exponential time이 걸리고 유한 가환 군에 적용되는 Pohlig-Hellman 알고리듬 또한 군의 order가 큰 소수로 나누어질 때는 비효율적이다[4]. 그 외 Weil pairing 을 이용하여 유한체 $F_q(q = p^n)$ 위에서 정의된 타원 곡선 E/F_q에서의 이산대수문제를 확장된 유한체 $F_{q^k}(k$는 자연수)위의 문제로 전환시킨 MOV공격이 있다[6]. 이 공격은 확장된 유한체에 index calculus 알고리듬을 적용하여 이산대수문제를 풀게하므로 타원 곡선을 사용한 암호 시스템이 안전하기 위해 확장 차수 k가 크거나 q가 커야 한다. Supersingular 타원 곡선에서는 확장차수가 작아($k \leq 6$)q를 크게 선택해야 하므로 이를 이용한 암호 시스템의 효율성이 줄어든다. 따라서 supersingeular 타원 곡선은 피해야 하고, 다행히 임의의 타원 곡선에서 이산대수문제는 MOV 공격에 의해서도 충분히 exponential하게 어려울 확률이 높다[3].

그러나 타원 곡선에서 군 연산(더하기라 하겠다)이 유한체에서의 연산보다 느리다. 암호 알리고듬은 랜덤 수 m에 대해 타원 곡선위의 한 점 P를 반복해서 더하는 계산을 많이 하므로 실용적인 암호 시스템을 위해서 더하기 속도를 단축시켜야 하는 것이 과제이다. 그리고 mP를 계산할때 보통 반복 더하기 연산으로하면 가장 많은 경우 $2\log_2 m$번 더해야하나 이것보다 더하기 횟수를 줄일 수 있는 타원 곡선을 선택하는 것도 실용화를 위한 한가지 방법이다.

본 논문에서는 Koblitz가 제안한 anomalous 타원곡선의 장점을 간단히 살피고 같은 장점을 가진 암호시스템에 응용 가능한 다른 타원곡선을 제안하고자 한다. 제안된 타원 곡선위에서 mP를 계산산하는데 가장 많은 경우 $\frac{3}{2}\log_2 m + 1$번의 더하기를 한다.

또한, 이러한 성질을 가지며 암호 시스템에 적용 가능한 타원 곡선을 실질적으로 어떻게 선택하는가도 설명하였다.

2. Anomalous 타원 곡선과 그 효율성

E/F_q를 유한체 $F_q(q = 2^n)$위에서 정의된 non-supersingular 타원곡선이라 하자. E/F_q의 일반적인 형태는

$$E/F_q = \{(x, y) \in \overline{F_q} \times \overline{F_q} | y^2 + xy = x^3 + a_2x^2 + a_6\} \cup \{O\}$$
$$a_2, a_6 \in F_q, a_6 \neq 0$$

이고, $E(F_q)$는

$$E(F_q) = \{(x, y) \in F_q \times F_q | y^2 + xy = x^3 + a_2x^2 + a_6\} \cup \{O\}$$

이고[5]. 여기서, O는 무한대점이다.

$P = (x_1, y_1)$와 $Q = (x_2, y_2)$를 $E(F_q)$위의 점이라 할때 $-P$, $P + Q = (x_3, y_3)$는 다음과 같이 계산한다[5]

$$x_3 = \begin{cases} \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_2 + x_2 + a_2, & P \neq Q\text{일때,} \\ x_1^2 + \frac{a_6}{x_1^2}, & P = Q\text{일때,} \end{cases}$$

$$y_3 = \begin{cases} \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 (x_1 + x_3) + x_3 + y_1, & P \neq Q\text{일때,} \\ x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)x_3 + x_3, & P = Q\text{일때,} \end{cases}$$

$$-P = (x_1, y_1 + x_1).$$

이 연산하에 $E(F_q)$가 가환군이 됨은 잘 알려진 사실이다.

P와 Q가 다른 경우, $P + Q$ 계산시, 유한체에서의 곱셈 연산이 2번, 제곱 연산이 1번, 그리고 곱셈에 대한 역수를 계산하는 횟수가 1번이다. P와 Q가 같은 경우, $P + Q$ 계산시, 유한체에서의 곱셈연산이 3번, 제곱 연산이 2번, 그리고 곱셈에 대한 역수를 계산하는 횟수가 1번이다. 따라서, 유한체만을 이용한 암호 시스템보다 타원 곡선을 이용하는 경우에 유한체에서의 연산이 더 많이 요구되어 계산 속도가 느리게 된다. 그러나, $-P$를 구할때엔 유한체에서의 덧셈만 요구되므로 계산량에 포함되지 않는다(유한체에서 덧셈 연산은 배타적 논리합(Exclusive-Or)을 이용하기 때문이다).

Koblitz는 Frobenious map의 특성 다항식(characteristic polynomial)을 이용하여 mP를 계산할때 타원 곡선위에서의 연산 횟수를 줄일 수 있는 타원 곡선을 제안하였다.

E/F_q에서 E/F_q로 정의된 Frobenious map

$$\varphi : (x, y) \rightarrow (x^q, y^q)$$

의 trace를 t, 즉 $Tr(\varphi) = t$, 라 하면 φ의 특성 다항식은

$$T^2 - tT + q = 0$$

이므로[8], $E(F_q)$위의 한 점 P를 q번 더하고자 할 때는

$$qP = t\varphi(P) - \varphi^2(P)$$

로써 계산할 수도 있다. 여기서 $Tr(\varphi) = t$가 작을수록 qP를 적은 더하기 횟수로 계산할 수 있는데 $t = 0$인 경우에는 E가 supersingular 타원 곡선이 되어[5] MOV 공격에 약하다[2, 6]. 그 다음 선택으로 $t = \pm 1$을 선택한다. $Tr(\varphi) = t = 1$인 타원 곡선을 anomalous 타원 곡선이라 하였다. 특히, F_2위에서 정의된 anomalous 타원 곡선은

$$y^2 + xy = x^3 + x + 1$$

이고,

$$y^2 + xy = x^3 + 1$$

은 anomalous 타원 곡선의 twist 타원 곡선이며, 이 곡선의 trace는 -1이다.

mP를 계산하는데 anomalous 타원곡선이 주는 장점은 φ의 특성 다항식

$$T^2 - T + q = 0$$

에서

$$\begin{aligned} qP &= \varphi(P) - \varphi^2(P) \\ &= (x^q, y^q) - (x^{q^2}, y^{q^2}) \end{aligned} \quad (1)$$

이므로 유한체 F_q에서 정규 기저(normal basis)[5]를 사용하면 qP를 한번의 더하기로 계산할 수 있다. $q = 2$일때

$$2P = \varphi(P) - \varphi^2(P)$$

$$4P = 2(\varphi(P) - \varphi^2(P))$$
$$= -\varphi^3(P) - \varphi^2(P)$$
$$8P = 4(2P) = -\varphi^3(P) + \varphi^5(P)$$
$$16P = 4(4P) = \varphi^4(P) - \varphi^8(P)$$

이므로 한번의 더하기만으로 각각을 계산할 수 있다[2].

그러나, $q = 4$일때 F_4위에서 정의된 anomalous 타원곡선을 E/F_4, E/F_4의 Frobenious map을 φ라 하면 식(1)에 의해

$$4P = \varphi(P) - \varphi^2(P) \qquad (2)$$

이지만

$$8P = 2\varphi(P) - 2\varphi^2(P)$$
$$\Leftarrow 3 \text{ additions}$$
$$16P = 4(4P) = \varphi^2(P) - 2\varphi^3(P) + \varphi^4(P)$$
$$\Leftarrow 3 \text{ additions}$$

가 되어 반복 연산법(repeated-doubling method)과 별 차이가 없음을 알 수 있다. 하지만 식(2)을 이용하면

$$8P = 2(4P) = 2(\varphi(P) - \varphi^2(P))$$
$$\Leftarrow 2 \text{ additions}$$
$$16P = 4(4P) = \varphi(4P) - \varphi^2(4P)$$
$$\Leftarrow 2 \text{ additions}$$

이 되어 계산을 줄일 수 있다. 즉 $16P$ 계산은 보통 $16P = 2(2(2(2P)))$으로 4번의 더하기가 필요하지만, 위의 경우엔 2번으로 계산 가능하다. 그러나, F_2 위에서 anomalous 타원 곡선을 이용하는 경우에는 1번으로 $16P$를 계산할 수 있었으므로 F_4 위에서 정의된 anomalous 타원 곡선을 사용하는 경우에는 연산에 대한 효율이 감소함을 알 수 있다.

3. $Tr(\varphi) = 3$ 인 타원 곡선 찾기

F_4 위에서 정의된 anomalous 타원 곡선은 mP 계산시 F_2 위에서 정의된 anomalous 타원 곡선에서 얻는 장점이 없음을 2절에서 보였다. 하지만, F_4 위에서 정의된 Frobenious map의 trace가 3인 타원 곡선을 사용하면 mP계산시 속도를 줄일 수 있다.

F_4 위에서 정의된 타원 곡선 E/F_4의 Frobenious map φ의 trace $Tr(\varphi)$가 3일때, φ는 특성 다항식

$$T^2 - 3T + 4 = 0 \qquad (3)$$

를 만족한다. $E(F_q)(q = 4^k$, k는 양의 정수)위의 한 점 P를 4번 더하고자 할 때는 식(3)을 이용하여

$$4P = 3\varphi(P) - \varphi^2(P)$$

로 계산할 수 있다. 또한 식(3)으로부터

$$16 = 4 \times 4$$
$$= (3T - T^2)(3T - T^2)$$
$$= (9 - 6T + T^2)T^2$$
$$= (2(3T - T^2) + 1 - 6T + T^2)T^2$$
$$= (-T^2 + 1)T^2 = T^2 - T^4$$

이므로

$$16P = \varphi^2(P) - \varphi^4(P) \qquad (4)$$

로써, 유한체 F_q에서 정규 기저를 사용하면, $16P$ 계산시 한번의 더하기만 필요하다. 이는 F_4위에서 정의된 anomalous 타원 곡선을 사용하는 것보다 mP 계산시 연산의 수가 줄어듦을 알 수 있다.

결론적으로, F_4위에서는 anomalous 타원 곡선보다 $Tr(\varphi) = 3$인 타원 곡선을 이용하는 것이 mP 계산시 효율적이다. 왜냐하면, F_2위에서 정의된 anomalous 타원 곡선과 같이 $16P$를 위한 4번 더하기를 한 번으로 줄였으므로, repeated-16-powering을 사용하여 일반적인 repeated-doubling 방법보다 더하기 횟수를 절약할 수 있기 때문이다.

다음은 F_4위에서 정의된 $Tr(\varphi) = 3$인 타원 곡선을 찾는 방법이다.

F_4위에서 정의된 타원 곡선을 찾기위하여 다음의 정리를 이용한다.

정 리 1 (Weil)[8]

E/F_q는 $F_q(q = p^k$, p는 소수이고 k는 양의 정수)위에서 정의된 타원 곡선이고 $E(F_{q^n})$의 order는

$$|E(F_{q^n})| = 1 + q^n - Tr(\varphi^n)$$

이다. 여기서 φ는 E/F_q에서 정의된 Frobenious map이다.

정 리 2 (nonsupersingular 타원 곡선찾기)[5]
characteristic이 2인 유한체 $F_q(q = 2^k)$위에서 정의된 nonsupersingular 타원 곡선은 다음과 같다.

γ를 F_q에 있는 원소중 $Tr(\gamma) \neq 0$인 것으로 택했을 때

$$E = \{(x, y) \mid y^2 + xy = x^3 + a_2x^2 + a_6\} \cup \{O\}, \quad a_2 \in \{0, \gamma\}, a_6 \in F_q - \{0\} \tag{5}$$

이다. 따라서, $2(q-1)$개의 nonsupersingular 타원 곡선들의 isomorphism class 가 존재한다.

(a_2, a_6)	F_4 − points	$Tr(\varphi)$
$E_1 = (0, 1)$	$(0, 1), (1, 0), (1, 1), (\gamma, 0), (\gamma, \gamma), (\gamma+1, \gamma+1)(\gamma+1, 0)$	−3
$E_2 = (0, \gamma)$	$(0, \gamma+1), (\gamma, 1), (\gamma, \gamma+1)$	1
$E_3 = (0, \gamma+1)$	$(0, \gamma), (\gamma+1, 1), (\gamma+1, \gamma)$	1
$E_4 = (\gamma, 1)$	$(0, 1)$	3
$E_5 = (\gamma, \gamma)$	$(0, \gamma+1), (1, \gamma), (1, \gamma+1), (\gamma+1, 0)(\gamma+1, \gamma+1)$	−1
$E_6 = (\gamma, \gamma+1)$	$(0, \gamma), (1, 0), (1, 1), (\gamma, 1), (\gamma, \gamma+1)$	−1

표 1 : F_4위에서 정의된 nonsupersingular 타원곡선

γ를 $x^2 + x + 1 = 0$의 근이라 하면 $F_4 = \{0, 1, \gamma, \gamma+1\}$이다. 그리고, $Tr(\gamma) = \gamma + \gamma^2 = 1$ 이므로 식(5)에서 a_2는 0 또는 γ이다. a_6는 $\{1, \gamma, \gamma+1\}$중의 하나를 선택한다. 따라서, [표1]과 같이 6개의 F_4 위에서 정의된 nonsupersingular 타원 곡선을 얻을 수 있다.

[표1]에서 각각의 경우에, 정리 1에서 $n = 1$, $q = 4$로 놓고 $Tr(\varphi)$를 얻는다.

결과적으로,

$$E_4 : y^2 + xy = x^3 + \gamma x^2 + 1$$

이 $Tr(\varphi) = 3$이 원하는 타원 곡선이다.

참고로, $E_1 : y^2 + xy = x^3 + 1$은 a_2, a_6가 F_2의 원소이므로, 이것은 F_2 위에서 정의된 곡선이다. E_1의 Frobenious map의 trace는 −1로 anomalous 타원 곡선의 twist타원 곡선이다[2]. 한편, 타원 곡선 선택시 Pohlig-Hellman 알고리듬이나 Shanks의 baby step giant step 공격에 안전하기 위하여 타원 곡선의 order $|E(F_q)|$가 40자리수 이상의 소인수를 갖도록 선택되어야 한다[5]. 따라서, 타원 곡선 E_4를 암호 시스템에 응용하기 위해서는 $|E_4(F_{4^n})|$가 40자리수 이상의 소인수를 갖도록 하는 n을 찾는다.

이를 위하여 다음의 정리들을 이용한다.

정 리 3 [2]
E/F_q는 F_q위에서 정의된 타원 곡선이고, $a_n = Tr(\varphi^n)$일때 $E(F_{q^n})$의 order는

$$|E(F_{q^n})| = q^n + 1 - a_n,$$
$$a_0=2,\ a_1=Tr(\varphi),\ a_n=Tr(\varphi)a_{n-1}-qa_{n-2}.$$

이다.

따름정리 4
$Tr(\varphi) = 3$이고 $q = 4$일때 정리 3에 의하여

$$N_n = |E(F_{4^n})| = 4^n + 1 - a_n,$$
$$a_0=2,\ a_1=3,\ a_n=3a_{n-1}-4a_{n-2}$$

이다.

정 리 5 [2]

n_1이 n을 나눌때, $|E(F_{q^{n_1}})|$은 $|E(F_{q^n})|$을 나눈다.

따름정리 4를 이용하여 $n = 2, 3, \ldots$로 증가하면서 N_n을 구하고 그 중 40자리 이상의 소인수를 갖는 N_n을 택한다. 정리 5에 의하여 n을 소수이거나 작은 소수와 큰 소수의 곱인 수를 선택하면 훨씬 작은 계산으로 원하는 타원 곡선을 찾을 수 있다.

4. F_4 위에서 $Tr(\varphi) = 3$인 타원 곡선 위에서의 mP 계산

Diffie-Hellman 형 암호체계에서는 반복 연산이 알고리듬의 대부분을 이루기 때문에 임의의 정수 m와 타원 곡선 위의 점 P에 대하여 mP를 빨리 계산할 수 있어야 한다.

타원 곡선위의 Frobenious map φ의 trace가 3이면, 즉 $Tr(\varphi) = 3$이면, φ은

$$T^2 - 3T + 4 = 0$$

를 만족하고

$$4 = 3T - T^2$$
$$16 = -T^4 + T^2$$

이므로 $16P = \varphi(P)^2 - \varphi(P)^4$를 한 번의 더하기로 구할 수 있음을 알았다.

주이진 정수 m을

$$m = a_k 16^k + \ldots + a_1 16 + a_0 \qquad (6)$$
$$k = \lfloor \log_{16} m \rfloor,\ a_k \neq 0,$$
$$0 \leq a_i \leq 15,\ i = 0, \ldots, k \qquad (7)$$

라 놓자.

다음은 repeated-16-powering을 이용한 알고리듬이다.

◐ 알고리듬 (1)

step 0 : $k = \lfloor \log_{16} m \rfloor$와 a_k를 계산한다.
step 1 : $Q = a_k P$를 계산한다.
step 2 : For $i = k - 1 \ldots 0$
step 3 : $Q := 16Q$
step 4 : if$(a_i \neq 0) Q := Q + a_i P$
output : Q

Iteration 수는 k번 이므로 k번의 $16Q$ 더하기가 있고 각 loop에서 $a_i \neq 0$의 경우 $a_i P$를 위한 더하기와 $Q + a_i P$를 위한 한 번의 더하기가 있다. $2 \leq a_i \leq 15$인 a_i에 대해 $a_i P$를 사전 계산해 둔다면 step 4에서 한 번의 더하기만 하므로 $\log_{16} m + \log_{16} m = \frac{1}{2} \log_2 m$번으로 mP를 계산할 수 있다. 따라서, 반복 연산만을 고려할 때 평균 $\frac{3}{2} \log_2 m$번의 repeated-doubling 방법보다 3배 빠르게 계산할 수 있다.

사전 계산을 고려하지 않을 경우 가장 많은 더하기 횟수를 구해보자. $a_i = 15$일때, $a_i P$ 계산에서 가장 많이 더하므로(6번) 최대한 $\log_{16} m + (6 + 1)\log_{16} m = 2 \log_2 m$번의 더하기를 해야 한다.

이는 보통의 반복 더하기(repeated-doubling method)와 같아 보이지만, 이 경우에도 식(4)를 이용한 다음의 알고리듬으로 mP 계산시 더하기 횟수를 줄일 수 있다.

◐ 알고리듬 (2)

step 0 : $k = \lfloor \log_{16} m \rfloor$와 a_k를 계산한다.
step 1 : $R = 16P$를 계산한다.
step 2 : $Q = a_k P$를 계산한다.
step 3 : For $i = k - 1 \ldots 0$
step 4 : $Q := 16Q$
step 5 : if$(1 \leq a_i \leq 10) Q := Q + a_i P$
step 6 : else if$(11 \leq a_i \leq 15)$
$Q := Q + R - (16 - a_i) R$
output : Q

알고리듬(2)를 이용하여 다음과 같은 결과를 얻을 수 있다.

정 리 6

$E_4 : y^2 + xy = x^3 + \gamma x + 1$위의 임의의 한 점 $P \in E(F_{4^n})$에 대하여 mP(m: 임

의의 정수)를 계산하는데 $\frac{3}{2}\log_2 m + 1$번 이하의 더하기로 계산할 수 있다. 평균 $\frac{5}{4}\log_2 m + 1$번이다.

증 명

$15P = 16P - P$이므로 보통 반복 더하기 방법시 6번 이었던 것을 두 번의 더하기로 줄일 수 있다. [표2]와 같이 a_iP를 구할때 더하는 횟수는 $a_i = 10$일때 4번으로 가장 많으므로 $11 \le a_i \le 15$이면

$$R - (16 - a_i)P = 16P - b_iP$$
$$= a_iP,\ 1 \le b_i \le 5$$

로 계산한다. 알고리듬(2)의 step 5 또는 step 6에서 최대한 5번, 평균 4번의 더하기가 있고 step 1에서 초기화를 위해서 한 번의 더하기가 있으므로

최대한 $\log_{16} m + (4 + 1)\log_{16} m + 1$
$$= \frac{3}{2}\log_2 m + 1,$$

평 균 $\log_{16} m + (3 + 1)\log_{16} m + 1$
$$= \frac{5}{4}\log_2 m + 1$$

번의 더하기로 mP를 구할 수 있다. □

a_i	b_i	number of additions		
		a_iP	b_iP	$16P - b_iP$
8	8	3	3	4
9	7	4	4	5
10	6	4	3	4
11	5	5	3	4
12	4	4	2	3
13	3	5	2	3
14	2	5	1	2
15	1	6	0	1

표2 : a_iP, b_iP, $16P - b_iP$를 구하기 위한 더하기 수 ($b_i = 16 - a_i$)

결국, mP 계산시 보통 반복 더하기 방법을 쓰면 m의 hamming weight가 가장 큰 경우 $2\log_2 m$번이 필요하다. 그러나, 알고리듬(2)를 사용하여 보통 반복 더하기 방법의 3/4 이하로 mP를 계산할 수 있다.

알고리듬(2)에서도 알고리듬(1)과 같이 $2 \le a_i \le 15$인 모든 수 a_i에 대해 a_iP를 사전 계산(pre-computation)해 놓았다면 step 4에서 계산으로 처리할 수 있으므로 $\log_{16} m + \log_{16} m = \frac{1}{2}\log_2 m$ 번 이하의 더하기로 계산할 수 있다.

5. 결 론

타원 곡선을 이용한 암호 시스템이 높은 안전성을 갖는 장점이 있지만 연산 속도가 유한체에서의 연산속도보다 느리기 때문에 실용화가 되지 못하고 있다. 연산 속도를 줄이는 것도 하나의 해결방법이겠지만 암호 알고리듬들이 반복 연산을 많이 사용하기 때문에 주어진 임의의 수 m과 곡선위의 임의의 점 P에 대해 mP를 적은 횟수로 계산할 방법을 찾을 수도 있다. Koblitz가 Frobenious map φ의 trace가 1인 경우의 타원 곡선을 사용할 것을 제안했다. 특히, F_2위에서

$$y^2 + xy = x^3 + x + 1,$$
$$y^2 + xy = x^3 + 1$$

사용시, 이 곡선 위에서 반복 연산을 효과적으로 할 수 있음을 밝혔다[2].

그런데, F_4 위에서 φ의 trace $Tr(\varphi)$가 3인 타원 곡선이 F_4 위에서 anomalous 타원 곡선보다 반복 연산을 효과적으로 할 수 있음을 알았다. 또한, 이를 이용하여 암호 시스템에 안전한 타원 곡선의 선택법 및 효율적 연산법에 대하여 설명하였다. 즉,

$$E(F_{4^n}) = \{(x, y) \in F_{4^n} \times F_{4^n} \mid y^2 + xy$$
$$= x^3 + \gamma x + 1, \gamma \in F_4\} \cup \{O\}$$

가 $|E(F_{4^n})|$이 40자리수 이상의 소인수를 갖도록 n을 선택한 후 $E(F_{4^n})$ 내의 한 점 P를 m번 더할 때 가장 많은 경우 $\frac{3}{4}\log_2 m + 1$번으로 계산 가능하

여 $2\log_2 m$ 번인 보통의 반복 연산 방법(repeated-doubling method)의 3/4임을 밝혔다.

앞으로, 실제 컴퓨터 구현을 통하여 어떤 n에 대하여 $E(F_4)$를 확장할때 $E(F_{4^n})$이 암호 시스템에 응용하기 좋은지, 즉 이 군의 order가 40자리 수 이상의 소인수를 갖는지를 찾는 작업을 할 것이다. 그리고, 유한체의 characteristic이 그 위에서 정의된 타원 곡선의 Frobenious map φ의 trace와 어떤 관계에 있을 때 효율적인 반복 연산이 가능한가를 밝히는 것은 흥미로운 일이 될 것이다. 또한 그러한 관계를 만족하는 타원 곡선을 얼마나 찾기 쉬운가도 같이 연구되어야 할 것이다.

참 고 문 헌

[1] N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.*, **48**, pp.203-209, 1987

[2] N. Koblitz, CM-curves with good cryptographic properties, *Advances in Cryptology*, **3**, pp.187-199, 1991.

[3] N. Koblitz, Elliptic Curve Implementation of Zero-Knowledge Blobs, *Journal of Cryptology*, **4**, pp.207-213, 1991.

[4] A. K. Lenstra and H. W. Lenstra Jr., Algorithms in number theory, in: *Handbook of Theoretical Science*, Vol. **A**, *Algorithms and Complexity*, ed. by J. Van Leeuwen, Amsterdam: Elsevier, pp.673-715, 1990.

[5] A. Menezes, *Elliptic Curve Pulic key Cryptosystems*, Kluwer Academic Publishers, 1993.

[6] A. Menzes, T. Okamoto, and S. A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *Proceedings of the 23rd ACM Symp. Theory of Computing*, 1991.

[7] V.Miller, Uses of elliptic curves in cyrptography, *advances in cryptology-Crypto '85*, Lecture notes in computer science, **48**, pp.417-426, 1986.

[8] J. H. Silverman, *The Arithmetic of Elliptic curves*, Spriger-Verlag, 1986.

□ 著者紹介

이 은 정

1993년 2월 포항공과대학교 수학과
1993년 3월 ~ 1995년 2월 포항공과대학교 대학원 수학과 석사과정
1995년 1월 ~ 현재 포항공과대학교 정보통신연구소 전임연구원

최 영 주 (종신회원)

1982년 2월 이화여자대학교 이학사
1986년 5월 Temple 대학교 이학박사
1986년 5월 ~ 1988년 8월 Ohio 주립대학교 강사
1988년 9월 ~ 1990년 1월 Maryland 대학교 조교수(방문)
1989년 9월 ~ 1990년 1월 Colorado 대학교 조교수
1990년 2월 ~ 현재 포항공과대학교 부교수