情報保護學會論文誌
第12券 第5號, 2002. 10

작은 유한체 위에 정의된 타원곡선의 고속연산 방법*

박 영 호**, 정 수 환***

A Fast Multiplication Method for Elliptic Curves defined on small finite fields

Young-Ho Park**, Souhwan Jung***

요 약

Koblitz 타원곡선과 같이 표수(characteristic)가 2인 작은 유한체 위에서 정의된 non-supersingular 타원곡선은 스칼라 곱을 효율적으로 구현하기 위하여 프로베니우스 자기준동형 (Frobenius endomorphism)이 유용하게 사용된다. 본 논문은 확장된 프로베니우스 함수를 사용하여 스칼라 곱의 고속연산을 가능하게 하는 방법을 소개한다. 이 방법은 Müller[5]가 제안한 블록방법(block method) 보다 선행계산을 위해 사용되는 덧셈량을 줄이는 반면에 확장길이는 거의 같게 하므로 Müller의 방법보다 효율적이다.

ABSTRACT

As Koblitz curve, the Frobenius endomorphism is known to be useful in efficient implementation of multiplication on non-supersingular elliptic curves defined on small finite fields of characteristic two. In this paper a method using the extended Frobenius endomorphism to speed up scalar multiplication is introduced. It will be shown that the proposed method is more efficient than Müller's block method in [5] because the number of point addition for precomputation is small but on the other hand the expansion length is almost same.

Keyword : *Elliptic Curve, Scalar Multiplication, Frobenius Endomorphism*

Ⅰ. 서 론

타원곡선 암호시스템의 효율적인 구현을 위하여 Koblitz 곡선[3] 또는 작은 표수의 유한체 위에서 정의된 타원곡선들의 사용이 제안되었다[5,8]. 최근 PKC02[6]에서 표수(characteristic)가 2인 작은 유한체 위에서 정의된 non-supersingular 타원곡선중 작은 cofactor를 갖는 암호학적으로 효율적인 곡선들의 사용을 제안하였다. 이들 곡선들은 타원곡선 연산 중 두 배하는 연산(doubling) 대신에 효율적으로 계산가능한 프로베니우스 자기준동형을 이용한 고속연산을 가능하게 하는 방법을 사용한다. 즉, 스칼라 곱 연산시 스칼라 정수를 이진 전개하는 대신에 프로베니우스를 사용하여 확장전개하여 고속연산을 가능하게 한다. 일반적으로 이러한 곡선의 연산속도는 프로베니우스 확장길이와 밀접한 관계가 있을 뿐만 아니라 binary method, window method[1]와 같은 프로베니우스 확장전개를 적용하는 방법에 따라 달라진다.

Müller는 연산속도를 향상시키기 위해 스칼라 정수

* 이 논문은 2001년도 한국학술진흥재단 협동연구지원사업지원에 의하여 연구되었음.(KRF-2001-042-E00045)
** 세종사이버대학교 정보보호시스템공학과(youngho@cybersejong.ac.kr)
*** 숭실대학교 정보통신전자공학부(souhwanj@ssu.ac.kr)

를 프로베니우스 함수로 확장전개 후 block 단위로 나누어 선계산을 사용하는 방법을 제안하였다[5]. 만일 타원곡선 $E(F_q)$에서 임의의 점에 대한 스칼라 곱셈을 할 때 block을 w로 잡으면 프로베니우스의 확장길이는 $1/w$ 배 줄어드는 반면에 매번 $((q+1)^w-1)/2$개 점들의 선계산을 해야한다. 따라서 유한체와 타원곡선의 위수 등의 파라미터들의 환경에 따라 효과적인 연산 방법이 될 수도 있다.

본 논문에서는 정수를 프로베니우스 자기준동형 Φ으로 확장전개하는 방법대신 적당히 작은 w에 대해 프로베니우스의 w 제곱승인 Φ^w로 전개하므로 Müller의 방법에 비해 확장길이는 거의 같고 선계산량을 줄이는 새로운 방법을 제안한다. 이 방법은 유한체 F_{q^w}이 F_q에서 정규기저(normal basis)로 구성되었을 경우 프로베니우스 Φ가 한번 쉬프트연산이라면 Φ^w도 w번 쉬프트 하는 연산이므로 Φ 처럼 쉽게 계산 가능하다는 것에 기반하였다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 표수 2인 유한체 위에 정의된 타원곡선의 일반적인 성질과 프로베니우스 자기준동형의 확장방법을 소개한다. 3절에서 확장된 프로베니우스 함수를 사용하여 확장길이를 줄이는 방법을 제안하고 효율적인 알고리즘을 제시한다. 4절에서 제안된 방법의 선계산 양을 Müller의 방법과 비교하여 분석하고 5절에서 결론을 맺는다.

II. 타원곡선과 프로베니우스 확장

암호학적 응용의 관점에서, 표수가 2인 유한체 위에서 정의된 non-supersingular 타원곡선들이 많은 관심을 끌어왔으므로 본 논문에서는 표수 2인 유한체 위에서 정의된 곡선들만 고려할 것이다. F_q는 q개의 원소를 갖는 표수 2인 유한체라 하고 $q=2^s(1\le s\le 5)$ 라 하자. 또한 $\overline{F_q}$를 F_q의 대수적 폐포(algebraic closure)라 하자. 다음과 같은 형태의 Weierstrass 방정식에 의해 주어진 non-supersingular 타원곡선 $E(F_q)$를 표현하면

$$y^2+xy=x^3+a_2x^2+a_6 \tag{1}$$

이다. 여기서 a_2, a_6는 F_q의 원소이고, $a_6\neq 0$이다. $E(F_q)$의 q-지수승 프로베니우스 자기준동형(Frobenius endomorphsim)은 다음과 같이 정의된다:

$$\Phi: E(\overline{F_q}) \to E(\overline{F_q}),\ (x,y)\mapsto (x^q, y^q).$$

유명한 Hasse 정리로부터, $E(F_q)$의 위수는

$$\#E(F_q)=q+1-t$$

로 정수 t와 밀접하게 관련된다. 여기서 t는 다음식 (2)을 만족하는 프로베니우스 자기준동형 Φ의 자취(trace)이다:

$$\Phi^2-t\Phi+q=0 \tag{2}$$

여기서 t는 non-supersingular 타원곡선 $E(F_q)$에 대해서는 홀수여야 함에 주의하자.

이제 작은 정수 $w\ge 1$에 대하여 $\Psi=\Phi^w$라 하자. 다시 Hasse-Weil 정리에 의해 다음을 만족한다:

$$\Psi^2-t_w\Psi+q^w=0 \tag{3}$$

여기서 $t_0=2$, $t_1=t$ 그리고 $l\ge 2$에 대하여,

$$t_l=t_1t_{l-1}-qt_{l-2}$$

이다. 그러면 함수 $\Psi=\Phi^w$는

$$\Psi: E(\overline{F_q}) \to E(\overline{F_q}),\ (x,y)\mapsto (x^{q^w}, y^{q^w})$$

이므로 타원곡선 E의 정의체가 F_{q^w}으로 간주하면 Ψ는 $E(F_{q^w})$에서의 프로베니우스 함수로 볼 수 있으므로 기존의 방식[2,4~6]의 프로베니우스 전개방식이 그대로 사용될 수 있다. 즉, Ψ의 기약다항식 (3)에 의해, 임의의 정수 m을 $m\in Z[\Psi]$로 간주하여 Ψ로 다음과 같이 확장전개할 수 있다. ([2,4,5,6] 참조).

[정리 1]

$q^w\ge 4$에 대하여, 임의의 $\rho\in Z[\Psi]$는 $\rho=\sum_{i=0}^{k} r_i\Psi^i$로 표현되어진다. 여기서 정수계수 $r_i\in\{-q^w/2+1,\cdots,q^w/2\}$이고,
$k\le \lceil \frac{1}{w}\log_q N_{Z[\Psi]/Z}(\rho)\rceil+3$이다.

다음은 정리 1의 결과를 얻기 위한 임의의 $\rho=a_1+$

$a_2\Psi \in Z[\Psi]$에 대하여 Ψ의 확장을 효율적으로 계산하는 알고리즘 제시한다.

【 알고리즘 1 】

입력 : $\rho = a_1 + a_2\Psi \in Z[\Psi]$.

출력 : $\rho = \sum_{i=0}^{k} r_i \Psi^i$를 만족하는 정수 r_i들.

1. $x = a_1$, $y = a_2$, $i = 0$로 놓는다.
2. $|x| > q^w/2$ 또는 $|y| > q^w/2$를 만족하는 동안 다음을 실행한다.
   (a) $x_1 \equiv x \pmod{q^w}$를 계산한다.
   (b) $r_i = \begin{cases} x & \text{if } x \le q^w/2, \\ x - q^w & \text{otherwise.} \end{cases}$로 둔다.
   (c) $h = (r_i - x)/q^w$, $x = y - th$, $y = h$, 그리고 $i = i+1$로 둔다.
3. $r_i = x$, $r_{i+1} = y$
4. r_i를 결과로 보낸다.

〔알고리즘 1〕

〔알고리즘 1〕은 Müller가 제시한 알고리즘과 거의 같으며 단지 단계 3에서 Ψ-확장의 계수 r_i는 $\pm q/2$까지 허용하는 것이 다르다. 하지만 ρ에 대하여 Ψ-확장이 유일하게 표현되지 않아도 되므로 아무런 문제가 발생하지 않으며 Müller의 알고리즘보다 확장길이를 조금이라도 줄일 수 있게끔 한다.

Ⅲ. Ψ확장을 이용한 곱셈연산

본 절에서는 정수 m을 Ψ로 확장전개시킬 때 확장길이를 축소시키는 방법을 살펴본다. 공개키 암호시스템에서는 $E(F_{q^n})$의 위수가 최소한 160비트 정도 길이의 큰 소수 인수 p을 가질 것을 요구한다[7]. 따라서 타원곡선 $E(F_{q^n})$가 큰 소수 위수 p인 점 P를 갖는다 하자. 타원곡선 군의 위수를 $\#E(F_{q^n})$로 표시하면 $\#E(F_{q^n}) = hp$로 나타낼 수 있고, 이 때 h를 $E(F_{q^n})$의 cofactor라 부른다. 또한 Hasse 정리에 의해 타원곡선의 위수는 $\Phi^n - 1$의 노름(Norm), 즉,

$$\#E(F_{q^n}) = N_{Z[\Phi]/Z}(\Phi^n - 1)$$

을 만족한다. 일반적으로, 타원곡선의 스칼라 곱은 큰 정수 $m \approx q^n$에 대하여 mP 계산을 해야한다. 정리 1에 따르면, m의 프로베니우스 확장길이는 m의 노름에 비례한다. 프로베니우스 확장길이를 줄이기 위해 Smart[8], Solinas[4]는 정수 m을 $Z[\Phi]$의 원소로 보아 m을 $\Phi^n - 1$로 나누어 나머지 $\rho \in Z[\Phi]$를 얻고 m대신에 ρ를 사용하여 프로베니우스 전개를 하는 방법을 제안했다. 이때 나머지 ρ은 m의 노름에 절반정도인 q^{n+1} 의 크기의 노름을 갖고 따라서 프로베니우스 확장길이를 약 50%나 줄일 수 있다. 또한 최근 PKC02[6]에서는 노름이 p인 원소 α를 사용하여 m을 나눈 나머지 ρ를 가지고 프로베니우스 확장길이를 최적화하여 줄이는 방법과 알고리즘을 제시하였다.

하지만 이 방법은 프로베니우스 Φ로 전개하는 것 대신에 $\Psi = \Phi^w$로 전개할 때에는 직접 적용할 수 가 없다. 왜냐하면 일반적으로 $\alpha \in Z[\Phi]$ 이지만 $\alpha \notin Z[\Psi]$ 이기 때문에 m을 α로 나눈 나머지 ρ는 $Z[\Phi]$의 원소이지만 $Z[\Psi]$의 원소라고 할 수가 없다. 따라서 알고리즘 1을 적용할 수 가 없다. 그러므로 이 문제를 해결하기 위한 방법으로 다음과 같은 정리들이 필요하다.

[보조정리 2]

작은 정수 $w \ge 1$에 대하여 $\Psi = \Phi^w$라하고 $\Phi^2 - t\Phi + q = 0$을 만족한다 하자. 정수 c_w, d_w에 대해서 $\Psi = d_w + c_w\Phi$라 하면

$$c_w = c_{w-1} \cdot t + d_{w-1} \tag{4}$$

$$d_w = -q \cdot c_{w-1} \quad (w \ge 2) \tag{5}$$

여기서 $c_1 = 1, d_1 = 0$ 이다. 또한 $c_w \le q^w/2$ 을 만족한다.

[증명]

$\Phi^2 = t\Phi - q$로 부터 $\Psi = \Phi^w = \Phi\Phi^{w-1} = \Phi(d_{w-1} + c_{w-1}\Phi) = d_{w-1}\Phi + c_{w-1}\Phi^2 = -qc_{w-1} + (c_{w-1}t + d_{w-1})\Phi$

을 얻고 이 식으로부터 첫 번째 식 (4)와 (5)를 얻는다. 이제 마지막 식을 증명하자. 만일 $w = 1, 2$ 경우 Hasse 정리에 의해 쉽게 증명할 수 있다. $w \ge 3$에 대하여 수학적 귀납법을 이용하여 증명하자. 먼저 w보다 작은 경우 마지막 식이 만족한다 가정하자. 그러면

$$|c_w| = |c_{w-1} \cdot t + d_{w-1}| \le |t \cdot c_{w-1}| + |d_{w-1}|$$
$$\le |t| q^{w-1}/2 + q|c_{w-2}| \le |t| q^{w-1}/2 + q q^{w-2}/2$$
$$\le (|t|+1) q^{w-1}/2$$

을 갖는다. t가 홀수이고 $|t| \le 2\sqrt{q}$이므로

$$|t| \le \begin{cases} 1 & \text{if } q=2, \\ 3 & \text{if } q=4, \\ q-1 & \text{if } q \ge 2^3. \end{cases}$$

을 갖는다. 그러므로 $|t+1| \le q$ 이므로 $|c_w| \le q^{w/2}$만족하고 증명이 완성된다. □

[정리 3]

작은 정수 $w \ge 1$에 대하여 $\Psi = \Phi^w = d + c\Phi$라하고 임의의 원소 $\rho = a + b\Phi \neq 0 \in Z[\Phi]$로 놓자. 그러면 다음을 만족하는 정수 x, y, z가 존재한다:

1. $\rho = (x + y\Psi) + z\Phi$, $-q^w/2 < z < q^w/2$.
2. $N_{Z[\Phi]/Z}(x + y\Psi) \le \mu N_{Z[\Phi]/Z}(\rho)$, $\mu \le q^{2w+1}$.
3. 만일 $N_{Z[\Phi]/Z}(\rho) \ge 2^{2w+1}$이면 $N_{Z[\Phi]/Z}(x + y\Psi) \le 2N_{Z[\Phi]/Z}(\rho)$ 이다.

[증명]

$T = Tr_{Z[\Phi]/Z}(\Phi/\rho)$로 놓자. 먼저 다음 조건과 $r_2 = sc + z$를 만족하는 $s, z \in Z$를 계산한다:

$$\begin{cases} 0 \le z < |c| & \text{if } T \ge 0, \\ -|c| < z \le 0 & \text{if } T < 0. \end{cases} \quad (6)$$

그러면 $\rho = r_1 + r_2\Phi = r_1 + (sc+z)\Phi = s(\Psi - d) + r_1 + z\Phi = s\Psi + r_1 - sd + z\Phi$이다. $x = r_1 - sd$와 $y = s$로 놓으면 보조정리 2에 의해 $\rho = (x + y\Psi) + z\Phi$ 와 $|d| \le q^w/2$를 만족한다. 이제 두 번째 부분을 증명하자. $K = Q(\Phi)$라 하자. $(x + y\Psi)/\rho = 1 - z(\Phi/\rho)$ 이므로

$$N_{K/Q}((x + y\Psi)/\rho) = N_{K/Q}(1 - z(\Phi/\rho))$$
$$= 1 - zTr_{K/Q}(\Phi/\rho) + z^2 N_{K/Q}(\Phi/\rho)$$

이다. 식 (6)에 의하여 $zT \ge 0$이므로

$$N_{K/Q}((x + y\Psi)/\rho) \le 1 + z^2 q/N_{Z[\Phi]/Z}(\rho)$$
$$\le 1 + z^2 q \le q^{2w+1}$$

이다. 마지막 부분은 두 번째 부분으로부터 쉽게 유도된다. □

정리 3은 정수 m을 α로 나눈 나머지 ρ가 $\rho \in Z[\Psi]$임을 제공하는 것이 아니고 $N_{Z[\Phi]/Z}(x + y\Psi) \le \mu N_{Z[\Phi]/Z}(\rho)$를 만족하는 $\rho = (x + y\Psi) + z\Phi$로 표현될 수 있음을 나타낸다. 또한, w가 작은 값이므로 대개 모든 나머지 ρ들의 노름 값은 q^{2w+1}보다 크다. 따라서 우리는 $N_{Z[\Phi]/Z}(x + y\Psi) \approx N_{Z[\Phi]/Z}(\rho)$ 임을 얻고 이것은 m의 Ψ로 확장전개길이나 Φ로 확장전개한 길이가 거의 같음을 알 수 있다. 특별히, $c_w = 1$인 경우 $z = 0$ 이고 $x + y\Psi = \rho$ 이다.

마침내, 정리 3에 따라 우리는 mP의 계산을 다음과 같이 시행한다 :

$$mP = \rho P = ((x + y\Psi) + z\Phi)P = (x + y\Psi)P + \Phi(zP)$$

여기서 $(x + y\Psi)P$를 계산시 $x + y\Psi \in Z[\Psi]$이므로 알고리즘 1을 사용하여 Ψ로 확장전개할 수 있다. 또한 $\Phi(zP)$의 계산은 $-q^w/2 < z < q^w/2$이므로 zP가 선계산에 포함된다. 그러므로 mP의 총 계산량은 Ψ 확장전개를 사용한 $x + y\Psi$의 계산에다 많아야 타원곡선 딧셈 하나만을 추가하면 충분하다.

다음은 정수 m에 대해 노름이 p인 원소 α ($N_{Z[\Phi]/Z}(\alpha) = p$)를 사용하여 m을 나눈 나머지 ρ를 구하는 알고리즘 $2^{[6,9]}$와 정리 3을 사용하여 mP를 계산하는 알고리즘 3을 제시한다.

알고리즘 2 (m를 $\alpha = a + b\Phi$로 나누기)

입력 : $m \in Z$, $\alpha = a + b\Phi$
출력 : $\rho = r_1 + r_2\Phi$

I 사전계산

1. $N_\alpha = N_{Z[\Phi]/Z}(\alpha)$, $c = -\lfloor t/2 \rceil$.
2. $\Phi' = \Phi + c$, $N = N_{Z[\Phi]/Z}(\Phi')$.
3. $a_1 = a - bc$, $b_1 = b$. ($\alpha = a_1 + b_1\Phi'$)

II 본 알고리즘

4. $x_1 = m(a_1 + b_1)$ 그리고 $x_2 = -mb_1$.
5. $y_i = \lfloor \frac{x_i}{N_\alpha} \rceil$ $(i = 1, 2)$.
6. $r'_1 = m - (a_1 y_1 - Nb_1 y_2)$, $r'_2 = -(a_1 y_2 + b_1 y_1 + b_1 y_2)$.
7. $r_1 = (r'_1 + r'_2 c)$, $r_2 = r'_2$..
8. r_1, r_2 결과를 보낸다.

[알고리즘2]

알고리즘 3 (mP 계산)

입력 : $m \in Z$, w (small), 위수가 p인 점 P.
출력 : mP

I $\Psi = \Phi^w$**사전계산**
1. $i \cdot P$ ($1 \le i \le q^w/2$) 를 계산 저장한다.
2. $\Psi = d + c\Phi$를 만족하는 d와 c를 구한다 (보조정리 2).

II Ψ **- 확장전개**
3. [알고리즘 2]를 사용하여 나머지 $\rho = r_1 + r_2\Phi$계산.
4. $T = Tr_{Q(\Phi)/Q}(\Phi/\rho)$ 계산.
5. $r_2 = sc + z$, $\begin{cases} 0 \le z < |c| & \text{if } T \ge 0, \\ -|c| < z \le 0 & \text{if } T < 0. \end{cases}$ 을 만족하는 $s, z \in Z$를 계산.
6. $x = r_1 - sd$와 $y = s$.
7. 알고리즘 1을 사용하여 $x + y\Psi$를 확장전개.
8. $mP = (x + y\Psi)P + \Phi(zP)$ 계산.

[알고리즘 3]

Ⅳ. 선계산량

3절에서 우리는 mP의 계산을 프로베니우스 Φ의 전개를 이용하지 않고 적당히 작은 정수 w에 대한 $\Psi = \Phi^w$의 확장전개를 이용한 고속곱셈 연산 방법을 소개했다. 본 절에서는 Müller의 Block 방법에 필요한 선계산량과 제안된 방법의 선계산량을 비교할 것이다.

프로베니우스 Φ의 확장방법을 이용하여 mP를 계산하는 방법은 다음과 같다[5] :

$$mP = \rho P = \sum_{j=0}^{k} r_j \Phi^j(P) \quad (7)$$

여기서 $-q/2+1 \le r_j \le q/2$ 이며 정리 1에서 $w=1$이므로 확장길이는 $k \le \lceil \log_q N_{Z[\Phi]/Z}(\rho) \rceil + 3$이다. 고속연산을 위해 Müller는 (7)식에 Block 방법을 다음과 같이 적용하였다. 만일 블록의 사이즈를

$w \ge 1$로 잡는다면

$$mP = \sum_{i=0}^{\lfloor k/w \rfloor} \Phi^{wi}\left(\sum_{j=0}^{w-1} r_{wi+j} \Phi^j(P) \right)$$

로 표현할 수 있다. 따라서 대괄호 안의 $\left(\sum_{j=0}^{w-1} r_{wi+j} \Phi^j(P) \right)$를 선계산하여 저장한 후 $\lfloor k/w \rfloor$ 개의 타원곡선 덧셈을 하여 mP를 계산한다. 이 때 $-q/2+1 \le r_j \le q/2$ 이므로 $(q+1)$개의 r_jP를 w개 만큼 더해야 하므로 $(q+1)^w$개를 개산해야 한다. 이 계산 중에 모든 $r_j = 0$ 경우와 $-Q$는 Q로부터 쉽게 구할 수 있으므로 이 사실들을 고려한다면 실질적으로 $((q+1)^w - 1)/2$ 점을 선계산하여 저장해야 한다 ([5, 정리3]). 이 방법은 블록 사이즈 w가 커지면 상당히 많은 양을 선계산해야 하므로 효율적이지 못하다. 그러므로 Müller의 Block 방법은 고정된 점의 스탈라곱이나 아주 작은 블록사이즈 경우에만 실질적인 적용이 가능하다.

우리의 방법은 적당한 블록사이즈 w를 미리 결정한 다음 정리 1과 3 그리고 알고리즘 1,2,3을 사용하여 다음과 같이 계산한다.

$$mP = \rho P = \sum_{j=0}^{k} r_j \Psi^j(P) + \Phi(zP)$$

여기서 $-q^w/2+1 \le r_j \le q^w/2$ 이다. 따라서 제안된 방법은 Müller의 Block 방법과 같은 기술을 적용하지 않지만 프로베니우스 Φ를 $\Psi = \Phi^w$로 확장하여 전개하므로 Müller의 Block 방법과 유사한 효과를 얻을 수 있다. 이 경우 선계산 하여 저장할 양은 정리 1에 의해 $q^w/2$점들이다. 따라서 이 방법은 Müller의 $((q+1)^w - 1)/2$ 의 선계산해야 할 점을 $q^w/2$개의 점으로 줄여준다. 여기서의 선계산은 mP를 계산하기 위한 임의의 점 P가 주어질 때마다 계산되어져야 하므로 선계산의 양을 줄이는 것은 고속연산을 가능케 하는 중요한 요소가 된다. 그 뿐만 아니라 모바일이나 스마트 카드와 같이 대폭역과 저장능력이 낮은 환경에서는 더욱더 중요하다.

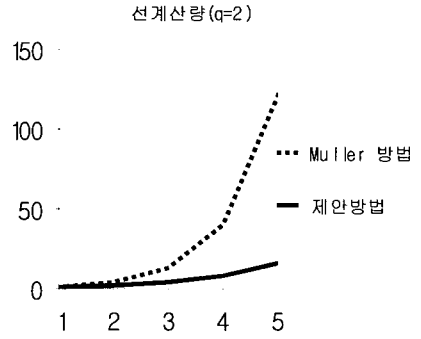
다음 [표 1]과 [그래프 1,2,3]들은 작은 q와 w에 대해서 Müller 방법과 우리의 방법의 선계산해야할 양을 비교한 것이다.

[표 1] 선계산양 비교

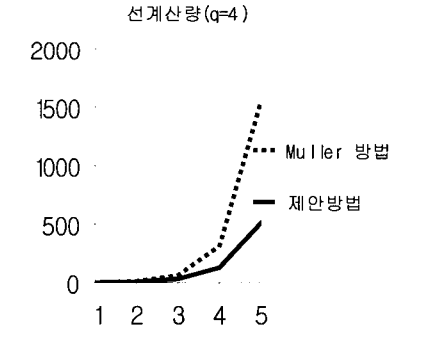
F_q	방법	$w=1$	$w=2$	$w=3$
$q=2$	Müller 방법	1	4	13
	제안방법	1	2	4
$q=4$	Müller 방법	2	12	62
	제안방법	2	8	32
$q=8$	Müller 방법	4	40	364
	제안방법	4	32	256

〔표1〕 선계산양 비교(계속)

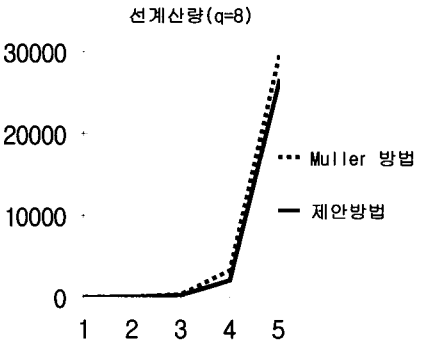
F_q	방법	$w=4$	$w=5$
$q=2$	Müller 방법	40	121
	제안방법	8	16
$q=4$	Müller 방법	312	1562
	제안방법	128	512
$q=8$	Müller 방법	3280	29524
	제안방법	2048	16384



〔그래프 1〕



〔그래프 2〕



〔그래프 3〕

V. 결 론

본 논문에서는 표수(characteristic)가 2인 작은 유한체 위에서 정의된 non-supersingular 타원곡선에서 스칼라인 정수를 프로베니우스 자기준동형 Φ으로 확장전개하여 적당한 블록사이즈 w에 대해 블록방법을 사용하는 Müller의 방법대신에 프로베니우스의 w 제곱승인 $\Psi=\Phi^w$로 전개하므로서 Müller의 방법에 비해 확장길이는 거의 같고 선계산량을 $((q+1)^w-1)/2$에서 $q^w/2$ 으로 줄이는 새로운 방법을 제안하여 스칼라 곱의 고속연산을 좀더 가능하게 하였다.

참 고 문 헌

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

〔2〕 J. Cheon, S. Park, C. Park, and S. Hahn, "Scalar Multiplication on Elliptic Curves by Frobenius Expansions", ETRI Journal. Vol. 32, No. 1, March 1999, pp. 27~38.

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

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

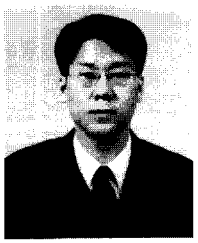
〔5〕 V. Müller, "Fast multiplication in elliptic curves over small fields of characteristic two", Journal of Cryptology, 1998, pp. 219~234.

〔6〕 Young-Ho Park, Sangho Oh, Sangjin Lee, Jongin Lim, and Maenghee Sung, "An improved method of multiplication on certain elliptic curves", Public Key Cryptography, PKC 2002, LNCS 2274, Springer-Verlag, pp. 310~322, 2002.

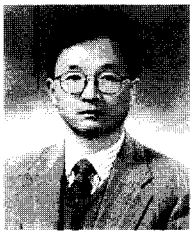
〔7〕 S. Pohlig, M. Hellman, "An improved algorithm for computing logarithms over

GF(p) its cryptographic significance," IEEE Trans. Inform. Theory, 24, pp 106~110, 1978.
[8] N. Smart, "Elliptic curve cryptosystems over small fields of odd characteristic", Journal of Cryptology, 1999, pp. 141~145.
[9] 박영호,한동국,오상호,이상진,임종인,주학수,"타원곡선에서 스칼라 곱의 고속연산", 한국정보보호학회 논문지", Vol. 12, No. 2, pp. 3~10. April 2002.

〈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~2002년 2월 : 고려대 정보보호기술연구센터(CIST) 객원조교수
2002년 2월~현재 : 세종사이버대학교 정보보호시스템공학과 조교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜



정 수 환 (Souhwan Jung)
1985년 서울대학교 전자공학과(학사)
1987년 서울대학교 대학원 전자공학과(석사)
1996년 미국 University of Washington(Ph.D.)
1988년~1991년 한국통신 연구개발단
1996년~1997년 미국 Stellar One Corp.
1997년~현재 : 숭실대학교 정보통신전자공학부 전임강사/조교수
〈관심분야〉 VoIP 보안, 네트워크 프로토콜 보안, 사용자 인증