情報保護學會論文誌
第12券 第1號, 2002. 2

홀수 표수 확장체위의 타원곡선 고속연산

김 용 호*, 박 영 호*, 이 상 진*, 황 정 연*, 김 창 한**, 임 종 인*

An improved method of scalar multiplication on Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic

Yong Ho Kim*, Young Ho Park*, Sangjin Lee*,
Jung Yeon Hwang*, Chang Han Kim**, Jongin Lim*

요 약

작은 홀수 표수를 갖는 유한체 위에 정의된 타원곡선에서 스칼라 곱을 효율적으로 구현하기 위해 프로베니우스 자기준동형(Frobenius endomorphism)이 유용하게 사용된다. 본 논문은 이러한 타원곡선에서 스칼라 곱 연산속도를 향상시키는 새로운 방법을 소개한다. 이 방법은 스칼라의 프로베니우스 자기준동형 확장길이를 기존의 것보다 줄이므로 속도개선을 얻는다.

ABSTRACT

For efficient implementation of scalar multiplication in Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic, Frobenius endomorphism is useful. We discuss new algorithm for multiplying points on Elliptic Curve Cryptosystems over Small Fields. Our algorithm can reduce more the length of the Frobenius expansion than that of Smart.

keyword : *Elliptic Curve, scalar multiplication, Frobenius expansion*

Ⅰ. 서 론

타원곡선 공개키 암호법은 이산대수문제(DLP)를 기반으로 하는 다양한 암호 시스템에 효율적으로 적용 가능하므로 타원곡선 암호시스템 구현에서 효율성을 높이는 연구는 중요한 문제이다. 특히, 타원곡선 위에서 스칼라 곱 연산을 효율적으로 하는 다양한 방법들이 연구되어 왔다. 처음에 Koblitz[3]는 anomalous 이진 타원곡선에서 두 배 연산을 대신하여 프로베니우스 자기준동형을 사용하였다. 그리고 Müller[5]는 이 방법을 확장하여 표수(characteristic)가 2인 작은 유한체 위에서 정의된 타원곡선에 적용하였다. 또한 Smart[10]는 이 방법이 홀수 표수를 갖는 확장체위의 타원곡선에서도 적용 가능함을 보이고, 타원곡선의 위수를 노름(Norm)으로 갖는 원소로 나누는 방법을 사용하여 스칼라의 프로베니우스 확장길이를 반으로 감소시키므로 스칼라 곱의 고속연산을 가능하게 하였다. 최근에, [6]에서 표수(characteristic)가 2인 작은 유한체 위에서 정의된 타원곡선에서 타원곡선의 위수를 노름(Norm)으로 갖는 원소대신 큰 소수 위수를 노름으로 갖는 원소를 사용하여 프로베니우스 확장길이를 최소화시키는 방법이 제안되었다.

따라서 본 논문에서는 표수가 홀수인 타원곡선에서

* 고려대학교 정보보호기술연구센터(CIST)({ kyh, youngho, videmot, sangjin, jilim}@cist.korea.ac.kr)
** 세명대학교 컴퓨터수리정보학과(CHKIM235@chollian.net)

[6]의 방법을 적용하여 Smart의 방법을 개선하고 스칼라 곱의 프로베니우스 확장길이를 보다 더 감소시킬 새로운 알고리즘을 제안한다. 이 제안된 알고리즘은 줄어든 확장길이 만큼 스칼라 곱의 속도를 향상시킬 수 있게 한다.

본 논문의 구성은 다음과 같다. 2절에서 타원곡선의 기본성질과 프로베니우스 확장을 이용한 스칼라 곱 연산 방법을 간략하게 소개한다. 3절에서는 홀수 표수 확장체위의 타원곡선에서 프로베니우스 자기준동형 확장길이를 줄이는 알고리즘을 제시한다. 그리고 4절에서는 제안된 방법과 기존의 방법들과의 확장길이를 비교하고, 5절에서 결론을 맺는다.

II. 프로베니우스 확장을 이용한 스칼라 곱 연산 방법

본 논문에서는 아래와 같은 형태의 Weierstrass 방정식에 의해 정의된 표수가 홀수인 확장체 위의 타원곡선 $E(F_{q^n})$만 고려한다.

$$E: y^2 = x^3 + ax + b$$

여기서 $p \geq 5$ 소수이고 $q = p^s$, $a, b \in F_q$이다. q-지수승 프로베니우스 자기준동형(Frobenius endomorphism) Φ는 다음과 같이 정의된다.

$$\Phi : E(F_{q^n}) \rightarrow E(F_{q^n}) \ by \ (x, y) \mapsto (x^q, y^q)$$

그러면 Φ는 $E(F_{q^n})$에서 $E(F_{q^n})$로 가는 자기준동형환(endomorphism ring $End(E(F_{q^n}))$의 원소이며 $\Phi^2 - t\Phi + q = 0$을 만족한다. 여기서 t는 Φ의 자취(trace)로 $t = q + 1 - \#E(F_q)$을 만족하고 타원 곡선의 위수는 $|E(F_q)| = q + 1 - t$을 만족한다. 또한 Hasse의 정리[9]에 의해 $|t| \leq 2\sqrt{q}$ 임을 알 수 있다.

암호학적 관점에서 MOV[4]의 공격에 안전하기 위해 non-supersingular 타원곡선만을 고려하자. non-supersingular 타원곡선의 자기준동형 $End(E(F_{q^n}))$은 complex multiplication을 가지므로 $End(E(F_{q^n})) \subset Q(\sqrt{t^2 - 4q})$이다. 또한 $D = t^2 - 4q$라 하면 Hasse의 정리[9]에 의해 $D < 0$ 이며,

$$End(E(F_{q^n})) \cong Z[\theta]$$
$$\theta = \begin{cases} (1+\sqrt{D})/2 & \text{if } t \text{ 홀수} \\ \sqrt{D}/2 & \text{if } t \text{ 짝수} \end{cases}$$

이다.

[보조정리 1]

$\beta \in Z[\Phi]$로 놓자. 그러면 $\beta = \delta\Phi + r$을 만족하는 원소 $\delta \in Z[\Phi]$와 정수 $r \in \{-(q-1)/2+1, \cdots, (q-1)/2\}$가 유일하게 존재한다.

[증명]

$\beta = a + b\Phi \in Z[\Phi]$의 정수 a에 대하여 다음을 만족하는 유일한 정수 k, r이 존재한다.

$$a = kq + r$$

여기서 $r \in \{-(q-1)/2, \cdots, (q-1)/2\}$ 이다. 그리고 $\Phi^2 - t\Phi + q = 0$ 이므로 $a = k(t\Phi - \Phi^2) + r$이다. 그래서 $\beta = a + b\Phi = r + \Phi(kt + b - k\Phi)$로 표현할 수 있다. □

[보조정리 2] ([10] 참조)

$N_{Z[\Phi]/Z}(\beta) \leq \frac{(\sqrt{q}+2)^2}{4}$을 만족하는 $\beta \in Z[\Phi]$에 대하여 $\beta = \sum_{i=0}^{3} r_i \Phi^i$ 로 표현할 수 있다.
여기서 $r_i \in \{-(q+1)/2, \cdots, (q+1)/2\}$ 이다.

[정리 1]

$\beta \in Z[\Phi]$로 놓자. 그러면 β을 다음과 같이 표현할 수 있다.

$$\beta = \sum_{i=0}^{k} r_i \Phi^i$$

여기서 $r_i \in \{-(q+1)/2, \cdots, (q+1)/2\}$이고, $k \leq \lceil \log_q 4N_{Z[\Phi]/Z}(\beta) \rceil + 3$ 이다.

[증명]

보조정리 1에 의해

$$\beta = \beta_0 = \beta_1\Phi + r_0 = (\beta_2\Phi + r_1)\Phi + r_0$$
$$= \sum_{i=0}^{j} r_i\Phi^i + \beta_{j+1}\Phi^{j+1}$$

이다. 여기서 $r_i \in \{-(q-1)/2, \cdots, (q-1)/2\}$ 이다. $\beta_j = \beta_{j+1}\Phi + r_j$ 에 삼각 부등식을 적용하여 다음을 유도할 수 있다.(단 $\| \cdot \| = \sqrt{N_{Z[\Phi]/Z}(\cdot)}$)

$$||\beta_{j+1}|| \le \frac{||\beta_j|| + ||r_j||}{||\Phi||} \le \frac{||\beta_j|| + \frac{q-1}{2}}{\sqrt{q}}$$
$$\le \frac{||\beta_0||}{q^{(j+1)/2}} + \frac{q-1}{2}\sum_{i=1}^{j+1} q^{-i/2}$$
$$\le \frac{||\beta_0||}{q^{(j+1)/2}} + \frac{\sqrt{q}+1}{2}$$

만약 $j \ge \lceil 2\log_q 2||\beta_0|| \rceil - 1$ 이면,

$$\frac{||\beta_0||}{q^{(j+1)/2}} \le \frac{1}{2} \text{이므로 } N_{Z[\Phi]/Z}(\beta_0) \le \frac{(\sqrt{q}+2)^2}{4}$$

이다. 그래서 보조정리 2에 의해 β_{j+1}의 확장길이는 최대 4이다. 따라서

$k \le \lceil \log_q 4N_{Z[\Phi]/Z}(\beta) \rceil + 3$ 가 된다.

[알고리즘 1]

입력 : $\beta = a_1 + a_2\Phi \in Z[\Phi]$, $P \in E(F_{q^n})$.

출력 : $\beta \cdot P \in E(F_{q^n})$

〔전개부분〕

1. $x = a_1$, $y = a_2$, $i = 0$로 놓는다.
2. i에 대하여 $|x| > \frac{q+1}{2}$ 또는 $|y| > \frac{q+1}{2}$ 을 만족하는 동안 반복 시행한다.
   (a) $z \equiv x \pmod q$를 계산한다.
   (b) $r_i = \begin{cases} z & \text{if } x \le \frac{q+1}{2} \\ z-q & \text{otherwise.} \end{cases}$ 로 둔다.
   (c) $h = (r_i - x)/q$, $x = y - th$, $y = h$, 그리 $i = i+1$로 둔다.
3. $r_i = x$, $r_{i+1} = y$.

〔계산부분〕

4. $H = y \cdot \phi(P) + x \cdot P$
5 j를 $i-1$에서 0까지 감소하면서 다음을 실행한다.
   (a) 만약 $r_j \ge 0$이면, $H = \phi(H) + r_j \cdot P$
   (b) 그렇지 않으면, $H = \phi(H) - |r_j| \cdot P$을 계산.
6. H을 출력한다.

〔전개부분〕의 결과 y는 정리 1에서의 r_k이다. 알고리즘 1은 Φ-확장의 계수 r_i을 $\pm(q+1)/2$까지 허용한다는 점에서 Müller[5]가 제시한 알고리즘과 다르다. 이 알고리즘은 Müller의 알고리즘보다 확장길이를 약간 더 줄일 수 있다. 물론 Φ-확장이 유일하게 표현되지는 않지만, 타원곡선 점의 스칼라 연산에서는 같은 결과를 산출하므로 문제가 되지 않는다. 단계 3에서 $\beta = \sum_{i=0}^{k} r_i \Phi^i$ 인 r_i들을 얻는다.

여기서 $k \le \lceil \log_q 4N_{Z[\Phi]/Z}(\beta) \rceil + 3$이다.

그래서 ,

$$\beta \cdot P = \sum_{i=0}^{k} r_i \Phi^i(P)$$
$$= \Phi(\cdots\Phi(r_k\Phi(P) + r_{k-1}P) + \cdots + r_1P) + r_0P$$

로 계산된다.

타원곡선의 스칼라 곱 mP을 계산하기 위해 직접 알고리즘 1을 사용한다면 m의 Φ-확장길이는 $k+1 \le \lceil \log_q Am^2 \rceil + 4$ 이다. mP 계산 속도는 확장길이에 밀접한 관계가 있다. 즉, Φ의 확장 길이가 줄어들면 연산 속도는 빨라진다. 따라서 다음절에서 프로베니우스 확장길이를 줄이는 방법에 대해 살펴본다

Ⅲ. 프로베니우스 자기준동형 확장길이를 줄이는 방법

암호학적 응용에 있어 타원곡선 $E(F_{q^n})$의 위수는 큰 소인수 p을 가져야 한다. 타원곡선의 위수 $\#E(F_{q^n}) = hp$라 하자 여기서 cofactor h는 작은 정수 값을 갖는다. 그리고 Hasse와 Weil 정리에 의해 다음을 만족한다.

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

정리 1에 의해 프로베니우스 확장방법을 사용하여 스칼라 곱을 사용할 때에는 확장길이가 짧을수록 속도가 빨라진다. 다음과 같이 Smart[10]에 의해 확장길이를 줄이는 방법이 제안되었다.

<확장길이를 줄이는 방법>

〔Step 1〕 m을 $\Phi^n - 1$로 나눈 나머지 ρ계산.

〔Step 2〕 $mP = \rho P = \sum_{i=0}^{k} r_i \Phi^i(P)$ 을 계산

임의의 점 $Q \in E(F_{q^n})$에 대해 $\Phi^n(Q) = Q$을 만족하므로 $(\Phi^n - 1)Q = O$이다. 타원곡선의 스칼라 곱은 큰 정수 $m \approx p \approx q^n$에 대하여 mP 계산하는 것이다. 여기서 m의 노름은 $N_{Z[\Phi]/Z}(m) = m^2 \approx q^{2n}$이다. 그러나 나머지 ρ의 노름은 m의 노름의 절반정도인 q^{n+1}이다.[10] 〔Step1〕에 의해 $m = \delta(\Phi^n - 1) + \rho$인 $\delta, \rho \in Z[\Phi]$가 존재한다. 결과적으로 $mQ = \delta(\Phi^n - 1)Q$

$+\rho Q=\rho Q$가 된다. 여기서 ρ을 사용하면 m을 바로 사용한 것 보다 확장길이를 반으로 줄일 수 있다.

본 절에서는 Smart의 방법을 개선하여 프로베니우스 확장 길이 $k+1$을 줄이는 방법에 대하여 설명한다. 타원곡선 암호시스템에서는 전체 군 $E(F_{q^n})$보다 큰 소수 p을 위수로 갖는 점 P로 생성되는 순환 부분군 $\langle P\rangle$을 사용한다. 따라서 전체 군이 아닌 순환 부분 $\langle P\rangle$에서 스칼라 곱의 연산을 고려할 것이다. 그래서 Φ^n-1대신에 $\langle P\rangle$에서 같은 역할을 하는 $\alpha=a+b\Phi\in Z(\Phi)$을 적용하여 확장길이를 최소화 한다.[6] 다음은 이런 α을 찾는 방법을 설명한다.

Cofactor $h<p$는 작은 정수 값이므로 $p\mid \#E(F_{q^n})$ $p^2\nmid \#E(F_{q^n})$이다. 그래서 타원곡선 군 $E(F_{q^n})$에서 위수가 p인 부분군은 오직 $\langle P\rangle$뿐이다. 또 Φ는 준동형 함수이므로 $p\Phi(P)=\Phi(pP)=\Phi(O)=O$을 만족하여 $\Phi(P)$의 위수도 p가 된다. 위수가 p인 원소는 반듯이 $\langle P\rangle$의 원소이어야 한다. 즉, $\Phi(P)=\mu P$ 을 만족하는 μ가 존재한다. 그리고 $\Phi^2-t\Phi+q=0$이므로 $\mu^2-t\mu+q\equiv 0 \bmod p$을 만족한다. 한편 Cornnacchia 알고리즘[1]을 사용하여 $N_{Z[\Phi]/Z}(\alpha)=m_s p$을 만족하는 $\alpha=a+b\Phi\in Z[\Phi]$을 찾을 수 있다. 또한 $\Phi(P)=\mu P$ $\overline{\Phi}(P)=(t-\Phi)(P)=(t-\mu)P$ 이므로 $(a+b\mu)(a+bt-b\mu)\equiv 0 \bmod p$을 유도할 수 있다. 따라서 $a+b\mu\equiv 0$ 또는 $a+bt-b\mu\equiv 0 \bmod p$ 이므로 $(a+b\Phi)P=O$ 또는 $(a+b\overline{\Phi})P=O$를 만족한다. 즉,

$$N_{Z[\Phi]/Z}(\alpha)=m_s p,\ \alpha P=O \qquad (1)$$

을 만족하는 $\alpha=a+b\Phi\in Z(\Phi)$을 쉽게 찾을 수 있다. 다음 정리로 m_s의 존재성과 m_s가 작은 정수 값임을 알 수 있다.

[보조정리 3] ([11] 참조)

$K=Q(\sqrt{D})$을 허수 이차 체라 하자. K의 영이 아닌 이데알(ideal) I에 대하여 이데알 J와 K의 정수환 O_K의 어떤 원소 α가 존재하여 다음을 만족한다:

$$N_{K/Q}(J)\le \frac{2}{\pi}\sqrt{|D|},\ I\cdot J=(\alpha)$$

[정리 2]

$p\mid \#E(F_{q^n})$ $p^2\nmid \#E(F_{q^n})$이라고 하자.

만약 $D=t^2-4q$가 제곱인수를 갖지 않으면 어떤 양의 정수 $m_s<1.28\sqrt{q}$에 대하여 $N_{Z[\Phi]/K}(a+b\Phi)=m_s p$을 만족하는 $\alpha\in Z[\Phi]$가 존재한다.

[증명]

$\#E(F_{q^n})=hp$, $K=Q(\sqrt{D})$라 하자.

$N_{K/Q}(\Phi^n-1)=\#E(F_{q^n})=hp$ 이므로 p는 K/Q에서 쪼개(split)진다. I을 $N_{K/Q}(I)=p$인 K의 소수 이데알이라 하자. 보조정리 3에 의해, 어떤 $\alpha\in O_K$에 대해 $I\cdot J=(\alpha)$을 만족하는 $N_{K/Q}(J)\le \frac{2}{\pi}\sqrt{|D|}$인 이데알 J가 존재한다. 즉,

$$\begin{aligned} m_s &= N_{K/Q}(J)\le \frac{2}{\pi}\sqrt{|t^2-4q|}\\ &\le \frac{2}{\pi}\sqrt{4q}<1.28\sqrt{q}. \end{aligned}$$

가 된다. 이제 $\alpha\in Z[\Phi]$임을 보이자. Φ의 자취(trace)에 따라

$$\theta=\begin{cases}(1+\sqrt{D})/2 & \text{if } t \text{ 홀수}\\ \sqrt{D}/2 & \text{if } t \text{ 짝수}\end{cases}\ \text{이고,}$$

$O_K=Z[\theta]$이다. 따라서 $\Phi=(t\pm\sqrt{D})/2$이므로,

$$\theta=\begin{cases}\Phi-(t-1)/2 & \text{if } t \text{ 홀수}, \Phi=(t+\sqrt{D})/2\\ -\Phi+(t+1)/2 & \text{if } t \text{ 홀수}, \Phi=(t-\sqrt{D})/2\\ \Phi-t/2 & \text{if } t \text{ 짝수}, \Phi=(t+\sqrt{D})/2\\ -\Phi+t/2 & \text{if } t \text{ 짝수}, \Phi=(t-\sqrt{D})/2\end{cases}\ \text{이고,}$$

$O_K=Z[\Phi]=Z[\theta]$이다. □

[Remark]

정리 2에서 $D=t^2-4q$가 $s^2\mid D$이고, $D'=D/s^2$가 제곱인수를 갖지 않으면 $m_s<s^2 1.28\sqrt{q}$ 조건을 만족하는 m_s을 찾을 수 있다. 그래서 일반적인 $D=t^2-4q$에 대하여도 m_s는 작은 정수 값으로 잡을 수 있다.

작은 정수 m_s에 대하여 (1)를 만족하는 $\alpha\in Z[\Phi]$을 찾는 과정은 사전계산으로 이루어지므로 스칼라 곱의 연산 속도에 영향을 주지 않는다.

정리 2 에서 사용된 θ을 세분화하여 다시 정의하자.

$$\theta=\begin{cases}(1+\sqrt{D})/2 & \text{if } t \text{ 홀수}, \Phi=(t+\sqrt{D})/2\\ (1-\sqrt{D})/2 & \text{if } t \text{ 홀수}, \Phi=(t-\sqrt{D})/2\\ \sqrt{D}/2 & \text{if } t \text{ 짝수}, \Phi=(t+\sqrt{D})/2\\ -\sqrt{D}/2 & \text{if } t \text{ 짝수}, \Phi=(t-\sqrt{D})/2\end{cases}$$

새롭게 정의한 θ 값은 $\Phi=(t-\sqrt{D})/2$인 경우에

기존의 θ값과 다른 값을 갖는다. 그러나 정리 2에 $O_K = Z[\Phi] = Z[\theta]$을 만족하는 역할에는 큰 변화가 없다. 그래서 새롭게 정의한 θ는 Φ값에 상관없이 $\theta = \Phi - \lfloor t/2 \rfloor$ 을 만족한다.

[정의 1]

실수 λ에 대하여, 다음과 같은 가환환 A와 승법적 함수 Ψ를 가정하자.

$$\Psi : A\backslash\{0\} \rightarrow N$$

만약 모든 $a, b \in A, b \neq 0$에 대해, $a = bq + r$ 같은 $q, r \in A$를 발견할 수 있다면, 가환환 A를 λ-유클리드 (λ-Euclidean) 환이라 한다. 여기서 r은 $r \neq 0$ 또는 $\Psi(r) < \lambda\ \Psi(b)$ 이다.

정리 3에서 $Z[\Phi]$가 양의 실수 λ에 대해 λ-유클리드 λ-Euclidean) 환이 됨을 보이고, 특히 θ을 사용하여 Smart[10]가 제시한 λ의 최대값을 1/4 줄일 수 있음을 보인다.

[정리 3]

$\alpha = a + b\Phi \neq 0 \in Z[\Phi]$로 놓자.

만약 $\beta \in Z[\Phi]$ 일 때, 다음을 만족하는 $\delta, \rho \in Z[\Phi]$가 존재한다.

$$\beta = \delta\alpha + \rho,\ N_{Z[\Phi]/Z}(\rho) \leq \lambda N_{Z[\Phi]/Z}(\alpha),$$

$$0 < \lambda \leq \begin{cases} (9+4q)/16 & \text{if } t \text{ 홀수} \\ (1+q)/4 & \text{if } t \text{ 짝수} \end{cases}$$

[증명]

$Z[\Phi] = Z[\theta]$이므로 Z-기저 $\{1, \Phi\}$을 $\{1, \theta\}$로 바꾼다. 그리고 Z-기저 $\{1, \theta\}$에 대하여 $\rho \in Z[\Phi]$을 계산하는 과정을 살펴본다.

$$\gamma = \beta/\alpha = \beta\overline{\alpha}/\alpha\overline{\alpha} = (x_1 + x_2\theta)/N_{Z[\Phi]/Z}(\alpha)$$

$$\delta = \left\lfloor \frac{x_1}{N_{Z[\Phi]/Z}(\alpha)} \right\rceil + \left\lfloor \frac{x_2}{N_{Z[\Phi]/Z}(\alpha)} \right\rceil \theta$$

여기서 $\overline{\alpha}$는 α의 켤레 복소수(complex conjugate)이고, $\lfloor x \rceil$ 는 x에 가장 가까운 정수이다.

그래서 $\rho = \beta - \delta\alpha = \alpha(\gamma - \delta)$가 되고,

$$N_{Z[\Phi]/Z}(\rho)/N_{Z[\Phi]/Z}(\alpha) = N_{Z[\Phi]/Z}(\gamma - \delta)$$

$$\leq N_{Z[\Phi]/Z}\left(\frac{1}{2} + \frac{1}{2}\theta\right) = \frac{1}{4} N_{Z[\Phi]/Z}(1+\theta)$$

$$= \begin{cases} \frac{1}{4} N_{Z[\Phi]/Z}\frac{(3 \pm \sqrt{D})}{2} & \text{if } t \text{ 홀수} \\ \frac{1}{4} N_{Z[\Phi]/Z}\frac{(2 \pm \sqrt{D})}{2} & \text{if } t \text{ 짝수} \end{cases}$$

$$= \begin{cases} \frac{(9-D)}{16} \leq \frac{(9+4q)}{16} & \text{if } t \text{ 홀수} \\ \frac{(4-D)}{4} \leq \frac{(1+q)}{4} & \text{if } t \text{ 짝수} \end{cases}$$

가 된다. □

다음 알고리즘 2는 m을 α로 나눈 나머지 $\rho = r_1 + r_2\Phi$ 구하는 방법이다. 알고리즘 2를 수행하기 전에 (1)를 만족하는 $\alpha = a + b\Phi \in Z[\Phi]$을 찾아둔다. 그래서 임의의 $m \in N$과 함께 $m_s, p, a_1, b_1, c, T, N$ 들도 [본 계산]의 입력 값으로 사용된다.

[알고리즘 2] m을 α로 나눈 나머지 구하기)

입력 : $m \in N$

출력 : $\rho = r_1 + r_2\Phi$ $N_{K/Q}(\rho) \leq \lambda N_{K/Q}(\alpha)$

[사전 계산]

1. $c = -\lfloor t/2 \rfloor = \begin{cases} -(t-1)/2 & \text{if } t \text{ 홀수}, \\ -t/2 & \text{if } t \text{ 짝수}, \end{cases}$
2. $T = \begin{cases} 1 & \text{if } t \text{ 홀수}, \\ 0 & \text{if } t \text{ 짝수}, \end{cases}$
3. $N = \begin{cases} q + c(c-1) & \text{if } t \text{ 홀수}, \\ q + c^2 & \text{if } t \text{ 짝수}, \end{cases}$
4. $a_1 = a - bc,\ b_1 = b,\quad \alpha = a_1 + b_1\theta$

[본 계산]

1. $x_1 = m(a_1 + b_1 T)$, 그리고 $x_2 = -mb_1$,
2. $y_i = \left\lfloor \frac{x_i}{m_s p} \right\rceil$, $(i = 1, 2)$.
3. $r'_1 = m - (a_1 y_1 - Nb_1 y_2)$,
   $r'_2 = -(a_1 y_2 + b_1 y_1 + Tb_1 y_2)$,
4. $r_1 = (r'_1 + r'_2 c)$, $r_2 = r'_2$,
5. r_1, r_2 결과를 보낸다.

[증명]

[사전 계산]의 단계 2, 3의 T, N은 θ의 자취(Trace)와 노름 (Norm) 이다. 그래서,

$T = Tr(\theta) = \theta + \overline{\theta} = \Phi + \overline{\Phi} + 2c = t - 2\lfloor t/2 \rfloor$

$N = \theta\,\overline{\theta} = (\Phi + c)(\overline{\Phi} + c) = q - tc - c^2$ 이다.

t가 홀수일 때 $t = 1 - 2c$ 이므로 $T = 1$, $N = q -$

$tc-c^2=q-(1-2c)c-c^2=q+c(c-1)$ 이고, t가 짝수일 때 $t=-2c$ 이므로 $T=0$, $N=q-tc-c^2$ $=q-(-2c)c-c^2=q+c^2$이다.

[본 계산]의 단계 1은 $m\,\overline{a}=x_1+x_2\theta$에 대한 결과값이다. $a=a_1+b_1\theta$ 이므로

$$\overline{a}=a_1+b_1\overline{\theta}=a_1+b_1(T-\theta)$$
$$=(a_1+b_1T)+(-b_1)\theta \text{ 이다.}$$

[본 계산]의 단계 3은 다음으로 증명된다.
$\theta^2=T\theta-N$ 이므로,
$\rho=m-(y_1+y_2\theta)(a_1+a_2\theta)$
$=m-(a_1y_1-Nb_1y_2)-(a_1y_2+b_1y_1+Tb_1y_2)\theta$
가 된다. □

알고리즘 2의 [사전계산]은 타원곡선의 구성단계에서 한번만 계산하면 되므로 ρ를 구하는 계산량에 포함되지 않는다. 그리고 T, N, c들은 q 보다 작은 수이므로 이들의 계산량은 무시한다. 따라서 이 알고리즘은 2번의 라운드 계산과 6번의 큰 정수 곱셈이 필요하다. 최종적으로, 스칼라 곱의 고속연산을 위한 새로운 방법을 요약하면 다음과 같다.

<스칼라 곱을 위한 새로운 방법>

[Step 1] $N_{Z[\Phi]/Z}(a)=m_sp$, $aP=O$ 조건을 만족하는 $\alpha=a+b\Phi\in Z[\Phi]$을 결정.

[Step 2] 알고리즘 2를 사용하 m을 α로 나눈 나머지 $\rho\in Z[\Phi]$구하기 .

[Step 4] 알고리즘 1을 사용하여 $\rho=\sum_{i=0}^{k} r_i\Phi^i$로 표현.

[Step 3] $mP=\rho(P)=\sum_{i=0}^{k} r_i\Phi^i(P)$을 계산

Ⅳ. 구현한 결과값 비교

암호학적으로 적절한 타원곡선은 큰 순환부분 군을 갖는 non-supersingular 곡선이다. 특히 이 순환부분 군의 위수는 2^{160}이상이어야 한다. 만약 2^{160}정도 위수의 유한체 위에서 정의된 타원곡선을 사용하면 다음과 같은 두 가지 문제점이 있다. 하나는 스칼라 곱 연산에서 프로베니우스 확장방법을 사용할 수 없다. 다른 문제는 타원곡선이 큰 순환부분 군을 갖는지 확인하기 위해 타원곡선의 위수를 얻어야 한다. 일반적으로 타원곡선의 위수를 구하는 알고리즘들인 Schoof[8], Satoh[7], Schoof-Elkies-Atkin[2]은 큰 유한체 위에서 정의된 큰 차수 다항식의 근을 얻어야 하므로 어려운 많은 시간과 작업을 요구한다. 하지만 본 논문에서 제안된 타원곡선의 경우와 같이 작은 유한체 위에서 정의된 타원곡선의 경우 위수를 쉽게 구할 수 있다. 작은 유한체 위의 $E(F_q)=q+1-t$ 의 위수는 쉽게 계산이 가능하고 $\#E(F_{q^n})=q^n+1-t_n$ 는 다음의 방법으로 쉽게 구할 수 있다.

$$t_0=2,\ t_1=t=q+1-\#E(F_q),$$
$$t_n=t_1t_{n-1}-qt_{n-2}.$$

그러나 표수가 2인 경우만 고려한다면 암호학적으로 적절한 타원곡선을 많이 얻을 수 없다. 그래서 작은 홀수 표수를 갖는 유한체에서 정의된 타원곡선도 유용하게 사용될 수 있다.

Smart는 프로베니우스의 확장길이를 줄이기 위해 Φ^n-1을 사용하여 m을 나누었다. 하지만 본 논문에서 제안한 방법은 Φ^n-1 대신 (1)을 만족하는 α을 사용하여 Φ의 확장길이를 더욱 줄였다. 그러나 Smart 방법은 타원곡선의 모든 점에 적용 가능하지만 본 논문에서 제안된 새로운 방법은 타원곡선 군 전체가 아니라 큰 소수 p을 위수로 갖는 순환 부분 군 $\langle P\rangle$에만 적용됨에 주의하자. 다음은 각각의 방법에 따른 확장길이를 비교하였다. Reduction을 사용하지 않은 [방법 1], Φ^n-1을 사용한 [방법 2]와 $\alpha=a+b\Phi\in Z[\Phi]$을 사용한 [방법 3]의 확장길이는 다음과 같다.

[표 1] 확장길이 비교

[방법 1]	$\lceil \log_q m^2 \rceil+4$
[방법 2]	$\lceil \log_q(\lambda\#E(F_{q^n})) \rceil+4$
[방법 3]	$\lceil \log_q(\lambda m_sp) \rceil+4$

이와 같이 새로운 방법은 기존의 방법보다 확장 길이를 약 $\lceil \log_q(h/m_s) \rceil$ 정도 줄일 수 있다. 다음 [표 2]는 각 방법의 결과를 비교한 실험 값이다. 특히 λ와 프로베니우스 확장길이는 10^5개의 난수 $m \le p$에 대한 평균값이다. 여기서 다루는 F_q의 q는 $5\le q\le 23$인 소수이다. 그리고 확장 체 q^n의 크기는 130비트에서 220비트로 제한한다. 또한 타원곡선의 위수는 155비트 이상의 큰 소인수를 가지고, cofactor h는 $\#E(F_q)<h<10^5$을 만족하는 타원곡선만 고려하였다.

〔표 2〕 확장길이 실험값 비교

q	n	t	$\lceil \log_2 p \rceil$	h
m_s	λ	확장길이 no reduction	확장길이 by Φ^n-1	확장길이 by α
5	79	-1	168	63049
1	0.497	142	78	71
7	61	-2	156	46370
2	0.590	109	60	55
7	67	3	174	29485
1	0.494	122	66	61
11	53	4	167	91592
1	0.673	95	52	48
11	53	6	168	68694
1	0.252	97	53	48
11	59	-1	189	36829
1	1.02	108	58	54
11	59	1	191	9097
1	1.000	109	58	54
11	61	4	199	7816
1	0.664	113	60	57
13	47	-3	162	4811
1	0.992	86	46	43
13	59	-3	205	12053
1	1.004	110	58	55
13	59	7	204	22309
1	0.167	110	59	55
17	47	-6	180	6792
3	0.746	87	47	44
17	47	-1	177	39311
1	1.486	85	46	43
19	41	2	164	1494
2	1.570	76	40	38
23	37	5	156	2831
1	1.509	68	37	34
23	41	-7	171	25451
1	1.001	75	41	37
23	41	-4	170	71204
4	1.663	74	40	37

V. 결 론

본 논문은 작은 홀수 표수를 갖는 유한체 위에서 정의된 non-supersingular 타원곡선에서 프로베니우스 자기준동형(Frobenius endomorphism)을 이용한 기존의 스칼라 고속연산을 개선하는 방법을 제안하였다. 이 방법은 Smart의 방법 보다 프로베니우스 자기준동형 (Frobenius endomorphism) 확장길이를 약 $\lfloor \log_q(h/m_s) \rfloor$ 정도 줄일 수 있다. 그리고 4절에서 새로운 방법과 기존의 방법들을 통한 구체적인 실험결과의 비교로 확장길이가 감소함을 보였다.

참 고 문 헌

〔1〕 G. Cornacchia, "Su di un metodo per la risoluzione in numeri interi dell' equazion $\sum_{h=0}^{n} C_h x^{n-h} y^h = P$", Giornale di Matematiche di Battaglini, 46, pp. 33~90, 1908.

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

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

〔4〕 A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to a finite field", IEEE Trans. Info. Theory, 39, pp. 1639~1646, 1993.

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

〔6〕 Y.-H. Park, S. Oh, S. Lee, J. Lim, M. Sung, "An improved method of multiplication on certain elliptic curves", To appear in PKC 2002.

〔7〕 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.

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

〔9〕 J. H. Silverman, Advanced Topics in the

Arithmetic of Elliptic Curves. Springer-Verlag, New York, 1994.

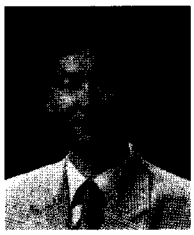
[10] N. Smart, "Elliptic curve cryptosystems over small fields of odd characteristic", Journal of Cryptology, 1999, pp 141~145.

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

〈著者紹介〉

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

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



이 상 진 (Samg-jin Lee) 정회원
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 2월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월 : 한국전자통신연구원 선임 연구원,
1999년 2월~현재 : 고려대학교 자연과학대학 부교수, 고려대학교 정보보호대학원 겸임교수, 고려대학교 정보보호기술연구센터 연구실장
〈관심분야〉 블록 암호 및 스트림 암호의 분석 및 설계, 암호 프로토콜, 공개키 암호 알고리즘의 분석

황 정 연 (Jung Yeon Hwang)
1999년 2월 : 고려대학교 수학과 학사
2001년 3월~현재 : 고려대학교 정보보호대학원 석사 과정
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜

김 창 한 (Chang-Han Kim) 정회원
1985년 2월 : 고려대학교 수학과 학사
1987년 2월 : 고려대학교 수학과 석사
1992년 2월 : 고려대학교 수학과 박사
2000년 2월~현재 : 세명대학교 컴퓨터수리정보학과 부교수
〈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 분석