情報保護學會論文誌
第12券 第5號, 2002. 10

Early-abort 전략을 이용한 타원곡선 생성 알고리즘에 사용되는 SEA 알고리즘 연구

정 배 은*, 류 희 수*

On the SEA algorithm used in finding secure elliptic curves with an early-abort strategy

Bae Eun Jung*, Heuisu Ryu*

요 약

타원곡선 암호 사용에 있어 암호학적으로 안전한 타원곡선의 선택이 안전한 암호 스킴 구성에 있어서 대단히 중요하다. 현재까지 알려진 공격에 대하여 타원곡선의 안전성 결정 요소 가운데 하나가 타원곡선 그룹의 위수이다. 따라서, 타원곡선의 랜덤 커브 생성에 있어 위수 계산은 필수적이다. characteristic이 2인 경우, 효율적인 랜덤 커브 생성 알고리즘은 early-abort 전략을 사용한 알고리즘으로 SEA 알고리즘을 abort 단계에 사용하고, 위수 계산에 Satoh 알고리즘을 사용한 방법이다[1]. [1]에서 abort 단계에서 사용하는 SEA 알고리즘을 변형하여 사용하였다고 기술되어있는데, 구체적인 방법이 제시되지는 않았다. 우리는 이 논문에서, abort 하게 되는 경우에 대하여 관련된 파라미터들을 살펴봄으로써, abort 단계에 소요되는 시간을 효율적으로 줄이는 SEA 알고리즘 변형 방법을 제안하고, 이의 근거로 실험한 결과를 제시하고자 한다.

ABSTRACT

In using elliptic curves in cryptography, it is important to find a secure elliptic curve. The security of elliptic curve cryptosystem is dependent on the cardinality of the given curve. So, it is necessary to count the number of points of a given elliptic curve to obtain secure curve. It is known that when the charateristic is two, the most efficient algorithm finding secure curves is combining the Satoh-FGH algorithm with early-abort strategy[11]. In [1], the authors wrote that they modified SEA algorithm used in early-abort strategy, but they didn't describe the varaint of SEA algorithm. In this paper, we present some modifications of SEA algorithm and show the result of our implementation.

Keyword : *타원곡선 암호, Elkies소수, SEA 알고리즘, early-abort 전략*

Ⅰ. 서 론

이산대수문제를 사용하는 공개키 암호에서는 기반 그룹의 전체 위수의 결정이 중요하다. 현재까지 알려진 공격에서 안전하기 위한 필요조건은 아주 큰 소수(적어도 160비트 이상)가 위수의 인수가 되어야 한다는 것이다. 타원곡선 암호 역시 이러한 조건을 만족하는 암호학적으로 안전한 타원곡선을 선택하는 것이 대단히 중요하다. 그러나, 타원곡선은 타원곡선의 위수를 계산하는 것이 간단하지 않다. 따라서, 현재 대부분은 일부 표준에서 각 비트 당 몇 개의 타원곡선만 제시하고 있다. 다양한 타원곡선을

* 한국전자통신연구원 정보보호연구본부({bejung, hsryu}@etri.re.kr)

사용하기 위해서는 타원곡선의 위수를 계산할 수 있어야 하며, 이러한 관점에서, 타원곡선의 위수 계산 알고리즘이 암호학에서 주목을 받아왔다. 크기가 q ($q=p^n$, p 는 소수)일 때 naive한 알고리즘에 요구되는 계산량은 $O(\log^8 q)$ 비트 연산으로 알려져 있다. 이를 개선한 첫 알고리즘이 Schoof 알고리즘으로, 작은 소수 l에 대하여 l-torsion 점들을 이용하여 위수의 modular 값을 구한 후, chinese remainder 정리를 이용하여 위수를 구하는 알고리즘이다[2]. Schoof 알고리즘에서는 차수가 $O(l^2)$ 인 division 다항식을 이용한다. Schoof 알고리즘의 계산량은 $O(\log^{5+\epsilon} q)$ 으로 알려져 있다. 한편, 차수가 $O(l^2)$ 인 division 다항식을 이용하는 대신 이 다항식의 인수 가운데, 차수 $O(l)$인 인수를 이용하여 계산량을 줄인 SEA알고리즘이 발표되었다[3]. 그러나, SEA 알고리즘은 characteristic 이 큰 소수인 경우에 효율적이며, 이의 계산량은 $O(\log^{4+\epsilon} q)$ 비트 연산이다. 이 후, Couveignes는 characteristic이 작은 경우로 SEA알고리즘을 확장하였고[4], Lercier는 characteristic이 2인 경우에 SEA알고리즘을 제안하였다[5]. 최근, characteristic 이 작은 소수인 경우에, 타원곡선의 위수 계산에 가장 효율적인 알고리즘으로 알려진 것은 Satoh 알고리즘으로, 계산량은 $p\geq 5$ 인 작은 소수인 경우, $O(\log^{3+\epsilon} q)$으로 알려져있다[6]. 이 알고리즘은 곧 Fouquet, Gaudry, Harley와 Skjernaa에 의해 2, 3인 경우로 확장되었다[7,8]. Characteristic이 2인 경우, [7]에 알려진 알고리즘은 $O(n^3 \log p)$정도의 메모리를 필요로 한다. 한편, Eurocrypt'01에서 Vercauteren, Prenel, Vandewalle는 이 알고리즘을 개선하여 $O(n^2)$의 메모리를 사용하며 계산 속도는 1.5배 정도 빠른 알고리즘을 발표한 바 있다[9]. 따라서, characteristic 2인 경우, 현재 알려진 가장 효율적인 알고리즘은 [9]에 발표된 Satoh 알고리즘이라 할 수 있다.

암호학적으로 안전한 타원곡선을 생성하는 방법은 랜덤한 곡선을 생성한 후, 알려진 공격에 대하여 안전한지 조사하여 선택하는 방법으로, 위수가 큰 소수(160비트 이상)를 약수로 갖는 꼴인지 검증해야 한다. 따라서, 랜덤한 타원곡선을 생성하기 위하여, 반드시 위수 계산 알고리즘을 필요로 한다. 현재 알려진 효율적인 방법은 Satoh 알고리즘과 ealry-abort 전략에 SEA 알고리즘을 사용한 방법이다[1]. Early-abort 전략은 이미 [10]에서 사용된 방법이기도 한다. 즉, 타원곡선의 위수를 계산하기 전에, 적당히 작은 소수들에 대하여 l-torsion 점들이 있는지 조사하여 abort 단계를 통과한 곡선들에 대하여만 위수 계산을 하는 방법이다.

우리는 이 논문에서 characteristic 2인 경우에 사용된 early-abort 전략 및 이에 사용되는 SEA 알고리즘의 개요 등을 살펴보고, abort 하는 시간을 줄이기 위한 SEA 알고리즘 변형 방법을 제시한다. Abort 단계에 사용되는 알고리즘은 SEA 알고리즘에서 Elkies 소수인 경우에 프로비니우스 함수의 trace의 modular 값을 계산하게 되는데, 우리는 abort 하기 위한 조건에 필요한 경우만 조사하게 됨으로써 불필요한 시간을 줄인다. 먼저 Ⅱ장에서는 SEA 알고리즘과 관련된 위수 계산 알고리즘의 개략적인 내용을 기술한다. Ⅲ장에서는 [1]에 기술된 early-abort 전략을 구체적으로 기술한다. Ⅳ장에서는 이를 개선한 방법에 대하여 알고리즘의 개요를 기술하고, 몇가지 사실들을 관찰한다. 또한, 실제 알고리즘을 구현하여, 작은 소수에 대하여 어느 정도 효율성을 얻었는지에 대한 자료를 제시하고 이에 근거한 가장 효율적인 방법을 제안하고, 논문을 정리하고자 한다.

Ⅱ. 위수 계산 알고리즘

2.1 Backgrounds

이 절에서는 논문 전개를 위한 notation 및 타원곡선의 위수와 관련된 기본 정리들을 기술한다.

Notation

- $q=2^m$
- $GF(q)$: 위수가 q인 유한체
- $\overline{GF(q)}$: $GF(q)$을 포함하는 대수적 닫힌체
- $E(GF(q))$: $GF(q)$에서 정의된 타원곡선의 집합
- $\overline{E}=E(\overline{GF(q)})$: $\overline{GF(q)}$에서 정의된 타원곡선의 집합
- O : 타원곡선의 무한점(항등원)
- 양의 정수 λ에 대하여 $[\lambda]P$: $P+\cdots+P$: λ번 의 덧셈
- $[-\lambda]P=-[\lambda]P$
- $(P)_X$: 점 P의 x 좌표
- $(P)_Y$: 점 P의 y 좌표
- $E[l]=\{P\in E(\overline{GF(q)}) \mid [l]P=O\}$
- l : 소수
- φ : $E(\overline{GF(q)})$에서 정의된 Frobenus 함수, 즉,

$\varphi(x, y) = (x^q, y^q)$
· q_l : $q \pmod l$
· t : 주어진 타원곡선에서의 φ의 trace
· t_l : $t \pmod l$
· $\#(E)$: 그룹 E 위수
· f_λ : 타원곡선의 λ번째 division 다항식
· $\Phi_l(x, y)$: 타원곡선의 l번째 modular 다항식
· CRT : Chinese Remainder Theorem
· BSGS : Baby Step Giant Step

아래 정리들은 타원곡선 이론에서 아주 잘 알려진 정리들이다.

[Hasse's 정리]
$\#(E(GF(q)) = q+1-t$, ($|t| \le 2\sqrt{q}$)이 성립한다.

[정리1][11]
(1) $P = (x, y) \in E(GF(q))$에 대하여
$\varphi^2(P) - [t]\varphi(P) + [q]P = O$ 이 성립한다.
(2) characteristic 2라고 가정하자. $\lambda \ge 2$ 이면서, $P \in E \backslash E[\lambda]$인 경우, 다음이 성립한다.

$$([\lambda]P)_X = x + \frac{f_{\lambda-1} f_{\lambda+1}}{f_\lambda^2},$$

$$([\lambda]P)_Y = x + y + \frac{(x^2+x+y) f_{\lambda-1} f_\lambda f_{\lambda+1} + f_{\lambda-2} f_{\lambda+1}^2}{x f_\lambda^3}$$

(Note1) : division 다항식들은 귀납적으로 정의된다. ([11] 참고)

2.2 Schoof 알고리즘

Hasse's 정리에서 $|t| \le 2\sqrt{q}$ 이므로, t의 modular $4\sqrt{q}$ 에서의 값을 구한 후, Hasse's 정리에 대입하면 위수를 구할 수 있다.

$L = \prod_{2 \le l \le l_{max},\ l:\text{소수}} l$ 이라 하자. 여기에서 l_{max}는 $L \ge 4\sqrt{q}$을 만족하게 하는 최소의 소수를 가리킨다. 이제 modular L에 대한 t의 값을 계산하는 방법은 다음과 같다.

Step 1 $2 \le l \le l_{max}$인 모든 l에서 $t \pmod l$을 구한다.
Step 2 CRT를 이용하여 각 l에서 $t \equiv t_l \pmod l$을 만족하는 t를 구한다.

위 단계에서 계산량이 많이 요구되는 단계는 (step 1)이다. Schoof 알고리즘의 기본 개요는 (step 1)을 계산하는 데에 있다. 이 절에서는 먼저 Schoof 알고리즘에 대하여 살펴본다. (정리1)에 의하여 $\varphi^2(P) - [t]\varphi(P) + [q]P = O$이 만족한다. 한편, $P \in E[l]$에 대하여, $[q]P = [q_l]P$이며, $[t]\varphi(P) = [t_l]\varphi(P)$이 성립한다. 따라서, $P \in E[l]$인 경우, $\varphi^2(P) - [t_l]\varphi(P) + [q_l]P = O$이 성립한다. 따라서, $\varphi^2(P) + [q_l]P = [\lambda]\varphi(P)$을 만족하는 O이 아닌 점 $P \in E[l]$이 존재하는 λ가 t_l의 후보가 될 수 있다. 한편, l이 소수이므로, 그러한 λ는 유일하게 존재한다. 따라서, 다음의 정리가 성립함을 알 수 있다.

[정리 2]
타원곡선 그룹 E에서 $\varphi^2(P) + [q_l]P = [\lambda]\varphi(P)$을 만족하는 $P \in E[l]$이 존재하는 λ가 바로 t_l이 된다.

따라서, t_l를 찾기 위하여 $0 \le \lambda \le l-1$인 λ에 대해 $\varphi^2(P) + [q_l]P = [\lambda]\varphi(P)$을 만족하는 $O \ne P \in E[l]$이 있는지 조사하면 되므로, 다음의 알고리즘을 얻는다.

Basic Schoof 알고리즘[2,11]
1. $M \leftarrow 2$, $l \leftarrow 3$, and $S \leftarrow \{t \pmod 2, 2)\}$
2. **While** $M < 4\sqrt{q}$, *do* :
3. **For** $\tau = 0, \ldots, (l-1)/2$, *do* :
4. **using the formulae (정리1) check whether, for** $P \in E[l] \backslash O$, $\varphi^2(P) + [q_l]P = \pm[\tau]\varphi(P)$
Exactly one such τ will pass this test.
5. $S \leftarrow S \cup \{(\tau, l)\}$ or $S \leftarrow S \cup \{(-\tau, l)\}$
6. $M \leftarrow M \times l$
7. $l \leftarrow nextprime(l)$
8. **Recover t using the set S and CRT**
9. **Return** $q+1-t$

2.3 SEA 알고리즘

SEA(Schoof-Elkies-Atkin)는 Schoof 알고리즘과 유사하지만 $\varphi^2(P) + [q_l]P = [\lambda]\varphi(P)$을 만족하는 점을 조사하는데 있어 Elkies와 Atkin 소수를 사용하여 좀 더 효율적인 방법을 제시한 알고리즘이다. Elkies 소수인 경우에는 정확한 t_l을 구하지만, Atkin 소수인 경우에는 t_l의 후보 집합을 얻게 된다.

Schoof 알고리즘은 characteristic과 상관없이 동일한 계산량으로 적용될 수 있는 반면, SEA 알고리즘은 characteristic이 큰 소수인 경우에 효율적인 것으로 알려져 있다. 이 논문에서는 characteristic 2인 경우를 다루고 있으므로, 이 경우를 중심으로 기술하기로 한다. 먼저 Elkies와 Atkin 소수의 정의는 다음과 같다.

[정의 1]

방정식 $F_l(u)=u^2-t_l\cdot u+q_l=0$의 근의 $GF(l)$에서의 존재 여부에 따라 정의된다.

(1) l : Elkies 소수 ⇔ 위 방정식의 근이 $GF(l)$에 존재, 즉, $\Delta_t\equiv t^2-4\cdot q \pmod l$이 $GF(l)$에서 제곱수임

(2) l : Atkin 소수 ⇔ 위 방정식의 근이 $GF(l)$에 존재하지 않음, 즉, $\Delta_t\equiv t^2-4\cdot q \pmod l$이 $GF(l)$에서 제곱수가 아님

t 혹은 t_l의 값을 알고있다면, 주어진 l이 어느 타입인지 결정할 수 있지만, SEA알고리즘의 목표가 t의 값을 구하는 것임을 상기해보면, t에 대한 정보 없이 l의 타입을 결정해야 한다. 이 때, 다음의 정리가 사용된다.

[정리 3][11, p.119]

Non-supersingular 타원곡선 E의 j-invariant를 j라 하자. l번째 modular 다항식 $\Phi_l(x,j)\in GF(q)[X]$에 대하여, $h_1\cdots h_s$를 $\Phi_l(x,j)\in GF(q)[X]$의 기약다항식의 곱이라 하자. $h_1,\ldots,h_s$의 차수는 다음 경우 가운데 하나가 성립한다.

(1) $1, l$ 이거나 모두 1 (이 경우는 $\Delta_t\equiv t^2-4\cdot q \pmod l\equiv 0$ 일 때 발생)

(2) $1,1,r,\ldots,r$ 이다. 이는 $\Delta_t\equiv t^2-4\cdot q \pmod l$이 $GF(q)$에서 0이 아닌 제곱수일 때 발생하며, 이 때, $r\mid l-1$ 이 성립한다.

(3) $r,\ldots,r$, $(r>1)$이다. 이는 $\Delta_t\equiv t^2-4\cdot q \pmod l$이 $GF(q)$에서 제곱수가 아닐 때에 발생하며, $r\mid l+1$ 이 성립한다.

(정의 1)와 (정리 3)에 의하여 Elkies 소수를 판단하는 다음의 정리를 얻는다.

[정리 4][11]

(1) $\gcd(x^q-x,\Phi_l(x,j))$의 차수가 $0\Leftrightarrow l$은 Atkin 소수이다.

(2) $\gcd(x^q-x,\Phi_l(x,j))$의 차수가 1,2 또는 $l+1$ $\Leftrightarrow l$은 Elkies 소수이다.

따라서, 소수 l의 타입을 결정하기 위해 $\gcd(x^q-x,\Phi_l(x,j))$을 계산한다.

(Remark 1)

$\gcd(x^q-x,\Phi_l(x,j))$을 계산하는데 있어, q의 값이 크므로, 먼저 x^q을 $\Phi_l(x,j)$로 reduce한 후, 계산해야 한다. 따라서, 많은 시간이 소요된다.

SEA 알고리즘[11]

1. $M\leftarrow 1$, $l\leftarrow 2$, and $A\leftarrow\{\}$ and $E\leftarrow\{\}$
2. **While** $M<4\sqrt{q}$, *do* :
3. **Decide whether** l **is an Atkin or Elkies prime, by finding splitting type of the modular polynomial** $x=0,\ldots,(l-1)/2$, *do* :
4. **If** l **is Elkies then do** :
5. **Determine** $F_l(\lambda)$
6. **Find an** $\lambda \bmod l$ **of** $F_l(\lambda)$
7. $t\leftarrow\lambda+q/\lambda \pmod l$
8. $E\leftarrow E\cup\{(t,l)\}$
9. **Else do** :
10. **Determine a (small) set** T **such that** $t \pmod l\in T$
11. $A\leftarrow A\cup\{(T,l)\}$
12. $M\leftarrow M\times l$
13. $l\leftarrow nextprime(l)$
14. **Recover** t **using the sets** A **and** E, **the CRT, BSGS**
15. **Return** $q+1-t$

이제, (정리 4)를 이용하여, l이 Elkies 소수로 판단 되었다고 가정하자. SEA 알고리즘에서 방정식 $F_l(u)=u^2-t_l\cdot u+q_l=0$의 두 근을 알면 근과 계수와의 관계를 이용하여 t_l을 구할 수 있다. 논문의 전개를 위해 SEA 알고리즘에서 Elkies 소수인 경우 t_l을 구하는 방법을 간단히 살펴보면 다음과 같다.

[정리 5][11, p.121]

l을 Elkies소수이고, $F_l(u)=u^2-t_l\cdot u+q_l=0$의 두 근을 λ, μ 라 하자. 특히, $\lambda\neq\mu$ 으로 가정하자. 즉, (정리 3)의 (2)에 해당하는 경우이다. 그러면 $\varphi(P_1)=[\lambda]P_1$을 만족하는 P_1과, $\varphi(P_2)=[\mu]P_2$을 만족하는 P_2가 $E[l]$에 존재하며, $C_1=\langle P_1\rangle$, $C_2=\langle P_2\rangle$로 놓으면, $E[l]=C_1\times C_2$이 된다. 또한, $\varphi(C_i)=C_i$가 성립한다.

(정리 5)에 의해, $F_l(x)$이 서로 다른 두 근을 갖는 Elikes 소수인 경우에는 $E[l]=C_1\times C_2$라 하자. C_i를 찾아서 $\varphi(P)=[\lambda]P$이 성립하는 $P\in C_i$이 존재하는 λ를 찾으면 된다. 또한, 두 근 중 하나만 알면 $t_l=\lambda+\frac{q_l}{\lambda}\pmod l$을 구할 수 있으므로 우리가 사용하는 C가 C_1, C_2 중에 어느 것이든 상관 없음을 알 수 있다. 따라서, λ를 구하는 방법은 다음과 같다.

Step 1 Elkies 소수 l에 대하여, 다음을 만족하는 다항식 $Q(x)$를 구한다.

$$Q(x)=\prod_{+P_i\in C\setminus O} x-(P_i)_X$$

Step 2. $0\le\tau\le\frac{l-1}{2}$에 대하여

Step 2.1 $h(x)=x^q+x+\frac{f_{\lambda-1}f_{\lambda+1}}{f_\lambda^2}$을 계산한다.

Step 2.2 $\gcd(h(x),Q(x))$ 을 구하여 차수가 0이면 다음 λ에 대하여 (step 2.1)로 돌아간다.

Step 2.3 타원곡선 $y^2=x^3+xy+a$ 을

$$y^q=x+y+\frac{(x^2+x+y)f_{\lambda-1}f_\lambda f_{\lambda+1}+f_{\lambda-2}f_{\lambda+1}^2}{xf_\lambda^3}$$

에 대입하여 $a(x)+b(x)y=0$ (1)

의 꼴로 정리한다. 식 (1)을 타원곡선 방정식에 대입하여 $h(x)=0$의 식을 얻는다.

Step 2.4 $\gcd(h(x),Q(x))$의 차수가 1 이상이면 $\lambda=\tau$ 를 출력하고 차수가 0이면 $\lambda=-\tau$ 를 출력한다.

Ⅲ. 랜덤 커브 생성 알고리즘 : Early-abort 전략

Characteristic 2인 유한체 위에서 정의된 타원곡선의 위수 계산에서 가장 효율적인 알고리즘은 서론에서 기술하였듯이 Satoh 알고리즘이다. 그러나, 암호학적으로 안전한 타원곡선을 얻기 위해서는 랜덤한 커브를 생성한 후, 위수를 계산하여, 위수의 조건이 만족하는지 검증하여 그렇지 않으면 다시 새로운 커브에 도전해야 한다. 이는 하나의 타원곡선을 얻기 위해서는 상당히 많은 Satoh 알고리즘을 적용해야 함을 의미한다. 따라서, Satoh 알고리즘을 적용하기 전에, 작은 소수에 대해 보다 간단히 test 할 수 있는 방법이 있다면 이를 적용하여 작은 소수를 위수의 인수로 갖는 타원곡선에 대해서는 Satoh 알고리즘을 적용하기 전에 abort 하고, 다시 새로운 랜덤 커브에 대해 조사하는 것이 효율적일 것이다. 바로 이러한 방법이 [1,10]에 발표된 early-abort 전략이다. Schoof 알고리즘은 소수 l마다 t_l의 값을 계산하므로, 단계마다 $q_l+1-t_l\pmod l$의 값을 알 수 있다. 따라서, $q_l+1-t_l\equiv 0\pmod l$ 이면 abort하고 아니면 다음 소수로 옮겨가면 된다. Schoof 알고리즘보다는 SEA 알고리즘이 효율적이므로 SEA 알고리즘을 적용하는 것이 효율적이다. 그러나, SEA 알고리즘에서는 Elkies 소수인 경우에는 정확한 t_l을 계산하지만, Atkin 소수인 경우는 t_l의 후보를 결정하게 될 뿐, 정확한 값을 얻지 못할 수 있다. 그러나, 아래 4.1절의 (보조정리 2])로부터 Atkin 소수인 경우에는 $q_l+1-t_l\not\equiv 0\pmod l$ 임을 알 수 있다. 따라서, SEA 알고리즘이 early-abort 전략에 사용된다. SEA 알고리즘을 사용한 early-abort 전략 알고리즘의 개요는 다음과 같다.

SEA 알고리즘을 이용한 early abort 전략

Step 1 랜덤한 타원곡선을 선택한다.

Step 2 작은 소수 l($l=3,\ldots,19$)에 대하여 다음을 수행한다.

Step 2.1 SEA알고리즘을 사용하여 t_l을 계산한다.

Step 2.2 $q+1-t_l\equiv 0 \bmod l$인지 판별 하여 $q+1-t_l\equiv 0 \bmod l$이면 다시 (step 1)로 돌아가고, 아니면 다음 소수에 대하여 (step 2.1)로 돌아간다.

Step 3 Satoh 알고리즘을 사용하여 전체 위 수를

계산한다.

Step 4 위 단계에서 계산한 위수가 암호학적으로 안전한지 판단하여 실패하면 (step 1)로 돌아간다.

Step 5 타원곡선을 출력한다.

(Note 2)

l이 크면 characteristic 2인 경우, t_l계산이 효율적이지 못하다. 따라서, [1]에서는 19이하 정도로 test할 것을 권고하고 있다.

Ⅳ. Speed up early-abort strategy

이 절에서는 characteristic 2인 타원곡선을 생성하는 알고리즘의 효율적인 방법에 대해 논의한다. 앞에서 언급하였듯이 현재까지 알려진 가장 빠른 알고리즘은 작은 소수에 대해 SEA 알고리즘을 적용하여 abort 단계를 거치고, 이를 통과한 곡선에 대해서만 Satoh 알고리즘을 사용하여 위수를 계산한 후, prime test를 거쳐 안전한 타원곡선을 생성하는 방법이다. 이러한 early-abort 방법이 효율적인 것으로 주목받는 이유 가운데 하나는 abort 단계에서 많은 곡선들이 버려진다는 사실이다. 따라서, abort 단계에서 좀더 효율적인 방법을 사용하면, 전체 알고리즘에 효율성을 증가시키게 될 것이다. 이 절에서는 abort 단계에서의 시간을 단축할 수 있는 방법을 제안한다.

4.1 Observations

이 절에서는 몇 개의 보조정리들을 살펴보기로 한다.

[보조정리 1]

타원곡선 $y^2=x^3+xy+a$ 와 3 이상 소수 l에 대하여 $q_l+1-t_l\equiv 0 \bmod l$ 라 하자. 그러면 1과 q_l이 $\lambda^2-t_l\lambda+q_l\equiv 0 \pmod l$의 두 근이 된다.

[증명]

$\lambda^2-t_l\lambda+q_l\equiv 0 \pmod l$은 체 위에서 정의된 방정식이므로 두 개의 근이 존재한다. 한편, $1 \pmod l$을 $\lambda^2-t_l\lambda+q_l\equiv 0 \pmod l$에 대입하면 식이 성립함을 알 수 있고, 근과 계수와의 관계와 조건 $q_l+1-t_l\equiv 0 \bmod l$을 이용하면 q_l이 근이 됨을 알 수 있다. ■

다음의 보조정리는 잘 알려진 정리로써, Elkies 소수의 정의로부터 쉽게 유도된다.

[보조정리 2]

타원곡선 $y^2=x^3+xy+a$ 와 3이상 소수 l에 대하여 $q_l+1-t_l\equiv 0 \bmod l$이라 하자. 그러면 l은 Elkies 소수이다.

[증명]

(보조정리 1)에 의하여 1과 q_l이 두 근이 된다. $\lambda^2-t_l\lambda+q_l\equiv 0 \pmod l$ 의 두근이 $GF(l)$의 원소이므로, ${t_l}^2-4\cdot q_l \pmod l$은 $GF(l)$에서 제곱수가 된다. 따라서, 정의에 의해 Elkies 소수이다. ■

[보조정리 3]

주어진 타원곡선 $y^2=x^3+xy+a$와 Elkies 소수 l이 주어져 있다고 가정하자. $q_l+1-t_l\equiv 0 \bmod l$이면 $\varphi(P_1)=P_1$ 인 $O\neq P\in E[l]$와 $\varphi(P_2)=[q_l]P_2$ 인 $O\neq P_2\in E[l]$존재하며, $E[l]=\langle P_1\rangle\times\langle P_2\rangle$가 된다.

[증명]

(보조정리 1)과 (정리5)로부터 자명하다. ■

4.2 SEA 알고리즘 modifications

이 절에서는 early-abort 전략에서 사용할 SEA 알고리즘의 변형 방법에 대해 살펴보기로 한다. 먼저, characteristc 2인 경우의 SEA 알고리즘으로부터 변형 없이 사용되는 알고리즘은 다음과 같다.

<u>Original SEA 알고리즘</u>

주어진 타원곡선 $y^2=x^3+xy+a$ 와 각 소수 l에 대한 original SEA 알고리즘의 개요는 다음과 같다. 이 때, 이 곡선의 j-invariant를 j라 하자.

Step 1 타원곡선의 l번째 modular 다항식 $\Phi_l(x,j)$의 근이 $GF(q)$에 존재하는지 조사한다. 존재하지 않으면, l은 Atkin 소수이므로, $q_l+1-t_l\neq 0 \bmod l$ 이다. 따라서, 다음 소수에 대한 SEA 알고리즘을 적용한다.

Step 2 위 단계에서 근이 존재하는 경우, 근을 구하여, 근에 의존하는 다항식 $Q(X)$를 계산한다. ($Q(X)$의 계산방법은 [5] 참고)

Step 3 $Q(X)$을 만족하는 타원곡선 위의 점들 가운데, 각, τ($\tau=0,1,\ldots,(l-1)$)에 대하여 $\varphi(P)=[\tau]P$를 만족하는 P가 존재하는지 조사하여 해당 τ를 찾는다. 이 때, τ를 eigenvalue로 부르기로 하자.

Step 4 $t_l = \tau + q/\tau \pmod l$을 계산하여 얻는다.

Step 5 위 단계에서 얻은 t_l을 이용하여 $q+1-t \equiv 0 \pmod l$인지 조사한다. 일치하면 해당 곡선에 대해 abort하고, 아니면 다음 소수에 대해 SEA 알고리즘을 적용한다.

[보조정리 4]

m과 l을 소수로서, $m \nmid l-1$ 이라 하자. 그러면, $q=2^m \not\equiv 1 \pmod l$이 성립한다.

[증명]

r을 $GF(l)$에서 2의 위수라 하자. 이 때, $2^m \equiv 1 \pmod l$ 이면 $r \mid m$ 이다. 또한, $r \mid l-1$이 성립한다. 따라서, $r \mid \gcd(m, l-1)$이다. 그리고, $\gcd(m, l-1) \mid m$이 성립한다. 한편, m은 소수이므로, $\gcd(m, l-1)=1$ 또는 m이다. 2의 위수가 1이 아니므로, $\gcd(m, l-1) \neq 1$이며, $\gcd(m, l-1)=m$은 $m \nmid l-1$ 라는 가정에 위배된다. 따라서, $q=2^m \not\equiv 1 \pmod l$이 성립한다.

(Note 3)

타원곡선에서 유한체의 크기를 결정하는 m에 대하여 l은 비교적 작은 소수이다. 따라서, $m \nmid l-1$이며, 암호학적 안전성 관점에서 m이 소수인 타원곡선 암호을 고려할 때, $q_l \neq 1$임을 알 수 있다. 이러한 m과 l은 우리가 랜덤 커브 생성 알고리즘과 이에 사용되는 early-abort 전략에서 사용하는 파라미터들의 범위를 포함한다.

[정리 6]

m을 소수라 하자. 타원곡선 $y^2=x^3+xy+a$의 $\Phi_l(x,j)$에 대하여, $\gcd(\Phi_l(x,j), x^q+x)$의 차수가 2가 아니면, $q_l+1-t_l \neq 0 \pmod l$이 성립한다.

[증명]

$\gcd(\Phi_l(x,j), x^q+x)$의 차수가 2가 아니라하자. (정리 3)와 (정리 4)에 의하여, 이 경우는 (정리 3)의 (1) 또는 (3)의 경우이다. 한편 (3)은 Atkin 소수이므로, (보조정리 2)에 의하여 $q_l+1-t_l \neq 0 \pmod l$이다. 따라서, (1)의 경우라 하자. (정리3)에 의하여 $\Delta_t \equiv t^2-4 \cdot q \pmod l \equiv 0$은 0이므로 $F_l(\lambda)$가 중근을 갖는다. 한편, $q_l+1-t_l=0 \pmod l$이면, $F_l(\lambda)$의 두 근은 1, q_l이다. 그러나, (보조정리 4)에 의하여 $q_l \neq 1 \pmod l$이므로, $\Delta_t \equiv t^2-4 \cdot q \pmod l \equiv 0$와 $q_l+1-t_l=0 \pmod l$을 동시에 만족시킬 수는 없다. 따라서, 정리가 성립한다. ■

위 사실들로부터 original SEA 알고리즘을 다음과 같이 변형하여 사용할 수 있다.

(방법 1)

(1) $q+1-t \equiv 0 \pmod l$을 Elkies 소수 가운데, $\gcd(\Phi_l(x,j), x^q+x)$의 차수가 2인 경우, 즉, $F_l(\lambda)$이 서로 다른 두 근을 갖는 경우에만 확인하면 된다.

(2) (보조정리에 3) 의해 abort 단계에서 사용할 SEA는 위 알고리즘 (step 3)에서 τ이 1 또는 q_l인 경우에만 조사하고, step 4,5 단계는 생략한다.

(detail of step 3 in original SEA)

$Q(X)$을 만족하는 타원곡선 위의 점들 가운데, 각, τ($\tau=0,1,\ldots,l-1$)에 대하여 $\varphi(P)=[\tau]P$를 만족하는 P가 존재하는지 조사하는 방법의 자세한 기술은 다음과 같다.

먼저, division 다항식 공식을 이용하여 $[\tau]P$을 계산하여, $\varphi(P)=(x^q, y^q)$와 일치하는 값 가운데, $Q(X)$를 만족하는지 조사하는데, 구체적 방법은 다음과 같다.

Step 1 τ에 따라 귀납적으로 division 다항식들을 구성하여 $f_{\tau-2}$, $f_{\tau-1}$, f_τ, $f_{\tau+1}$을 얻는다. (관계식은 [11] 참고)

Step 2 $F(x)=x^q \pmod{Q(X)}$을 계산한다.

Step 3 다음과 같이 $G(x)$을 계산한다.

$$G(x)=\gcd\left((F(x)+x)\, f_\tau^2 + f_{\tau-1}f_{\tau+1},\ Q(X)\right)$$

$G(x)$의 차수가 0이면 τ는 eigenvalue가 아니라고 판단하고, 차수가 1이상이면 다음 단계를 계속 수행한다.

Step 4 타원곡선 $y^2=x^3+xy+a$을 이용하여 다음을 만족하는 두 식 $a(x)$, $b(x)$을 얻는다.

$$y^q = a(x) + b(x)y$$

Step 5 다음과 같이 $h(x,y)$를 계산한다.

$$h(x,\ y) = x f_\tau^3 (y + a(x) + b(x)y) + f_{\tau-2} f_{\tau+1}^2 + (x^2+y) f_{\tau-1} f_\tau f_{\tau+1}$$

$h(x,y)=0$을 $G(x)$로 reduce 시켜 다음을 얻는다.

$$\overline{a(x)} + \overline{b(x)}y = 0 \qquad (2)$$

식 (2)와 타원곡선 방정식을 이용하여

$$\overline{h}(x) = \overline{a}(x)^2 + \overline{a}(x)\,\overline{b}(x)y + (x^3+a)\,\overline{b}(x)^2$$

을 계산한다.

Step 6 $\gcd(\overline{h}(x),\ Q(x))$의 차수가 0이면, τ가 eigenvalue 이고, 차수가 1이상이면, $-\tau \pmod l$을 eigenvalue로 판단한다.

(Note 4)

(1) (detail of step 3 in original SEA)의 (step 5)에서 사용한 $h(x,y)$가 $[-\tau]P$의 y좌표를 이용한 것이므로, step 6에서 출력 값의 순서가 바뀌게 된다.

(2) (방법 1)에서 $\tau = q_l$인 경우, $q_l > \frac{l-1}{2}$일 수 있으므로, $\tau = -q_l \pmod l$에 대하여 조사하면 된다. 따라서, $\overline{q_l}$을 다음과 같이 정의하여 사용한다.

- $q_l > \frac{l-1}{2}$인 경우, $\overline{q_l} = -q_l \bmod l$
- $q_l \le \frac{l-1}{2}$인 경우, $\overline{q_l} = q_l \bmod l$

Division 다항식들이 귀납적으로 생성되는 점과 f_τ의 차수가 $O(\tau^2)$인 것을 고려하면 $\overline{q_l}$의 값이 작은 경우, 효율적일 것으로 예상되며, $\tau=1$인 경우엔 좀 더 빠르게 처리될 것이다. 따라서, 다음의 방법을 생각할 수 있다.

(방법 2)

(방법 1)과 유사하게 수행하되, $\Phi_l(x,j)$의 $GF(q)$에서의 서로 다른 근에 대하여 각각 $Q_1(x), Q_2(x)$를 계산하여 각각에 대하여 original SEA 알고리즘 (step 3)에서 $\tau=1$에 대하여 조사한다. 이 방법의 효율성은 $Q_i(x)$ 계산과 각각의 경우에 대하여 original SEA 알고리즘의 (step 3)에 소요되는 시간에 의존하게 된다.

(방법 3)

(방법 1)에서 $\tau=1$부터 $\tau=\overline{q_l}$까지 차례로 조사하는 방법으로, (방법 1)에서 $\overline{q_l}$보다 작은 값이 eigenvalue인 경우 (방법 1)보다 1부터 차례대로 조사하는 것이 효율적일 가능성도 있다.

위에서 기술한 방법들 가운데, 효율적인 방법을 찾기 위하여, 다음을 조사하여야 한다.

(1) $\Phi_l(x,j)$의 $GF(q)$에서의 근에 대하여 근에 따른 $Q(x)$를 계산하는데 걸리는 시간
(2) (detail of step 3 in original SEA)에서 division 다항식 계산 시간
(3) (detail of step 3 in original SEA)에서 Step 2 : $F(x) = x^q \pmod{Q(X)}$ 계산 시간
(4) (detail of step 3 in original SEA)에서 Step 3 : $\gcd((F(x)+x) f_\tau^2 + f_{\tau-1} f_{\tau+1},\ (X))$ 계산 시간
(5) (detail of Step 3 in original SEA)에서 step 4, 5, 6 계산 시간

위 항목 가운데, (3)은 모든 τ에 대하여 계산할 필요가 없다. 그러나, 반드시 한번은 계산해야 한다. 한편, $F(x) = x^q \pmod{Q(X)}$은 $Q(x)$에 의존하므로, (방법 2)를 적용할 경우, 각각의 $Q(x)$에 대하여 계산되어져야 한다. 따라서, (방법 2)는 $Q(x)$시간 뿐 아니라, $F(x) = x^q \pmod{Q(X)}$ 계산 시간에도 의존한다. 그리고, step 4, 5, 6은 주어진 곡선과 l에 대하여 한번 만 거치게 된다. (왜냐하면, 주어진 $Q(x)$에 대하여 eigenvalue는 단 하나만 존재) 만약, eigenvalue에 해당하는 τ가 $\tau \neq \pm 1$이고, $\tau \neq \pm \overline{q_l}$ 인 경우, step 4, 5, 6은 생략할 수 있다. 따라서, $\tau=1$, -1, $q_l, -q_l$일 때에만 step 4, 5, 6을 구현하는 것이 효율적일 것이다.

4.3 구현 결과 및 분석

앞절에서 기술한 경우에 대한 시간 측정 결과는 아래 표들과 같다. 이는 free 소프트웨어인 miracl

을 사용하여 구현한 결과를 이용해 측정한 것이다. 시간 측정 함수는 C++의 clock() 함수를 사용하였다. 단위는 생략한다.

우리는 $m=163,193,233,239$ 인 경우에 대해 조사하였으며, 모든 경우에 대해 $l=3$인 경우는 생략하였다.

(Note 5)

Division 다항식의 계산은 100을 넘지 않았으며, 보통은 10~30 정도가 대부분이었다. 따라서, 특별히 고려하지 않아도 될 것으로 보이며, 아래 표에서는 생략하였다.

아래 표들의 구분 항목은 다음을 가리킨다.

1 : $x^q+x \pmod{\Phi_l(x,j)}$ 계산 시간
2 : $Q(x)$ 계산 시간
3 : $F(x)=x^q \pmod{Q(x)}$ 계산 시간
4 : $\tau=1$일 때, step 4,5,6 계산 시간
5 : $\tau=\overline{q_l}$일 때, step 4,5,6 계산 시간
6 : $\overline{q_l}$의 값

아래표의 각 항목의 단위는 CLOCKS_PER_SEC와 sec단위의 시간을 곱한 값이며, 이 논문의 경우 CLOCKS_PER_SEC의 값은 1000 이다.

〔표 1〕 m=163 인 경우

구분	5	7	11	13	17	19
1	250	350	681	871	1042	1262
2	0	0	10	10,20	30	30
3	60	100	210	290	440	541
4	401	721	1582	2963	3315	3335
5	501	731	1883	2073	3745	3265
6	2	2	3	2	8	2

〔표 2〕 m=193 인 경우

구분	5	7	11	13	17	19
1	281	341	841	1062	1282	1543
2	0	0	10	10	30	30
3	70	120	251	341	540	661
4	525	871	1883	2554	4236*	4006*
5	500	901	1902	2664	4216	4116
6	2	2	3	2	2	3

〔표 3〕 m=233 인 경우

구분	5	7	11	13	17	19
1	361	451	1071	1352	1653	2063
2	0	0	10,20	10,20	40	40
3	90	150	330	441	691	841
4	611	1082	2374	3214	5178	5137
5	620	1112	2404	3455	5218	5698
6	2	3	3	6	2	92

〔표 4〕 m=239 인 경우

구분	5	7	11	13	17	19
1	371	451	1100	1392	1692	2013
2	0	0	10,20	10,20	30,40	30,40
3	90	150	330	441	711	891
4	621	1112	2434	3285	5328	5438
5	631	1142	2524	3445	5748	5508
6	2	3	5	6	8	6

Test 결과

- 시간이 가장 많이 소요되는 부분은 step 4,5,6 이었음을 알 수 있었다.
- gcd $((F(x)+x)f_\tau^2+f_{\tau-1}f_{\tau+1}, Q(X))$의 계산 시간은 τ의 값이 작을수록 빠름을 알 수 있었다. 그러나, 그 값의 차이는 그리 크지 않았다. 따라서, 평균적으로는 τ를 1부터 $\overline{q_l}$까지 차례로 하는 것이나, 1과 $\overline{q_l}$만 계산하는 것이나 별 차이가 없을 것으로 예상한다. 즉, (방법 1)과 (방법 3)은 별 차이가 없을 것으로 기대한다.
- 예상보다, $Q(x)$ 계산 시간은 길지 않았다. $Q(x)$의 계산 시간은 차원보다는 소수 l에 더 의존함을 알 수 있었다. 또한, 이 시간은 step 4,5,6과 비교하여 무시할만한 시간임을 알 수 있었다.
- $\overline{q_l}$가 크지 않은 경우, $\tau=1$인 경우나, $\tau=\overline{q_l}$인 경우 모두 step 4,5,6,의 소요 시간은 비슷하였다.
- x좌표를 비교하는 단계 (즉, gcd $((F(x)+x)f_\tau^2+f_{\tau-1}f_{\tau+1}, Q(X))$ 계산)에서의 시간은 step 4,5,6에 비해 무시할 만하다. 거의(division 다항식 계산 시간+10~20) 정도 소요됨을 알 수 있었다.
- $x^q+x \bmod \Phi_l(x,j)$의 계산 시간은 소수의 값이 클수록, q의 값이 클수록 증가함을 알 수 있다. q의 증가는 for문의 크기가 증가하므로, 이에

따른 reduction 계산의 회수가 증가함에 따른 것이고, l의 값에 의존하는 것은 reduction에서 modular하는 다항식의 차수가 영향을 미침을 의미한다. 이는 $F(x)=x^q \pmod{Q(x)}$의 계산 시간이 l이 증가할수록 커지는 이유와 동일한다. 실제 $Q(x)$의 차수는 $(l-1)/2$ 이다.

(Note 6)

Step 4,5,6에서 계속해서 $Q(x)$로 reduce시키면서 계산해야 한다. 그러나, 위 표들의 data를 보면 $F(x)=x^q \pmod{Q(x)}$와 $\tau=1$일 때, step 4,5,6의 시간차이가 두드러지지 않음을 볼 수 있다 (표에서 *부분 참고). 이 사실은 $Q(x)$의 차수가 증가함에 따른 reduction 계산 시간이 증가해야 한다는 사실과 모순처럼 보인다. 그러나, 실제 계산에서, 우리는 $\gcd(\overline{h(x)}, Q(x))$을 계산하기 위해, $Q(x)$로 reduce하지 않고, x좌표 비교에서 얻은 $G(x)$으로 reduce하였다. 실제로, $G(x)=\gcd((F(x)+x)f_\tau^2+f_{\tau-1}f_{\tau+1}, Q(X))$의 근 가운데, $\gcd(\overline{h(x)}, Q(x))$의 근이 존재하는지 확인하는 것이므로, $G(x)$으로 reduce해도 된다. 이 때, $G(x)$의 차수가 $Q(x)$의 차수보다 크지 않으므로, 이 방법이 훨씬 효율적이다.

따라서, 우리는 위 test 결과 및 분석을 바탕으로 다음과 같이 SEA 알고리즘을 변형하여 사용할 것을 제안한다.

Early-abort 전략에서의 SEA 변형 알고리즘

Step 1 타원곡선의 j-invariant j를 계산하여, l 번째 modular 다항식 $\Phi_l(x,j)$의 근이 $GF(q)$에 서로 다른 두 근으로 존재하는지 조사한다. 즉, $\gcd(\Phi_l(x,j), x^q+x)$의 차수가 2일 때에만 다음 단계로 진행한다. 그렇지 않으면 다음 소수에 대해 (step 1) 돌아간다.

Step 2 위 단계에서 근이 존재하는 경우, 근을 구하여, 근에 의존하는 다항식 $Q(x)$를 계산한다.(계산방법은 참고 문헌 [5] 참고)

Step 3 $F(x)=x^q \pmod{Q(x)}$을 계산한다.

Step 4 $\tau=1$에 대해 다음을 조사한다.

Step 4.1 $\gcd(F(x)+x, Q(x))$의 차수가 0이면 (step 5)로 이동한다.

Step 4.2 $\gcd(F(x)+x, Q(x))$의 차수가 1이상이면 step 4,5,6 계산 후, 리턴 값이 1이면 test 곡선은 버리고, 알고리즘을 종료한다. 리턴 값이 -1이면 다음 소수에 대해 (step 1)로 돌아간다.

Step 5 $\tau=\overline{q_l}$에 대해 다음을 조사한다.

Step 5.1 $\gcd((F(x)+x)f_\tau^2+f_{\tau-1}f_{\tau+1}, Q(X))$의 차수가 0이면 다음 소수에 대해 (step1)로 돌아간다.

Step 5.2 $\gcd((F(x)+x)f_\tau^2+f_{\tau-1}f_{\tau+1}, Q(X))$의 차수가 1이상이면 step 4,5,6 계산하여, 리턴 값 $\tau=q_l$이면 test 곡선은 버리고 알고리즘을 종료한다. $\tau=-q_l$이면 다음 소수에 대해 (step 1)로 돌아간다.

(Note 7)

step 4,5,6에서 $\gcd(\overline{h(x)}, Q(x))$을 계산할 때, $G(x)$로 reduce한다.

V. 결 론

우리는 이상에서 characteristic 2인 유한체 위에서 정의된 암호학적으로 안전한 타원곡선 생성 알고리즘 가운데, SEA 알고리즘과 Satoh 알고리즘을 결합한 알고리즘 사용시, early-abort 전략에 해당하는 SEA 알고리즘의 개요 및 이의 효율적 변형 방법에 대해 알아보았다.

구현을 통하여, l이 작은 소수인 경우, 실제 계산에서 시간이 가장 소요되는 부분이 $Q(x)$계산 단계가 아니고, y좌표를 비교하는 step 4,5,6임을 알 수 있었다. Original SEA 알고리즘은 eigenvlaue를 찾는 것이었으므로, step 4,5,6을 항상 거치게 되는데, 변형한 알고리즘에서는 상당한 경우 생략될 수 있음을 시사한다. 향 후, Satoh 알고리즘과 결합하여 구현한 결과를 테스트하여, abort 단계에서 어느 정도의 곡선의 버려지는지 조사하는 것과, 또한, step 4,5,6이 생략되는 빈도 수를 조사하여 제안 알고리즘의 효율성을 검증하는 것도 의미가 있을 것으로 생각된다.

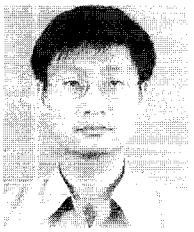
참 고 문 헌

[1] M. Fouquet, P. Gaudry, and R. Harley, Finding Secure Curves with the Satoh-

FGH Algorithm and an Early-Abort Strategy, *Springer-Verlag*, Advaces in Cryptology-Eurocrypt'01, LNCS 2045, 14-29, 2001.
[2] R. Schoof, Counting points on elliptic curves over finite fields. J. Theory. Nombres Bordeaux, Vol. 7, 219-254, 1995.
[3] N. Elkies, Elliptic and modular curves over finite fields and related computational issues, Computational Perspectives on Number Theory, 21-76, 1998.
[4] J. M. Couveignes, Computing l-isogenies with the p-torsion, *Springer-Verlag*, ANTS-II, Lecture Notes in Comp. Sci., Vol. 1122, 59-65, 1996.
[5] R. Lecier, Computing isogenies in $GF(2^n)$, *Springer-Verlag*, ANTS-II, Vol. 1122, Lecture Notes in Comp. Sci., 197-212, 1996.
[6] T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, J. Ramanujan Math. Soc., Vol. 15, 247-270, 2000.
[7] M. Fouquet, P. Gaudry, and R. Harley, An extension of Satoh's algorithm and its implementation, J. Ramanujan Math. Soc., Vol. 15, 281-318, 2000.
[8] B. Skjernaa. Satoh's algorithm in characteristic 2. Copies available at http://www.imf.au.dk/~skjernaa/.
[9] F. Vercauteren, B. Preneel, and J. Vandewalle, A memory Efficient Version of Satoh's Algorithm, *Springer-Verlag*, Advances in Cryptology-Eurocrypt'01, LNCS 2045, 1-13, 2001.
[10] R. Lercier, Finding good random elliptic curves for cryptosystems defined over F_{2^n}, *Springer-Verlag*, Advances in Cryptology-Eurocrypt'97, Vol. 1233 of LNCS, 379-392, 1997.
[11] I. F. Blake, G. Seroussi, N. P. Smart, *Elliptic Curves in Cryptography*, *Cambridge university press*, LMSLNS. 265. 1999.

〈著者紹介〉

정 배 은 (Bae Eun Jung) 정회원
1993년 2월 : 서울대학교 수학교육과 학사
1995년 2월 : 서울대학교 수학과 석사
2000년 2월 : 서울대학교 수학과 박사
2000년 4월~현재 : 한국전자통신연구원 선임연구원
〈관심분야〉 암호이론, 이동통신 정보호호, 가환대수



류 희 수 (Heuisu Ryu) 정회원
1990년 2월 : 고려대학교 수학과 학사
1992년 2월 : 고려대학교 수학과 석사
1999년 5월 : Johns Hopkins University 수학과 박사
2000년 7월~현재 : 한국전자통신연구원 선임연구원
〈관심분야〉 정보보호, 타원곡선 암호, 이동통신 보안