한국학교수학회논문집 제 3권, 제 2호
Journal of the Korean School Mathematics Society
Volume 3, Number 2, 165-176, December 2000

Javascript를 이용한 타원곡선 암호 연습 프로그램

김 승 동[1] · 정 상 조[2]

I. 서 론

현대에는 컴퓨터의 발달에 따라 은행업무, 전자상거래와 메일 주고받기 등 국가간의 전쟁이 아닌 곳에서도 암호의 사용이 아주 많이 사용하게 되었다. 인수분해의 어려움을 이용한 RSA암호는 사용하는 숫자의 자리수가 너무 많아 계산하는 데 시간이 너무 많이 걸리므로 이를 대용할 방법으로 RSA 암호(1024비트 이상)보다 적은 비트(160비트) 정도로 더 좋은 보안 효과를 가져 올 수 있는 타원곡선 암호는 요즈음 각광을 받고 있는 방법이다.(박창섭)

타원곡선 암호는 1985년 Neil Koblitz, Victo miller가 소개한 이래 암호이론에서 RSA 암호보다 짧은 키를 사용할 수 있고, ECDLP(Elliptic Curve Discrete Logarithm Problem)인 경우는 아직은 준지수시간 알고리즘(subexponential-time algorithm)이 존재치 않아 무선 통신, 휴대용 컴퓨터, 방송, Smart Card 등에 폭 넓게 이용되고 있다.

본 논문에서는 타원곡선 암호의 이론적인 근거를 언급하고, 이를 이용한 타원곡선 암호 연습을 위한 알고리즘을 준다. 또한, 무료로 제공되고 있는 웹 브라우저(Internet Explorer 4.0 이상 또는 Netscape Communicator 4.0 이상)가 설치된 컴퓨터에서 실행할 수 있는 Javascript로 프로그램을 작성하여, 프로그램의 소스를 알 수 있고 어디서나 암호연습을 할 수 있도록 저자의 홈페이지(http://my.dreamwiz.com/math88)에 올려놓아, 타원곡선 암호를 이해하고자 하는 사람들에게 연습을 통하여 타원곡선 암호의 암호화와 복호화의 과정을 보여 줌으로서 타원암호의 구조와 효율성에 대해 직접 체험할 수 있는 기회를 주고자한다.

먼저 타원곡선 암호에 쓰이는 정리를 소개한 다음, 타원곡선 암호가 어떻게 구성되는가에 대해 언급을 하고, 프로그램의 알고리즘을 작성하고 Javascript로 만든 타원곡선 암호 프로그램을 제공한다.

II. 타원곡선암호에 쓰이는 이론적 배경

타원곡선 암호 시스템은 군과 체의 몇 가지 기본정리가 이용된다.

A. 용어의 정의

(정의 1). [군(group)] 집합 G가 연산 $*$에 대해 닫혀있고, 다음 3가지 조건을 만족하면 $(G,*)$를 **군(group)**이라고 한다.

1) $\forall a,b,c\in G,\ (a*b)*c=a*(b*c)$.
2) $\exists e\in G$, such that $\forall a\in G$, $e*a=a*e=a$.

1) 공주대학교 사범대학 수학교육과
2) 충남대학교 자연과학대학 수학과

3) $\forall a\in G,\ \exists b\in G$ such that
$a*b=b*a=e.$

또한 군 (G,*)가

4) $\forall a, b\in G,\ a*b=b*a$

를 만족하면 **가환군(commutative group)**이라 한다.

(정의 2). [체(Field)] 집합 F가 두 연산 +, · 에 관하여 닫혀있고, 다음 5가지 조건을 만족하면 $(F,+,\cdot)$를 **체(field)**라고 한다.

1) $\exists\ e=0\in F$ such that $\forall a\in F,$
$a+e=e+a=a,$
$\exists\ e'=1\in F$ such that $\forall a\in F,$
$a\cdot e'=e'\cdot a=a.$

2) $\forall a\in F, \exists b\in F$ such that
$a+b=b+a=e=0,$
$\forall 0\neq a\in F, \exists b\in F$ such that
$a\cdot b=b\cdot a=e'=1.$

3) $\forall a,\ b\in F,\ a+b=b+a,$
$a\cdot b=b\cdot a.$

4) $\forall a,\ b,\ c\in F,$
$(a+b)+c=a+(b+c),$
$(a\cdot b)\cdot c=a\cdot(b\cdot c).$

5) $\forall a,\ b,\ c\in F,$
$a\cdot(b+c)=a\cdot b+a\cdot c$

(정의 3). [위수(order)] 어떤 군 G의 원소의 개수를 군 G의 **위수(order)**라 한다. G의 원소에 a에 대해 $a^h=e$(연산이 덧셈이면 $h\cdot a=0$)인 최소의 양의 정수 h가 존재하면 h를 원소 a의 **위수(order)**라고 한다.

(정의 4). [타원곡선(Elliptic curve)] 체 F 위에서의 **타원곡선**(Elliptic curve)이란

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$$

$(a_1,\cdots,a_6\in F)$의 형태로 주어진 곡선을 말한다. 체 F의 표수(characteristic)(즉, 표수는 $\forall a\in F,\ n\cdot a=0$인 가장 작은 양의 정수 n을 말함)가 3보다 큰 경우에는 적당한 좌표 변환을 통하여 $y^2=x^3+ax+b$로 변환할 수 있다. 본 논문에서는 알고리즘의 편의상 체의 표수가 3보다 큰 경우에 대해서만 논의하겠다. 따라서 여기서는 타원곡선이란 주어진 F의 두 원소 a, b에 대한 집합

$$E'=\{(x,y)\in F\times F:\ y^2=x^3+ax+b\}$$

을 말한다. 이 때 x^3+ax+b가 중근을 갖지 않을 때, 즉 $4a^3+27b^2\neq 0$인 경우에 집합 E'와 무한(infinite)원점 O와의 합집합 $E(F)=E'\cup O$는 아래 (정의5)와 같은 덧셈을 주면 가환군이 됨을 알 수 있다. 이를 **타원곡선군**(a group over Elliptic curve)이라 정의한다.

(정의 5). [덧셈의 정의] 타원곡선 군에서의 덧셈을 다음과 같이 정의한다. 단, 체 F의 표수(characteristic)는 3보다 크다고 가정한다.

① $\forall P\in E(F),\ P+O=O+P=P.$

② $\forall P=(x,y)\in E(F),$
$(x,y)+(x,-y)=O,$
즉, $-P=(x,-y).$

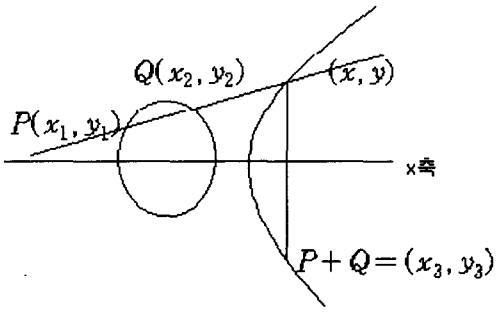
③ $\forall P=(x_1,y_1),\ Q=(x_2,y_2),$
$P+Q=(x_3,y_3).$

여기서 $\begin{cases} x_3=a^2-x_1-x_2, \\ y_3=-y_1+a(x_1-x_3). \end{cases}$

단, $a=\begin{cases} \dfrac{y_2-y_1}{x_2-x_1} & P\neq Q, \\ \dfrac{3x_1^2+a}{2y_1} & P=Q. \end{cases}$

덧셈 ③에 대한 기학학적인 그림을 그려보면 다음과 같이 P, Q점을 잇는 직선과 만난 점의 x축 대칭점 (x_3,y_3)을 $P+Q$로 정의한다.

<그림 1> 타원곡선군의 덧셈



(예 6). [타원 곡선군의 예] $F=Z_{11}$이고 $a=4$, $b=3$일 때의 타원곡선군은

$E(Z_{11})=\{O,(0,5),(0,6),(3,3),(3,8),(5,4),(5,7),(6,1),(6,10),(7,0),(9,3),(9,8),(10,3),(10,8)\}$

인 14개의 원소를 가진다. 또한 연산정의에 의해 계산하면

$(0,5)+(3,3)=(6,10)$이고

$(0,5)$의 위수는 7이다.

B. 타원곡선의 이산대수 문제

타원곡선군의 경우에서 이산대수의 문제(Elliptic Curve Discrete Logarithm Problem ; ECDLP)란 타원곡선군의 임의의 두 원소 P, Q에 대해서 $Q=xP$를 만족하는 정수 x를 구하는 문제를 말한다. 이 때의 x를 구하는 것은 매우 어렵다고 알려져 있다. 현재까지는 타원곡선의 이산대수문제는 인수분해나 일반적인 이산대수 문제보다 훨씬 어려운 문제로 인식되고 있다. 타원곡선 암호시스템의 안전도는 이러한 타원곡선 이산대수문제에 기반을 두고 있다.

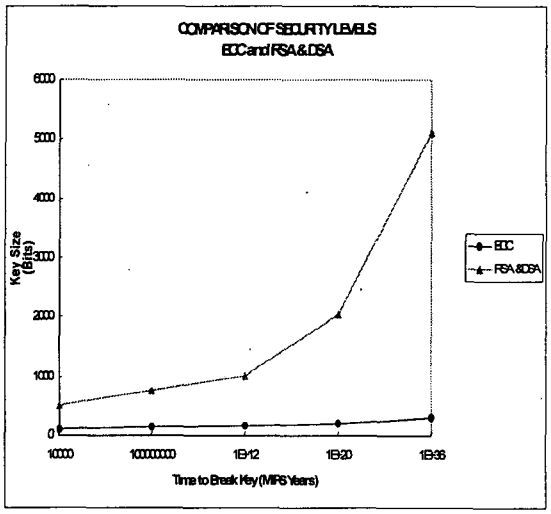
실수체 상에서의 계산은 일반적으로 느리며 또한 반올림에 따른 오차로 인하여 정확하지가 않기 때문에 암호 응용에는 적합하지가 않다. 따라서 암호시스템 구축을 위해서는 원소 수가 유한인 유한체(finite field) 위에서 정의된 타원곡선 군을 이용한다. 본 논문에서의 유한체는 3보다 큰 소수 p에 대한 유한체 Z_p 상의 알고리즘을 작성하였다.

C. 타원곡선 암호시스템의 배경

타원곡선(Elliptic Curves)은 약 150년 전부터 수학적으로 광범위한 연구가 있어 왔으며, 타원곡선암호시스템은 10여년 전 비트당 안전도가 타 공개키 시스템보다 효율적이라는 것이 알려졌고, 컴퓨터의 성능향상에 따라 최근 높은 속도의 구현이 가능하게 되었다.

<표 1>에서 암호에 쓰이는 비트수(세로축)와 컴퓨터가 계산할 수 있는 계산량(가로축)과의 관계를 알 수 있다. 예를 들면, RSA 1024비트 키와 ECC 160비트 키를 갖는 암호시스템은 같은 안전도를 갖는다는 것을 알 수 있다.

<표 1> 타원곡선(ECC)암호와 RSA&DSA암호의 안전도 비교



D. 타원곡선 암호시스템의 장점

타원곡선 암호시스템은 유한체의 곱셈군에

근거한 시스템으로써 다음의 장점을 가진다.

① 여러 가지 타원곡선을 활용하여 다양한 군(Group)을 제공할 수 있으므로, 다양한 암호시스템 설계가 용이하다.

② 특수한 몇 가지 타원곡선을 피하면, 이 군에서의 subexponential time algorithms이 존재하지 않는다. 즉, 안전한 암호시스템을 설계하는 것이 용이하다.

③ 타원곡선 암호시스템은 현재의 다른 공개키 암호시스템보다 작은 키길이(예 RSA시스템의 160/1024 정도의 길이)를 가지고 같은 안전도를 제공한다.

④ 타원곡선에서의 더하기 연산은 유한체에서의 연산을 포함하므로, 하드웨어와 소프트웨어로 구현하기가 용이하다.

⑤ 타원곡선군에서의 이산대수 문제는 특히, 같은 크기의 유한체 K에서의 이산대수 문제보다 훨씬 어렵다고 알려져 있다.

III. 타원곡선 공개키 암호 시스템의 구성

타원곡선을 이용한 공개키 암호시스템 즉, 유한체(finite fields)위에서 정의된 타원곡선군에서의 이산대수 문제에 기초한 타원곡선 암호시스템(ECC, Elliptic Curve Cryptosystem)은 1985년 N. Koblitz와 V. Miller에 의해 처음 제안된 이후 활발히 연구되고 있다. 더욱이, 타원곡선방법(ECM, Elliptic Curve Method)은 최근 RSA 암호시스템의 근간이 되는 인수분해 문제와 소수성 테스트를 위한 효율적 알고리즘을 제공하기도 하였다.

유한체 위에서의 타원곡선 암호는 Diffie-Hellman 키교환 교환 스킴, ElGamal, Schnorr와 NIST 서명스킴을 구현하는 데에 사용될 수도 있다. 이러한 시스템들은 짧은 키 길이를 가지고도 다른 공개키 스킴과 동등한 안전도를 제공할 수 있다. 짧은 키길이를 갖는다는 것은 대역폭과 메모리가 작아짐을 의미한다. 이것은 메모리와 처리능력이 제한된 스마트 카드 같은 응용에서 중요한 요소일 수도 있다.

타원곡선을 사용하면서 또 다른 이점은 비록 모든 사용자가 같은 기저체 K를 사용한다할 지라도 각 사용자가 다른 곡선 E를 선택할 수도 있다는 것이다. 즉, 모든 사용자는 체 연산을 수행하기 위해 같은 하드웨어를 사용할 수 있으며, 추가 보안을 위해 주기적으로 곡선 E를 바꿀 수 있다. 이것은 저자의 타원곡선암호의 연습프로그램을 통해서도 확인할 수 있다.

타원곡선 암호 중 가장 널리 이용되고 있는 타원곡선을 이용한 ElGamal암호의 알고리즘에 대해 살펴보고자 한다.

다음은 타원곡선 암호 중 ElGamal 공개키 암호의 암호화 · 복호화 과정이다.

ElGamal 공개키 암호

① **공개키 준비**: 유한체 F상의 타원곡선군 $E(F)$와 $E(F)$의 원소 중 큰 위수를 갖는 공개키 G를 정한다.

② **송신자**: $E(F)$의 원소 중 평문 M을 준비한다.

③ **수신자**: 개인비밀키 y로 yG를 계산하여 공개키로 yG를 공개한다.

④ **암호화**: 송신자는 임의의 정수 k를 선택하여 $C_1 = kG$, $C_2 = M + k(yG)$을 계산하여 암호문 (C_1, C_2)을 수신자에게 보낸다.

⑤ **복호화**: 수신자는 $C_2 - yC_1$을 계산하여 평문 M을 얻는다.

(예 7). [암호화와 복호화하기] (예6)에서와 같이 공개키로서 $F = Z_{11}$, $a = 4$, $b = 3$과 $G = (0,5)$를 공개한다. 그리고 수신자는 개인비밀키로 $y = 5$, 송신자는 수신자에게 보

낼 평문 $M=(3,3)$을 준비한다.

먼저 수신자가 개인비밀키 5로

$$yG=5(0,5)=(5,7)$$

을 계산하여 공개키로 $(5,7)$을 공개한다.

다음에 송신자는 임의의 정수 $k=3$을 선택하여 공개키 $G=(0,5)$와 수신자의 공개키 $(5,7)$를 이용하여 암호문

$$C_1 = kG = 3(0,5)=(10,8)$$과

$$\begin{aligned} C_2 &= M+k(yG) \\ &=(3,3)+[3*(5,7)] \\ &=(3,3)+(0,5) \\ &=(6,10) \end{aligned}$$

을 계산하여 (C_1, C_2)를 수신자에게 보낸다.

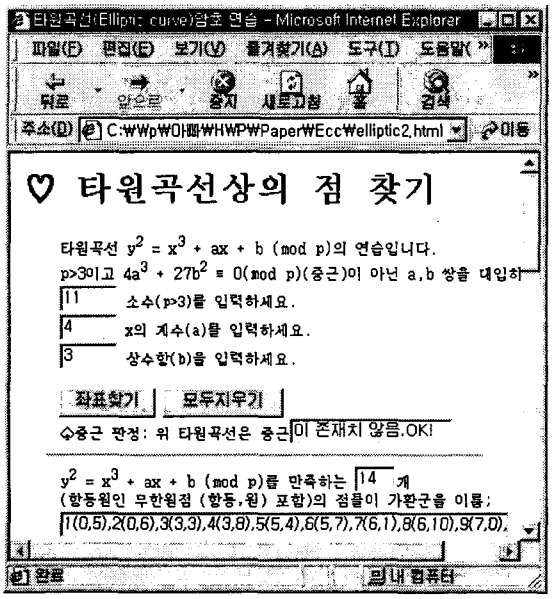
수신자는 다음 계산을 하여 복호화한다.

$$\begin{aligned} C_2-yC_1 &= M+k(yG)-y(kG) \\ &=(6,10)-5(10,8) \\ &=(6,10)-(0,5) \\ &=(6,10)+(0,6) \\ &=(3,3). \end{aligned}$$

IV. 타원곡선 공개키 암호 알고리즘과 프로그램

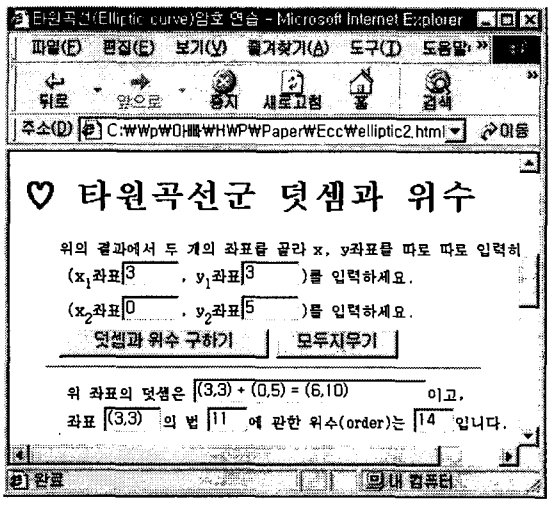
이제까지 논의한 타원곡선 공개키 암호 시스템에 대한 알고리즘을 종합하면 다음과 같다. 먼저 <그림 2>에서와 같이 공개키 준비를 위한 3보다 큰 소수 p와 a, b를 입력하여 [좌표찾기] 버튼을 누르면, 타원곡선군이 될 수 있는 중근이 존재하는지 알 수 있도록 하였고, 타원곡선상의 점의 개수와 모든 점의 좌표를 표시하도록 하였다. 중근이 존재하면 다른 수치를 입력하여 다시 시도하면 된다.

<그림 2> 타원곡선상 점 찾기



다음 단계로 <그림 3>에서는 위 <그림 2>에서 구한 좌표 중에서 2개의 원소를 입력하여 덧셈을 하고, 한 원소의 위수를 구할 수 있도록 하였다.

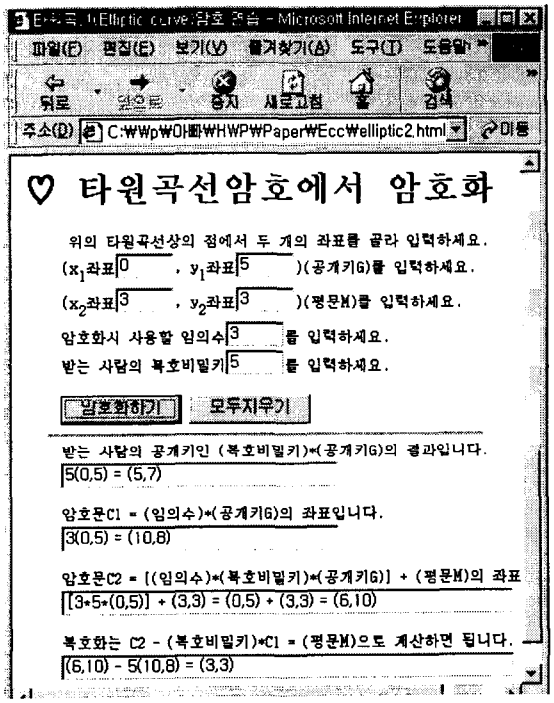
<그림 3> 타원곡선군의 덧셈과 원소의 위수



마지막으로 <그림 4>에서는 암호화와 복호화를 연습할 수 있는 단계로서 공개키 G

와 평문 M을 입력하고, 수신자의 개인비밀키 y와 암호화시 사용할 임의의 정수 k를 입력하여 [암호화하기]를 누르면 암호화 및 복호화하는 각 단계의 계산이 나타나도록 하였다. 이 프로그램에서는 복호화는 알고리즘을 주지 않고 <그림 3>에서 계산하여 확인할 수 있도록 하였다.

<그림 4> 타원곡선암호의 암호화 · 복호화



다음은 이들을 구하는 타원곡선 알고리즘을 소개하고자 한다.

제1단계 알고리즘에서는 유한체 Z_p와 타원곡선군의 원소를 구할 수 있는 알고리즘과 타원곡선이 중근을 갖는가를 판정하는 알고리즘을 주었다.

제2단계에서는 제1단계에서 구한 2개의 원소를 입력하여 덧셈과 원소의 위수를 구하는 알고리즘을 주었으며, 제3 · 4단계에서는 암호문 C_1과 C_2를 구하는 알고리즘을 구하고, 복호화하는 과정을 나타내는 알고리즘을 주었다.

제1단계 타원곡선상의 점 찾기와 중근판정 알고리즘

1. **input** *cleartext*, p, a, b
   // p는 소수, a, b는 임의의 정수
2. **let** $ii=0$; //타원곡선군의 원소수
3. **for** $i=0$ **to** $p-1$ //원소찾기
   for $j=0$ **to** $p-1$
   if $j^2 \equiv i^3+a\cdot i+b \pmod p$
   then $ii=ii+1$; **print** (i,j);
4. **if** $4a^3+27b^2 \equiv 0 \pmod p$//중근판정
   then print "중근이 존재! 다시!"
   else print "중근이 존재치 않음."

Subroutine1 유클리트 알고리즘에 의한 법 p 에 관한 역원구하기

sub1. **let** $p_1=p$; $sx_1=1$; $sy_1=0$;
$sx_2=0$; $sy_2=1$; $ii=0$;
sub2. **while** $p_2 \neq 0$ **do**
$qq=[p_1/p_2]$; // 몫
$r_1=p_1\%p_2$; // 나눈 나머지
//유클리트 알고리즘으로 역원 구하기
sub3. **if** $ii\%2\equiv 0$
then $sy_1=sy_1-(sy_2\cdot qq)$
else $sy_2=sy_2-(sy_1\cdot qq)$
$ii=ii+1$; $p_1=p_2$; // p_1이 공약수
$p_2=r_1$;
sub4. **if** $ii\%2\equiv 0$ **then** $yy=sy_1$
else $yy=sy_2$;
// yy 는 p_2의 역원(mod p)
sub5. **if** $yy<0$ **then** $yy=yy+p$;
// 양수로 만들기
sub6. **return**

제2단계 타원곡선군의 덧셈 알고리즘

1. **input** $xx_1; yy_1; xx_2; yy_2$ //좌표 입력
2. **let** $xx_3=0;\ yy_3=0;$ // 더한 좌표
   $xxx=0;$ // x 좌표
   $yy=0;$ // 역원
3. **if** $xx_1 \neq xx_2$ // x 좌표가 다른 경우

3-1. **then** $xxx=xx_2-xx_1;$
   if $xxx<0$ **then** $xxx=xxx+p;$
   let $p_2=xxx;$
   go subroutineI;
   $xx_3=(((yy_2-yy_1)\cdot yy)^2 - xx_1-xx_2)\%p;$
   $yy_3=(((yy_2-yy_1)\cdot yy)\cdot (xx_1-xx_3)-yy_1)\%p$

3-2. **else** // x 좌표가 같은 경우
   if $(yy_1+yy_2)\%p\equiv 0$
   // 역원관계 덧셈
   then $xx_3=$ 항등; $yy_3=$ 원
   else // 같은 좌표 덧셈
   $p_2=(2\cdot yy_1)\%p;$
   go subroutineI;
   $xx_3=(((3*xx_1^2+a)\cdot yy)^2 - xx_1-xx_2)\%p;$
   $yy_3=(((3\cdot xx_1^2)+a)\cdot yy)\cdot (xx_1-xx_3)-yy_1)\%p;$

4. **if** $xx_3<0$ **then** $xx_3=xx_3+p;$
5. **if** $yy_3<0$ **then** $yy_3=yy_3+p;$
6. **print**
   $(xx_1, yy_1)+(xx_2, yy_2)=(xx_3, yy_3);$
   // 더한 결과 출력

제2단계 타원곡선군의 위수 알고리즘

7. **let** $xxo=xx_1;\ yyo=yy_1;$ //x,y 좌표
   $order=2;$ //위수
8. **while** $xx_1\neq xxo \parallel (yy_1+yyo)\%p\neq 0$

8-1 **if** $xx_1\neq xxo$ //x 좌표가 다른 경우
   $xxx=xxo-xx_1;$
   if $xxx<0$ **then** $xxx=xxx+p;$
   $p_2=xxx;$
   go subroutineI;
   $xxo=(((yyo-yy_1)\cdot yy)^2 - xx_1-xxo)\%p;$
   $yyo=(((yyo-yy_1)\cdot yy)\cdot (xx_1-xxo)-yy_1)\%p;$
   if $xxo<0$ **then** $xxo=xxo+p;$
   if $yyo<0$ **then** $yyo=yyo+p;$
   $order=order+1;$

8-2 **else** // x 좌표가 같은 경우
   $p_2=(2\cdot yyo)\%p;$
   go subroutineI;
   $xxo=(((3\cdot xx_1^2+a)\cdot yy)^2 - xx_1-xxo)\%p;$
   $yyo=(((3\cdot xx_1^2+a)\cdot yy)\cdot (xx_1-xxo)-yy_1)\%p;$
   if $xxo<0$ **then** $xxo=xxo+p;$
   if $yyo<0$ **then** $yyo=yyo+p;$
   $order=order+1;$

9. **print** $order$
   // 위수출력

제3단계 타원곡선의 수신자 공개키 알고리즘①

1. **input** xx_1; yy_1; xx_2; yy_2; key_2; key_3
   //공개키 $G=(xx_1, yy_1)$,
   //평문 $M=(xx_2, yy_2)$
   //암호화하는 적당한 수 $k=key_2$,
   //복호비밀키 $y=key_3$ 입력

2. **let** $xx_3=xx_1$; $yy_3=yy_1$; //공개키
   $cx_1=xx_1$; $cy_1=yy_1$; //암호문C1
   $xxx=0$; // x 좌표구하기
   $yy=0$; // 역원구하기
4. **let** $xx_4=xx_1$; $yy_4=yy_1$;
   //공개키 좌표
5. **for** $j=0$ **to** $j=key_3-2$
   if $xx_4=$항등
   then $xx_4=xx_1$; $yy_4=yy_1$
   // 항등원계산
   else
5-1 **if** $xx_1\neq xx_4$ //x좌표가 다른 경우
   then $xxx=xx_4-xx_1$;
   if $xxx<0$
   then $xxx=xxx+p$;
   let $p_2=xxx$;
   go subroutineI ;

$$xx_4=(((yy_4-yy_1)\cdot yy)^2-xx_1-xx_4)\%p;$$

$$yy_4=(((yy_4-yy_1)\cdot yy)\cdot(xx_1-xx_4)-yy_1)\%p$$

제3단계 타원곡선의 수신자 공개키 알고리즘①

else // x 좌표가 같은 경우
if $(yy_1+yy_4)\%p=0$
// 역원관계 덧셈
then $xx_4=$항등; $yy_4=$원
else // 같은 원소 덧셈
$p_2=(2\cdot yy_1)\%p$;
go subroutineI;

$$xx_4=(((3\cdot xx_1^2+a)\cdot yy)^2-xx_1-xx_4)\%p;$$

$$yy_4=(((3\cdot xx_1^2+a)\cdot yy)\cdot(xx_1-xx_4)-yy_1)\%p;$$

5-2 **if** $xx_4<0$ **then** $xx_4=xx_4+p$;
5-3 **if** $yy_4<0$ **then** $yy_4=yy_4+p$;
6. **print** $key_3\cdot(xx_1, yy_1)=(xx_4, yy_4)$;
// 개인 공개키 yG 결과 출력

지금까지의 알고리즘에서는 타원곡선을 이용하여 타원곡선군의 원소(제1단계)를 찾아내고, 제2단계에서 원소들의 위수와 두 원소의 합을 계산할 수 있는 알고리즘을 주어 직접 계산한 결과를 알 수 있도록 하였다.

다음에 제3단계에서는 주어진 공개키 G를 이용하여 개인 비밀키 y로 개인 공개키 yG를 계산하는 알고리즘을 구하였다.

다음 제4단계는 송신자가 평문 M을 수신자에게 보내는 암호문을 만드는 알고리즘으로, 공개키 G와 개인 공개키 yG를 이용하여 송신자가 임의의 양의 정수 k를 선택하여, $C_1=kG$와 $C_2=M+k(yG)$를 계산하여, 평문 M에 대응하는 한 쌍의 암호문 (C_1, C_2)를 계산하는 알고리즘이다.

제4단계 타원곡선의 암호문 C_1 알고리즘②

// 암호문 C_1 (적당한키)·(공개키G) //

7.for $j=0$ to $j<key_2-1$

if $cx_1=$항등

then $cx_1=xx_1$; $cy_1=yy_1$

else

7-1 if $xx_1\neq cx_1$ //x좌표가 다른 경우

then $xxx=cx_1-xx_1$;

if $xxx<0$

then $xxx=xxx+p$;

let $p_2=xxx$;

go subroutineI;

$cx_1=(((cy_1-yy_1)\cdot yy)^2 - xx_1-cx_1)\%p$;

$cy_1=(((cy_1-yy_1)\cdot yy)\cdot (xx_1-cx_1)-yy_1)\%p$

else // x 좌표가 같은 경우

if $(yy_1+cy_1)\%p=0$ //역원덧셈

then $cx_1=$항등; $cy_1=$원

else // 같은 원소 덧셈

$p_2=(2\cdot yy_1)\%p$;

go subroutineI;

$cx_1=(((3\cdot xx_1^2+a)\cdot yy)^2 - xx_1-cx_1)\%p$

$cy_1=(((3\cdot xx_1^2+a)\cdot yy)\cdot (xx_1-cx_1)-yy_1)\%p$;

7-2 if $cx_1<0$ then $cx_1=cx_1+p$;

7-3 if $cy_1<0$ then $cy_1=cy_1+p$;

8. print $key_2\cdot(xx_1,yy_1)=(cx_1,cy_1)$;

// 암호문 $C_1=kG$ 출력

제4단계 타원곡선의 암호문 C_2 준비 알고리즘③

//암호문 C_2 구하기의 준비로서

//(임의수)·(비밀키)·(공개키G)계산

9. let $ccx_1=cx_1$; $ccy_1=cy_1$;

10. let $cx_1=xx_4$; $cy_1=yy_4$;

$clxl=cx_1$; $clyl=cy_1$;

11. for $j=0$ to $j<key_2-1$

if $clxl=$항등 && $cx_1=$항등

then $clxl=$항등; $clyl=$원;

break // 항등원이면 계산중지

else

11-1 if $clxl=$항등

then $clxl=cx_1$; $clyl=cy_1$

else

if $cx_1=$항등

then $cx_1=clxl$; $cy_1=clyl$

else

if $cx_1\neq clxl$ //x 좌표 다름

then $xxx=clxl-cx_1$;

if $xxx<0$ then

$xxx=xxx+p$;

let $p_2=xxx$; go subroutineI;

$clxl=(((clyl-cy_1)\cdot yy)^2 - cx_1-clxl)\%p$;

$clyl=(((clyl-cy_1)\cdot yy)\cdot (cx_1-clxl)-cy_1)\%p$

else // x 좌표가 같은 경우

if $(cy_1+clyl)\%p=0$

// 역원관계 덧셈

then $clxl=$항등; $clyl=$원

제4단계 타원곡선의 암호문 C_2 준비 알고리즘③

else $p_2=(2\cdot cy_1)\%p$;

// 같은 원소 덧셈

go subroutineI;

$clxl=(((3\cdot cx_1^2+a)\cdot yy)^2 - cx_1-clxl)\%p$;

$clyl=(((3\cdot cx_1^2+a)\cdot yy)\cdot (cx_1-clxl)-cy_1)\%p$;

if $clxl<0$

then $clxl=clxl+p$;

if $clyl<0$

then $clyl=clyl+p$;

12. **print** $key_2\cdot key_3\cdot(xx_1,yy_1)$

$=(clxl,clyl)$;

//준비 $k(yG)$ 결과 출력

제4단계 타원곡선의 암호문 C_2 알고리즘④

// 암호문 C_2 만들기//

13. **let** $cx_2=clxl$; $cy_2=clyl$;

14. **if** $cx_2\equiv$항등

then $cx_2=xx_2$; $cy_2=yy_2$

else

14-1 **if** $xx_2\neq cx_2$ //x 좌표가 다름

then $xxx=cx_2-xx_2$;

if $xxx<0$

then $xxx=xxx+p$;

let $p_2=xxx$;

go subroutineI;

제4단계 타원곡선의 암호문 C_2 알고리즘④

$cx_2=(((cy_2-yy_2)\cdot yy)^2 - xx_2-cx_2)\%p$;

$cy_2=(((cy_2-yy_2)\cdot yy)\cdot (xx_2-cx_2)-yy_2)\%p$

else // x 좌표가 같은 경우

if $(yy_2+cy_2)\%p=0$

// 역원관계 덧셈

then $cx_2=$항등; $cy_2=$원

else $p_2=(2\cdot yy_2)\%p$;

// 같은 원소 덧셈

go subroutineI;

$cx_2=(((3\cdot xx_2^2+a)\cdot yy)^2 - xx_2-cx_2)\%p$;

$cy_2=(((3\cdot xx_2^2+a)\cdot yy)\cdot (xx_2-cx_2)-yy_2)\%p$;

14-2 **if** $cx_2<0$

then $cx_2=cx_2+p$;

14-3 **if** $cy_2<0$

then $cy_2+=p$;

15. **print**

$[key_2\cdot key_3\cdot(xx_1,yy_1)]+(xx_2,yy_2)$

$=(clxl,clyl)+(xx_2,yy_2)$

$=(cx_2,cy_2)$;

// 암호문 C_2 출력

16. **print**

$(cx_2,cy_2)-key_3\cdot(ccx_1,ccy_1)$

$=(xx_2,yy_2)$

// 복호화 과정 출력

위 알고리즘에 따라서 Javascript로 작성한 타원곡선암호 연습 프로그램은 저자의 홈페이지(http://my.dreamwiz.com/math88)에 올려놓았다. 그 외에 몇 가지 고전암호와 RSA 암호에 관한 연습 프로그램도 올려놓았다.

V. 결 론

현재는 인터넷의 발달이 너무나 빨라, 요즈음은 컴퓨터를 모르면 거의 모든 일을 처리하지 못할 정도이다. 특히 암호의 중요성은 전쟁에서의 예를 들지 않더라도, 컴퓨터 통신, 은행 업무, 전자카드, 전자상거래 및 국가간의 정보전쟁 등 현대에는 암호가 쓰이지 않는 곳이 거의 없다고 해도 과언이 아니다.

현재 활용되고 있는 여러 암호이론 중에서 타원곡선암호는 소인수분해, 소수 판정법 및 공개키 암호와 관련된 다양한 알고리즘을 설계하는데 이용되고 있다.(박창섭) 하지만 일반인들이 접근하기에는 쉽지 않다. 그래서 약간의 정수론 및 대수학의 지식만 있다면 접근하여 계산해보고 암호의 이론을 이해할 수 있는 프로그램의 개발이 필요하다.

따라서 본 논문에서는 이들 이론을 활용하여 작성한 타원곡선암호 연습 프로그램을 제공하였으며, 이 프로그램을 통하여 얻을 수 있는 효과로서 다음과 같은 것을 기대할 수 있다.

첫째, 타원곡선 암호를 이해하고자 하는 사람들에게 유용하게 이용될 것이다. 즉 구체적인 수학적인 이론을 알지 못하더라도 타원곡선군의 구체적인 원소수를 구할 수 있고 원소의 위수를 구할 수 있을 뿐만 아니라 복잡하게 정의된 덧셈 연산의 결과를 알 수 있다. 따라서 암호학 강좌를 수강하는 학생들에게 도움을 줄 수 있다.

둘째, 구체적인 예의 다양한 연습을 통하여 타원곡선 암호의 구조와 유용성에 대하여 쉽게 체험할 수 있다.

셋째, 인터넷이 설치된 컴퓨터라면 언제 어디서나 접근하여 활용할 수 있도록 저자의 홈페이지(http://my.dreamwiz.com/math88)에 올려놓았다.

넷째, Javascript로 작성하였으므로 프로그램의 소스를 알 수 있어 이러한 종류의 프로그램을 작성하고자 하는 사람들에게 참고자료가 될 수 있다.

참 고 문 헌

강주성외(2000), 현대암호학, 경문사

박창섭(1999), 암호이론과 보안, 대영사

岡本龍明, 太田和夫(1995), 暗號, ゼロ 知識証明 數論, 共立出版

岡本龍明, 山本博資(1998), 現代暗號, 産業圖書

Beutelspacher, A.(1994), Cryptology, The Mathematical Association of America

koblitz, N.(1994), A course in Number Theory and Cryptography, Springer-Verlag New York

Menezes, A.(1993), Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers

A Practice Program of Elliptic Curve Cryptosystems with Javascript

Kim, Seung Dong[3] · Chung, Sang Cho[4]

ABSTRACT

This note introduces elliptic curve cryptosystems and related algorithms and gives an elliptic curve cryptosystems practice program made with Javascript. We can find the practice program at author's homepage "http://my.dreamwiz.com/math88". It is useful for students to study about elliptic curve cryptosystems.

3) Dept. of Mathematics Education, Kongju National University, Konju, 314-701, Korea
4) Dept. of Mathematics, Chungnam National University, Taejon, 305-764, Korea