通信情報保護學會誌
第3卷 第1號, 1993. 3

타원곡선위에서의 연산

최영주* · 황효선**

요 약

Finite field $GF(2^n)$5에서 정의된 elliptic curve가 있을 때, 그 curve 위의 어떤 point P를 k배하는 연산은 암호론에서 매우 자주 쓰여진다. 이때 optimal normal bases를 이용하여 $GF(2^n)$의 element를 표현하고, 또 elliptic curve를 선택할 때 anomalous curve가 되도록 한다면, 기존의 방법 보다 매우 빠르게 kP를 구할 수 있다.

1. Introduction

공개키 시스템 암호에서 많이 사용되는 방법 중 하나가 Diffie-Hellman이 제시한 Finite field 위에서 discrete log를 푸는 어려움을 이용하는 것이다. 그러나 최근의 많은 연구로 인하여 $GF(2^n)$인 경우 안전성을 위해서는 $n > 1280$정도가 되어야 함이 알려져 있다. 실제 computer로 구현함에 있어서 이와같이 큰 수를 다루는 것은 쉬운일이 아니다. 이를 극복하기 위하여 도입된 것이 elliptic curve이다.

일반적으로 어떤 field 위에서 Weierstrass equation

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

을 만족하는 점들과 infinity라 불리는 점 O의 집합을 E라고 하면, 연산을 잘 정의함으로써 E는 abelian group이 된다. 우리는 Finite field 위의 elliptic curve에서 이 연산을 이용하여 아래의 예와 같은 암호체계를 만들 수 있다. 이때 우리는 일반적으로 큰 k에 대해 kP를 계산해야 한다. (여기서 kP는 P를 k번 연산한 것이다.) 이 논문은 특별한 elliptic curve를 선택함으로써 kP를 빠르게 계산할 수 있는 방법을 제시하였다.

여기서 알아두어야 할 것은 당연하게도 $GF(p^n)$ 위의 elliptic curve에서의 연산이 $GF(p^n)$에서 곱하기를 하는 것보다 더 많은 시간이 걸린다. 또 지금까지는 elliptic curve가 supersingular인 경우에는 elliptic curve에서 discrete log 문제가 보통의 finite field에서의 discrete log 문제와 비슷한 어려움을 갖는다고 알려져 있다.

예 : Diffie-Hellman's key exchanges의 변형

어떤 finite field F 위에서 elliptic curve E가 정해지고 그 위의 point G가 선택되어 (F, E, G)가 모두 공개되어 있다. 이용자 A는 임의적으로 양의

* 포항공과대학 조교수
** 포항공과대학 대학원 석사과정

정수 a를 선택하고 aG를 공개한다. 역시 이용자 B도 임의적으로 양의 정수 b를 선택하고 bG를 공개한다. 상호간의 정보 교환을 위해 A는 공개된 bG와 자신만이 알고 있는 a를 이용하여 $a(bG)$를 만들고 B도 공개된 aG와 자신만이 알고 있는 b를 이용하여 $b(aG)$를 만든다. A와 B만이 알고 있는 Key로 쓰는 또 다른 암호 체계를 이용하면 A와 B는 안전하게 정보를 교환할 수 있다. 이 때 제3자인 C가 A와 B가 교환하는 정보를 알아내기 위해서는 G와 aG로부터 a, 또는 G와 bG로부터 b를 구할 수 있어야 하는데 이것이 elliptic curve 위에서 discrete log를 푸는 것이다.

2. Anomalous curve

q개의 원소를 가진 field $GF(q)$ 위에서 어떤 elliptic curve가 있을 때, 그 curve의 Frobenius map $\varphi : (x, y) \longmapsto (x^q, y^q)$의 trace가 1이라면, 이 curve를 anomalous curve라고 부른다. 어떤 curve가 anomalous하다는 것과 그 curve의 order $\#E$가 q이다는 것은 상등이다. 또 anomalous curve 상에서 φ는 characteristic equation $T^2-T+q=0$를 만족한다[1].

여기서 중요한 것은 Frobenius map과 characteristic equation의 성질이 $GF(q)$의 extention인 $GF(q^n)$에서도 유지된다는 것이다. 다시 말하면 어떤 point P를 q배 하고 싶다면 $qP=\varphi(P)-\varphi^2(P)$를 이용하여 계산할 수 있다. 이 때 우리가 normal bases를 이용하여 $GF(q^n)$의 element를 표현한다면 φ의 값은 계산이 쉬운 shift에 의해 구할 수 있다.

특별히 q가 2인 경우에는 $2^lP(l=1, 2, \cdots, 8)$를 아래와 같이 적은 수의 연산으로 계산할 수 있다.

$$
\begin{aligned}
2^1 &= T-T^2\\
2^2 &= 2(T-T^2)=2T-2T^2=(T-T^2)T-2T^2=-T^3-T^2\\
2^3 &= 4\cdot 2=(-T^3-T^2)(T-T^2)=-T^3+T^5\\
2^4 &= 4\cdot 4=(-T^3-T^2)^2=T^6+2T^5+T^4\\
&= T^6+(T-T^2)T^5+T^4=-T^7+2T^6+T^4\\
&= -T^7+(T-T^2)T^6+T^4=T^4-T^8\\
2^5 &= \cdots = T^5+T^6-T^7-T^8\\
2^6 &= \cdots = T^6-T^9+T^{11}-T^{12}\\
2^7 &= \cdots = -T^7+T^9+T^{11}-T^{13}\\
2^8 &= \cdots = T^8-T^{13}+T^{14}+T^{16}
\end{aligned}
$$

일반적으로 anomalous curve를 찾는 것은 간단하지 않다. 하지만 우리는 매우 작은 q(주로 2, 4 또는 8)만 다루기 때문에 $GF(q)$ 상의 모든 elliptic curve E에 대해 $\#E=q$이 되는지를 확인함으로써 anomalous curve를 찾을 수 있다.

3. Normal bases

어떤 n차의 irreducible polynomial f에 대해 $GF(2^n)$와 $GF(2)[x]/(f)$을 서로 isomorphic 하다는 것이 알려져 있다. 위의 사실을 이용하여 $GF(2^n)$의 element를 차수가 $n-1$ 이하인 polynomial로 표현할 수 있다. 달리 말하면 $GF(2^n)$을 polynomial bases $\{1, x, x^2, \cdots, x^{n-1}\}$을 가지는 vector space처럼 표현하는 것이다. 그러나 $\{\beta, \beta^2, \beta^{2^2}, \cdots, \beta^{2^{n-1}}\}$가 $GF(2^n)$의 bases가 되게하는 β가 항상 존재한다는 것이 알려져 있고, 이러한 bases를 normal bases라고 한다[4].

$GF(2^n)$을 normal bases로 표현할 때의 좋은 점은 square를 쉽게 할 수 있다는 것이다. 왜냐하면 어떤 $A=a_0\beta+a_1\beta^2+a_2\beta^{2^2}+\cdots+a_{n-1}\beta^{2^{n-1}} \in GF(2^n)$에 대해 $GF(2^n)$의 characteristic이 2라는 사실을 이용해서

$$
\begin{aligned}
A^2 &= (a_0\beta+a_1\beta^2+a_2\beta^{2^2}+\cdots+a_{n-1}\beta^{2^{n-1}})^2\\
&= a_0\beta^2+a_1\beta^{2^2}+a_2\beta^{2^3}+\cdots+a_{n-1}\beta^{2^n}\\
&= a_{n-1}\beta^{2^n}+a_0\beta^2+a_1\beta^{2^2}+a_2\beta^{2^3}+\cdots+a_{n-2}\beta^{2^{n-1}}\\
&= a_{n-1}\beta+a_0\beta^2+a_1\beta^{2^2}+a_2\beta^{2^3}+\cdots+a_{n-2}\beta^{2^{n-1}}
\end{aligned}
$$

이 된다. 그러므로 A^2은 A에서 한번의 shift로 구할 수 있다.

한편 일반적인 두 elements A, B의 곱의 경우도 별로 복잡하지 않다. 먼저

$$A = \sum_{i=0}^{n-1} a_i\beta^{2^i},\quad B \sum_{i=0}^{n-1} b_i\beta^{2^i}$$

$$C = AB = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_ib_j\beta^{2^i}\beta^{2^j}$$

라 두자. 여기서 우리는 다시 $\beta^{2^i}\beta^{2^j}$를 normal bases로 나타내어서

$$\beta^{2^i}\beta^{2^j}=\sum_{k=0}^{n-1}\lambda_{ij}^{(k)}\ \beta^{2^k}$$

라 둘 수 있는데 이러한 $\{\lambda_{ij}^{(k)}\ i, j, k=0, 1, \cdots, n-1\}$을 알고 있다면 C를 알 수 있다. 그런데

$$\sum_{k=0}^{n-1}\lambda_{ij}^{(k)}\ \beta^{2^k}=\beta^{2^i}\beta^{2^j}=(\beta^{2^{i-1}}\beta^{2^{j-1}})^2$$

$$=(\sum_{k=0}^{n-1}\lambda_{i-1,\ j-1}^{(k)}\beta^{2^k})^2=\sum_{k=0}^{n-1}\lambda_{i-1,\ j-1}^{(k)}\beta^{2^{k+1}}$$

의 양변을 비교해보면 $\lambda_{i,j}^{(k)}=\lambda_{i-1,\ j-1}^{(k-1)}$이 되고, 일반적으로 $\lambda_{ij}^{(k)}=\lambda_{i-k,\ j-k}^{(0)}$이 된다. 그러므로 우리는 $\{\lambda_{ij}^{(0)}\ ;\ i, j=0, \cdots, n-1\}$만 알아도 C를 구할 수 있다. 일반적으로 $\lambda_{ij}^{(0)}$를 구하기 위해서는 Gaussian Elimination을 한번정도 해야 하는데 그나마 한 번 계산해두면 반복적으로 사용할 수 있기 때문에 $\lambda_{i-k,\ j-k}^{(0)}$을 찾는 것은 큰 문제가 되지는 않는다.

여기서 C_n을 $\{\lambda_{ij}^{(0)}\ ;\ i, j=0, \cdots, n-1\}$에서 0이 아닌 원소의 갯수라 하면, 모든 normal bases는 $C_n\geq 2n-1$을 만족시킨다는 것이 알려져 있으며, 특히 $C_n=2n-1$을 만족하는 normal bases를 optimal normal bases라 한다[2]. 모든 n에 대해 optimal normal bases가 존재하는 것이 아니지만 더 작은 C_n을 가지는 normal bases를 찾기 위한 노력은 Vanstone 등에 의해 이루어졌고 어느정도의 결과도 나와 있다. 우리가 optimal normal bases를 이용했을 경우 더하기와 곱하기는 polynomial bases를 이용했을 때의 거의 비슷한 시간에 할 수 있고, 특히 square는 한번의 shift로 계산 가능하므로 거의 무시할 수 있는 시간에 할 수 있다.

또 우리가 자주 이용하는 것은 어떤 element A의 inverse를 찾는 것이다. 일반적으로 polynomial bases의 경우에는 A와 irreducible polynomial f와의 최대공약수 1을 찾으면서 몫에 대한 정보를 저장해둔 후, 거슬러 올라가면서 inverse를 찾는 extended Euclidean algorithm을 이용하는데, 이 때 약 $O(\log_2^3 2^n)=n^3$의 시간이 걸린다.

그러나 normal bases인 경우에는 square가 단지 한 번의 shift로 계산 가능하기 때문에 $A^{-1}=A^{2^n-2}$을 이용하여 A^{-1}을 계산한다. 먼저 Wang이 제시한 방법을 보자.[3] $2^n-2=2+2^2+\cdots+2^{n-1}$이므로 $A^{-1}=A^2\cdot A^{2^2}\cdots A^{2^{n-1}}$이다. 따라서 A^{-1}를 $n-1$번의 shift와 $n-2$번의 곱하기로 구할수 있다. 한 번의 연산에 약 $O(\log_2^2 2^n)=n^2$의 시간이 걸리므로 전체적으로는 $O(\log_2^2 2^n)=n^3$의 시간이 걸린다.

하지만 다음과 같은 방법을 쓴다면 시간을 좀 더 절약할 수 있다.(우선 $B=A^2$이라 두자)

먼저 $k=\log_2\ n$에 대해 $B^{2^{2^i}-1}$ $i=0, 1, \cdots, k$의 table을 만든다. 만드는 방법은 $B^{1(2)}$을 한 칸 shift 하여서 $B^{10(2)}$를 얻고, 그것을 본래의 $B^{1(2)}$와 연산하여 $B^{11(2)}=B^{2^{2^2}-1}$을 얻는다. 이 때 한 번의 shift와 한 번의 연산을 사용하였다. 다음으로 $B^{11(2)}$을 두 칸 shift 하여서 $B^{1100(2)}$을 얻고, 또 그것을 본래의 $B^{11(2)}$와 연산하여 $B^{111(2)}=B^{2^{2^3}-1}$을 얻는다. 역시 한 번씩의 shift와 연산을 사용하였다. 계속해서 $i=4, 5, \cdots, k$인 동안 $B^{2^{2^i}-1}$을 얻을 수 있고, 이 때 $k-1$번의 shift와 연산을 사용한다. 그리고 이 $B^{2^{2^i}-1}$를 적당히 shift하여 서로 연산하면 $B^{2^{n-1}-1}$을 얻을 수 있다. 이 때에도 최대한 k번 정도의 shift와 연산을 사용하면 된다. 전체적으로 $2k$ 정도의 shift와 연산을 사용하면 되므로 어떤 element의 inverse를 구하기 위하여 $O(n^2\ \log_2\ n)$ 정도의 시간이 걸린다.

4. Running time에 대해

polynomial bases를 이용하는 경우 kP를 구하기 위해 repeated squaring을 하여 $\frac{3}{2}\log_2 k$ 정도의 연산을 해야한다. 그러나 normal bases와 anomalous curve를 이용한다면 어떤 point Q에 대해 $16Q$를 구하는 것이 shift와 한번의 연산으로 가능하기 때문에 $\{iP\ ;\ i=0, 1, \cdots, 15\}$의 table을 만들어 두고 일종의 repeated 16-powering을 사용한다면 $\frac{2\log_2 k}{4}$ 정도의 연산으로 kP를 구할 수 있다.

결과적으로 이 새로운 방법은 연산만으로 비교했을 때 기존의 repeated squaring 보다 3배 정도 빨라진다고 생각된다. 하지만 $\lambda_{ij}^{(k)}$를 계산하는 과정과 초기에 $\{iP\ ;\ i=0, 1, \cdots, 15\}$를 미리 계산해 두는 시간이 포함될 경우 기존의 방법 보다 얼마나 빨라

질지를 계산하기는 어렵다. 하지만 n이 충분히 크지면 앞의 두 요인이 거의 무시될 수 있을 것이라고 추측할 수는 있다.

5. Example

간단한 예로 $GF(2^5)$에서 $y^2+xy=x^3+x^2+1$인 경우에 대해서만 살펴보자.

먼저 normal bases를 구해보자. [2]에서는 $f=x^5+x^2+x$ 일 때 $\beta=x^2+1$가 하나의 normal bases가 됨을 보였다. 그러나 일반적으로 normal bases를 구하기 위해 [2. theorem 3.2]을 이용하자.

irreduciable polynomial $f=x^{10}+x^3+1$에 대해 $GF(2)[x]/(f)$에서 $\beta=(x)^{93}$으로 잡으면, $\beta=x^9+x^7+x^6+x^3+1$이 되고 $\beta^{-1}=x^8+x^7+x^6+x^3+x+1$이 된다. 그래서 $\gamma=\beta+\beta^{-1}=x^9+x^8+x$이 된다. 이 γ를 이용하여 bases를 만들어 보면 $N=\{\gamma, \gamma^2, \gamma^{2^2}, \gamma^{2^3}, \gamma^{2^4}\}$이 되고 여기서

$$\begin{aligned}
\gamma &= x^9+x^8+x \\
\gamma^2 &= x^9+x^8+x^6+x^4+x^2+x \\
\gamma^{2^2} &= x^9+x^6+x^5+x \\
\gamma^{2^3} &= x^8+x^5+x^4+x^3+x+1 \\
\gamma^{2^4} &= x^9+x^8+x^3+x^2
\end{aligned}$$

이 된다. 위의 γ의 power들로 부터 Gaussian elimination을 써서 $\lambda_{ij}^{(k)}$을 구해보면

$\lambda_{ij}^{(k)}$		k				
i	j	0	1	2	3	4
0	0	0	1	0	0	0
0	1	1	0	0	1	0
0	2	0	0	0	1	1
0	3	0	1	1	0	0
0	4	0	0	1	0	1

와 같이 된다. 이 결과는 위의 $n=5$, $f=x^5+x^2+1$, $\beta=x^2+1$인 경우와 일치한다. 여기서 우리는 $GF(2^5)$에서 $GF(2^{10})$의 subfield로 가는 isomorphism을 찾는 문제에 당면하게 된다. γ를 $GF(2)[x]/(x^5+x^2+1)$의 element로 표현할 수 있다면, 일반적으로 말해서 $GF(2^{2n})$에서 찾아진 γ를 $GF(2^n)$의 element로 쉽게 바꾸어 나타낼 수 있다면 우리는 위의 Gaussian elimination 등에서 상당한 이득을 볼 수 있고, 또 normal bases form과 polynomial bases form 사이에서 대응되는 point를 쉽게 찾을 수 있다. 그러나 일반적으로는 위에서 만든 것처럼 그것의 polynomial bases form과는 전혀 관계를 모르는 상태에서 normal bases를 이용해야 한다. 앞의 예와 같은 암호 체계를 만들때에는 이러한 normal bases를 이용하더라도 전혀 불편함이 없다.

다음으로 위의 elliptic ucrve $y^2+xy=x^3+x^2+1$ 상에서의 연산을 알아보자.

E 위의 두 point $P=(x_1, y_1)$, $Q=(x_2, y_2)$에 대해

case 1 if P is infinity O then $P+Q=Q$

case 2 if $x_1=x_2$ and $x_1=y_1+y_2$ then $P+Q=O$(P, Q가 서로 inverse임)

case 3 if $x_1=x_2$ and $y_1=y_2$ then put $\alpha=(x_1^2+y_1)(x_1^{-1})$ and $P+Q=(y_3, y_3)$ with $x_3=1+\alpha+\alpha^2$, $y_3=\alpha x_3+\alpha x_1+y_1+x_3$

case 4 if $x_1 \neq x_2$ then put $\alpha=(y_1-y_2)(x_1-x_2)^{-1}$ and $P+Q=(y_3, y_3)$ with $x_3=1+\alpha+\alpha^2+x_1+x_2$, $y_3=\alpha x_3+\alpha x_1+y_1+x_3$

이 되고, $\#E$=22임을 계산할 수 있다. 이제 우리는 order가 비교적 큰 하나의 point만 찾으면 된다. 현재 가능한 방법은 임의의 point를 잡아서 주어진 curve를 만족시키는지를 확인하면서 운이 좋아서 그것을 만족시키는 point가 우연히 찾아지기를 바라는 것 뿐인데, $GF(2^n)$의 경우 가능한 point의 수는 $(2^n)^2$인데 $\#E$는 약 2^n이므로 $\frac{1}{2^n}$ 의 확률로 찾을 수 있다. 특히 위의 예에서는 $y^2+xy=x^3+x^2+1$을 만족시키는 point가 22개 밖에 안되므로 $\frac{22}{(2^5)^2}$ 의 확률밖에 안된다. 하지만 실제로 찾아본 결과 (11000, 10001)과 (11000, 01001)이 위의 식을 만족시킨다.

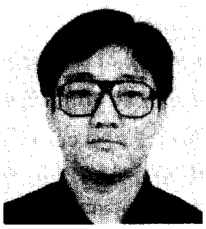
참 고 문 헌

1. Neal Koblitz, *CM-Curves with Good cryptographic Properties*, in Crypto'91.
2. R.C. Mullin, I.M. Onyszchuk and S.A. Vanstone, *Optimal Normal Bases in GF(2ⁿ)* Discrete Applied Mathematics 22 (1988/1989) pp. 149/161.
3. 조인호, 임종인, 서광석, 김창환, 갈로아체에서의 고속연산법 개발 및 응용, Comm. Korea Math. Soc. 7(192), No. 2, pp.347-364.
4. R. Lidl and H. Neiderreiter, *Finite Fields* (Addison ; Wesley, Reading, MA, 1956)

□ 著者紹介

최 영 주(정 회 원)

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



황 효 선

1992년 2월 포항공과대학 이학사
1992년 3월~현재 포항공과대학 대학원 석사과정