Digit-Serial 유한체 연산기와 Elliptic Curve algorithm에 기반한 암호프로세서 설계

남기훈, 이광엽
서경대학교 컴퓨터공학과
전화 : 02-940-7240

Design of a Cryptography processor based on Elliptic Curve Algorithm and Digit-serial Finite Field Circuits

Ki-Hun Nam, Kwang-Youb Lee,
Dept. of Computer engineering, seokyeong University
E-mail : kylee@skuniv.ac.kr

Abstract

본 논문에서는 타원곡선 알고리즘에 기반한 공개키 암호시스템 구현을 다룬다. 공개키의 길이는 193비트를 갖고 기약다항식은 $p(x)=x^{193}+x^{15}+1$ 을 사용하였다. 타원곡선은 polynomial basis 로 표현하였으며 SEC 2 파라메터를 기준으로 하였다. 암호시스템은 polynomial basis 유한체 연산기로 구성되며 특히, digit-serial 구조로 스마트카드와 같이 제한된 면적에서 구현이 가능하도록 하였다. 시스템의 회로는 VHDL, SYNOPSYS 시뮬레이션 및 회로합성을 이용하여 XILINX FPGA로 회로를 구현하였다. 본 시스템 은 Diffie-Hellman 키 교환에 적용하여 동작을 검증하였다.

I. 서론

공개키 암호시스템은 큰 유한군위에서의 지수 연산에 기반을 두고 있다. 이러한 시스템의 안전도는 유한군에서 이산 대수 문제를 계산하기 어렵다는 사실에 의존한다. 타원곡선을 사용한 암호시스템의 안전도는 타원 곡선 점위에서 정의된 대수 시스템의 이산 대수 문제가 풀기 어렵다는 사실에 의존한다. [1]

1976년 W. Diffie와 M. E. Hellman이 위의 문제를 해결한 "New Directions in Cryptography"에서 공개키 암호의 개념을 처음 소개하였다. 이후 1978년 소인수 분해의 어려움에 기반을 둔 RSA가 소개되어 지금까지 넓게 사용되고 있다. 그러나 RSA는 비도를 높이기 위해 1024 비트 이상으로 확장되는 추세로, 스마트카드와 같이 제한된 면적에 탑재되는데 어려움이 있다. 1987년 Koblitz와 Miller는 공개키 암호화 타원곡선(ECC) 알고리즘을 적용하였다. ECC는 적은 비트로 높은 비도를 보이기 때문에 최근 스마트 카드의 암호화 프로세서구현에 활용되는 추세에 있다.[2][3][4]

본 논문에서는 스마트카드에서 공개키 암호화 프로세서 모듈로 사용할 수 있는 암호 시스템을 타원곡선 알고리즘에 기반하여 구현하였다.

유한체위에서 정의된 타원 곡선은 원소의 개수가 2^m 개인 유한체 $GF(2^m)$상에서 다항식으로 표현되며 비초특이 타원 방정식을 갖는다. 타원 곡선을 정의하기 위한 방정식 파라메터는 Certicom Research사의 SEC 2를 사용하였다. 방정식 파라메터는 곡선의 계수 a, b와 베이스 좌표, 베이스 좌표의 차수, seed 등으로 구성하였다.

타원곡선 암호 시스템을 회로로 구현하는데는 유한체 승산기, 역원기, 가산기를 사용하였으며 스마트카드와 같이 제한된 면적에서 활용할 수 있도록 유한체 연

본 논문은 시스템 IC2010 사업의 지원으로 작성되었습니다.

산기는 digit-serial 구조를 갖도록 하였다.

시스템 구현 방법은 우선, Visual C 언어를 이용하여 상위단계 프로그램을 검증을 한 후 VHDL 모델링 및 시뮬레이션을 진행하였다. SYNOPSYS에서 시뮬레이션과 회로합성 검증을 마친 후 XILINX FPGA로 회로를 제작하였다.

ECC 암호시스템을 검증하기 위해서는 Diffie-Hellman의 키 교환 알고리즘을 이용하였다. 키 교환에는 스칼라 곱셈이 기본 연산이 되며 본 논문에서 제작한 FPGA는 스칼라 곱셈을 수행한다. 스칼라 곱셈기에 비밀키 k 와 타원곡선상의 임의의 좌표를 제공하여 공개키를 생성하고 이 를 교환하도록 하였다.

II. 타원곡선 알고리즘

2.1 타원곡선의 정의

유한체위에서 정의된 타원곡선군에서의 이산 대수 문제에 기초한 이산 대수 암호시스템의 한 종류인 타원곡선 암호시스템(ECC : Elliptic Curve Crypto systems)은 1985년 N.Koblitz[5]와 V.Miller에 의해 처음으로 제안된 이후 활발히 연구되고 있다.

유한체 K위에서 정의된 타원곡선 E위의 점들은 가환군의 형태를 이룬다. 이 가환 군의 더하기 연산은 기초체 K에서의 산술 연산 몇 개를 포함하며, 하드웨어로 구현하기가 쉽다. 또한 이 가환군에서 이산 대수를 사용하는 암호시스템은 유한체의 곱셈군에 기초한 시스템에 비해 장점을 가지고 있다.

첫째, 가환군에서의 이산 대수 문제는 같은 크기인 K 유한체에서의 이산대수 문제보다 더 어렵다. 즉, ECC는 작은 키 길이를 가지고도 현존하는 공개키 시스템의 안전도를 보장받을 수 있다.

둘째, 타원곡선의 기초체 K를 같이 사용하여도 사용자가 다른 곡선 E를 선택할 수 있다. 즉, 주어진 군에서 다양한 타원곡선을 사용할 수 있다. 결과적으로 모든 사용자들은 같은 하드웨어로 체 연산을 실행하고, 요구되는 안전도를 위해 주기적으로 곡선 E를 변화시킬 수 있다.

한편, 1990년 Menezes, Okamoto와 Vanstone의 연구에서 소위 초특이 타원 곡선이라 불리우는 타원 곡선의 이산 대수 문제가 유한체에서의 이산 대수 문제로 바뀔 수 있음을 보였다. 그러므로 만약 완전 지수 복잡도로 이 암호시스템이 깨지기를 원한다면 초특이 타원곡선을 피해야 한다.[5]

따라서, 본 논문에서는 비초특이 타원곡선으로 유한체 $GF(2^m)$상에서 정의되는 다음과 같은 타원곡선 방정식을 선택하였다.

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

2.2 F_{2^m} 상에서 타원곡선의 파라메터

타원곡선을 정의하는 파라메터는 Certicom Research 사의 SEC 2 에 따른다. F_{2^m} 상에서 정의되는 파라메터는 다음과 같이 표현된다.

$$T=(m, f(x), a, b, G, n, h)$$

m 은 기약다항식 $f(x)$의 차수를 나타내며 polynomial basis로 표현된다. 두 원소 $a, b \in F_{2^m}$ 는 타원곡선의 방정식을 정의하는 계수가 된다. $G=(x_G, y_G)$ 는 base point이며 n은 G의 차수 이다. h는 타원곡선상의 좌표의 갯수를 n으로 나눈 cofactor이다.

SEC 2에서 제안하는 m은 다음과 같다.

$$m \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}$$

본 논문에서는 m=193을 선택하였는데 이에 대한 근거는 Public Key Cryptography 2000에 따르면 192비트의 타원곡선 암호기반의 암호키는 2020년까지 안전한 것으로 증명하고 있으며 이에 필요한 암호공격 계산량은 6.54 x 1011 Years/450MHz P5 PC 으로 보고 있기 때문이다.

기약 다항식은 $f(x)=x^{193}+x^{15}+1$ 로 정하였다. 기약 다항식은 대체적으로 3항(trinomial) 다항식과 5항(pentanomial) 다항식이 사용되는데 3항 다항식은 하드웨어 구현에 보다 적합한 조건을 제시하기 때문에 본 논문에서도 3항 다항식을 선택하였다. 3항 다항식은 modular reduction 회로를 단순화 한다.

III. 유한체 연산기를 이용한 회로 설계

3.1 LFSR구조의 유한체 승산기

N차의 다항식 A, B와 m차의 원시다항식 α가 주어질 때 타원곡선에서 구현되는 유한체 승산의 표현식은 다음과 같다.

$$Z=A\cdot B=\sum_{i=0}^{m-1} b_i(A\alpha_i)$$
$$=[\cdots[[b_0(A)+b_1(A\alpha)]+b_2(A\alpha^2)]+\cdots]$$
$$+b_{m-1}(A\alpha^{m-1}) \quad (1)$$

원시다항식을 $\alpha^m=p_0+p_1\alpha+\cdots+p_{m-1}\alpha^{m-1}$ 라 정의하고 식(1)에 대입하여 정리하면 다음과 같다.

$$A\alpha=a_0\alpha+a_1\alpha^2+\cdots+a_{m-1}\alpha^m$$
$$=a_0\alpha+a_1\alpha^2+\cdots+a_{m-1}(p_0\alpha+p_1\alpha_2+\cdots$$
$$+p_{m-1}\alpha^{m-1})$$
$$=a_{m-1}p_0+(a_0+a_{m-1}p_1)\alpha+(a_1+$$
$$a_{m-1}p_2)\alpha^2+\cdots+(a_{m-2}+a_{m-1}p_{m-1})\alpha^{m-1} \quad (2)$$

식 (2)를 레지스터와 논리게이트를 이용하여 회로로 구현하면 그림 1과 같은 bit-serial 승산기를 얻을 수 있다

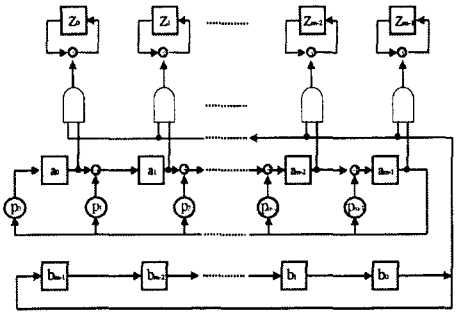

그림 1 순차회로를 이용한 승산기 I

3.2 Almost Inverse algorithm으로 구현된 유한체 역원기

$GF(2^m)$상의 원소의 역원은 원소 사이의 제산을 승산회로로 실현하기 위하여 정의하고 있다. 종래의 역원은 일반적으로 table look-up 방법이나 Euclid 알고리즘을 사용하여 왔으나 하드웨어 구현에 보다 장점이 있는 almost inverse algorithm을 이용하여 ECC 시스템을 구현하였다.

Almost inverse algorithm 과정은 그림 2에서 흐름도로 나타내었다. 흐름도에서 Z, A, B, D는 193비트 레지스터로 레지스터 A는 역원을 구하고자 하는 입력 다항식이 저장된다. 레지스터 A는 기약다항식을 저장하고 레지스터 D는 almost inverse algorithm을 거친 다항식이 저장되며 이 다항식은 차수 k에 의하여 축약된다. 차수 축약(modular reduction) 단계는 almost inverse algorithm 단계에서 구해진 B, k를 가지고 B를 x^k 로 나누는 연산을 수행한다. 이 과정이 끝나면 B는 A의 올바른 역원 값이다.

Z 레지스터는 상위비트에서 하위비트로 쉬프트하는 right shift 기능을 갖추고 B 레지스터는 하위비트에서 상위비트로 쉬프트하는 left shift 기능을 갖추고 있다. Z 레지스터의 최하위비트를 check하여 '0' 인경우 쉬프트를 계속 진행하며 '1'인경우에는 레지스터 Z에 저장되어있는 다항식의 차수와 G 레지스터의 차수를 비교하여 차수 G가 큰 경우에는 레지스터 A와 레지스터 Z의 내용을 교환한다. 또한, 레지스터 B와 레지스터 D의 내용을 교환한다.

위와같은 동작은 Z 레지스터의 차수가 '0' 이 될때까지 진행된다. '0' 이 되면 D 레지스터의 다항식과 차수 K가 dividing out 회로로 전달되어 modular reduction을 수행한다.

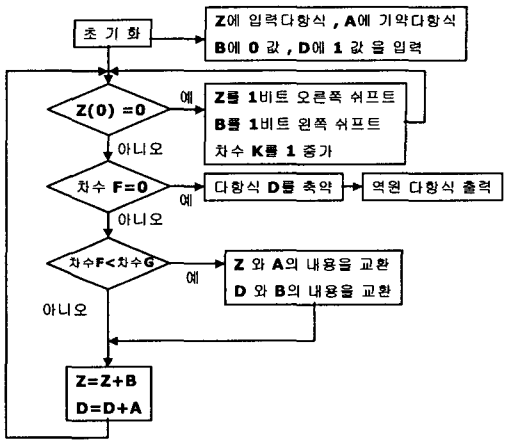

그림 2. Almost inverse algorithm의 흐름도

3.3 ECC 회로 설계

앞에서 설명된 Serial 구조의 승산기와 역원기를 바탕으로 193비트 키길이의 ECC 회로를 구현하였다. 승산기와 역원기를 결합한 ECC전체 회로는 그림 3에 나타내었다.

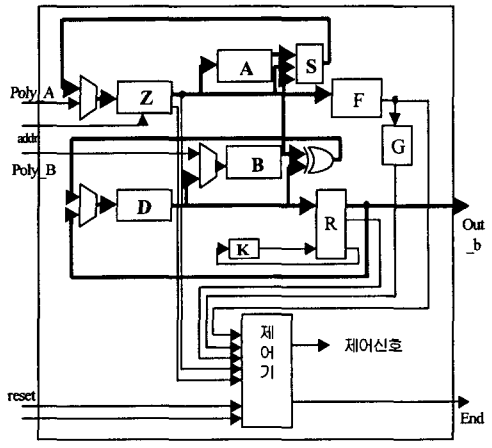

그림 3 ECC 전체 회로도

ECC 전체회로에서 A,B,Z 레지스터와 다중화기 S회로가 LFSR구조의 승산기를 나타낸다. F,G,K는 다항식의 차수를 계산하고 이를 저장 하는 회로이며 R은 차수 K에 따라 다항식을 축약(modular reduction)하는 회로이다. 승산연산에서는 out_b가 곱의 결과가 되며 입력으로는 poly_A와 poly_B를 이용한다. 역원기로 동

작하면 입력다항식이 poly_A로 입력되고 출력 out_b는 입력다항식의 역원을 나타낸다.

Ⅳ. 회로검증 및 FPGA구현

타원곡선의 이산대수 문제는 스칼라 곱셈으로 나타내진다. 즉, Q = k ·P 에서 k 는 임의의 정수값으로 비밀키에 해당되고 P 는 타원곡선상의 임의의 좌표이다. 곱셈 결과는 공개키가 된다. 본 연구에서는 타원곡선상에서 스칼라 곱셈을 수행하는 회로를 설계하였으며 스칼라 곱셈은 *double and add algorithm*에 의해 타원곡선상의 좌표의 덧셈문제로 분해할 수 있다. 이러한 문제를 정리하면 다음과 같다.

타원곡선 $y^2+xy=x^3+ax^2+b$ (1)

임의의좌표 $P(x1,y1), Q(x2,y2)$ (2)

$R=P+Q, R(x3,y3)$ (if $P\neq Q$) (3)

$$x3=\frac{(y1-y2)}{(x1+x2)}^2+\frac{(y1-y2)}{(x1+x2)}+x1+x2+a \quad (4)$$

$$y3=\frac{(y1-y2)}{(x1+x2)}(x1+x3)+x3+y1 \quad (5)$$

$R=2P, R(x3,y3)$ (if $P=Q$) (6)

$$x3=\frac{(y1-y2)}{(x1+x2)}^2+\frac{(y1-y2)}{(x1+x2)}+a \quad (7)$$

$$y3=\frac{(y1-y2)}{(x1+x2)}x3+x3+y1 \quad (8)$$

다른 두 좌표를 덧셈 하기 위해서 식(4),(5)를 계산한다. 식(4),(5)를 계산하는데는 1번의 역원(inversion)과 2번의 승산(multiplication), 그리고 1번의 제곱연산과 9번의 가산이 필요하다. 같은 두 좌표를 덧셈 하기 위해서 식(7),(8)를 계산한다. 식(7),(8)를 계산하는데는 1번의 역원과 2번의 승산, 그리고 1번의 제곱연산과 7번의 가산이 필요하다.

위 식에서 좌표 덧셈을 위해서는 유한체 역원, 승산, 제곱, 가산의 연산기가 필요한 것을 알 수 있다. 따라서 암호화기는 유한체 역원,승산,제곱,가산 연산기로 구성되며 이 들 연산기를 이용하여 다음 그림 4와 같이 타원곡선 암호 시스템을 구현하였다. 그림 4의 시스템은 XILINX FPGA XCV300을 이용하여 제작되었으며 그 결과는 그림 5에 수록하였다.

Ⅴ. 결 론

본 논문에서는 스마트카드내에 탑재되는 암호화프로세서 가운데 타원곡선(ECC)알고리즘 모듈을 구현하였다. 제안된 회로는 유한체 승산기와 역원기를 이용하였으며 승산기와 역원기 회로를 겸용하기에 적합하도록 LFSR구조의 승산기와 Almost inverse algorithm의 역원기를 사용하였다. 원시다항식이 $p(x)=x^{193}+x^{15}+1$인 다항식기저(polynomial basis) 유한체상에서 타원곡선 알고리즘을 사용한 암호화 프로세서에 적용하였다. 설계된 회로를 검증하기 위하여 SYNOPSYS에서 VHDL시뮬레이션과 회로합성을 하였고 XILINX FPGA로 제작하였다. 제작된 FPGA에서 스칼라 곱셈이 수행되는 것이 확인되었다.

***본논문에서 사용된 SYNOPSYS툴은 IDEC의 지원을 받았습니다.**

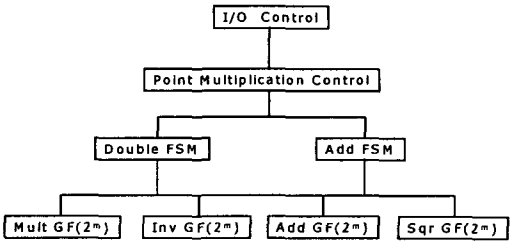

그림 4 타원곡선 암호화기의 전체 구성도

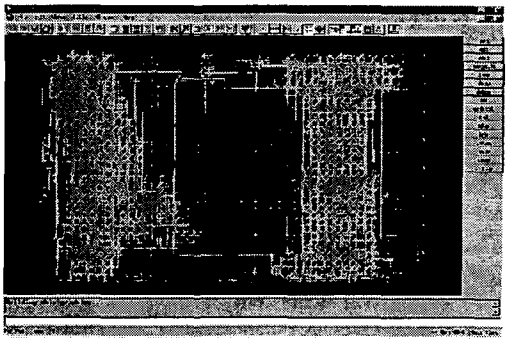

그림 5 암호화기의 FPGA구현 결과

참고문헌

[1] 이인수, "타원곡선 암호시스템에 관한 연구", 연세대학교 대학원 석사 논문, pp3-28, 12월, 1996

[2] J. Guajardo and C. Paar, " Efficient Algorithms for Elliptic Curve Cryptosystems", Advances in Cryptology - CRYPTO 97, B.S Kaliski, ed., pp. 342-356, 1997

[3] B, Sunar and C.K. Koc, "Mastrovito Multiplier for All Trinomials", IEEE Tran. Computers, Vol 48, NO5, pp. 522-527, May,1999

[4] David Naccache David M'Raïhi, "Cryptographic Smart Cards", IEEE MICRO, Vol 16, No 3, pp. 14-23, June, 1996

[5] N. Koblitz,"Elliptic Curve Cryptosystems", Mathematics of Computation, vol 48, pp203-209,1987

[6] A.Menezes, T. Okamoto and S.Vanstone," Reducing elliptic curve logarithms to logarithms in a finite field", IEEE Tran. on Information Theory, Vol.39, pp1636-1646, 1993.