타원곡선 암호시스템을 위한 기저체 연산기의 FPGA 구현

조성제* , 권용진
한국항공대학교 항공통신정보공학과

FPGA Implementation of Underlying Field Arithmetic Processor for Elliptic Curve Cryptosystems

Seong-Je Jo* , Yong-Jin Kwon
Dept. of Telecom. and Inform. Eng., Hankuk Aviation University
E-mail : sjjo@mail.hangkong.ac.kr

Abstract

In recent years, security is essential factor of our safe network community. Therefore, data encryption/ decryption technology is improving more and more. Elliptic Curve Cryptosystem proposed by N. Koblitz and V. Miller independently in 1985, require fewer bits for the same security, there is a net reduction in cost, size, and time. In this paper, we design high speed underlying field arithmetic processor for elliptic curve cryptosystem. The targeting device is VIRTEX V1000FG680 and verified by Xilinx simulator.

1. Introduction

최근 들어 통신망상의 보안이, 원활한 정보 통신 세계의 구축에 중요한 요소가 됨에 따라 정보의 암호 기술 또한 점차 발달하게 되었다. 초기의 암호 기술은 간단한 대치 및 전치방식을 사용했으나, 근래 컴퓨터의 계산능력이 비약적으로 증대되고 암호를 해독할 수 있는 새로운 알고리즘 및 수학적 이론들의 등장으로 인해 원하는 수준의 보안을 유지하기 위해 제안되는 암호 방식은 점점 더 복잡해지고 있다. 그러나 암호 방식의 복잡도가 증가함에 따라, 이를 응용하고 사용하기 위해 암호 알고리즘을 실제로 구현하는 것이 또 다른 문제로 대두되었다. 즉, 계산량이 커지고 암·복호화 시간이 오래 걸리며, 시스템을 구현하기 위해서는 사양이 높은 하드웨어가 필요하게 되었다.

1985년 N. Koblitz와 V. Miller가 각각 독립적으로 제안한 타원곡선 암호시스템(ECC : Elliptic Curve Cryptosystem)은 보다 짧은 비트 길이의 키만으로도 다른 공개키 시스템과 동일한 수준의 안전도를 유지할 수 있는 장점으로 인해 IC 카드와 같은 메모리와 처리능력이 제한된 하드웨어에도 이식가능 하다[1]. 또한 동일한 유한체 연산을 사용하면서도 다른 타원곡선을 선택할 수 있어서 추가적인 보안이 가능하기 때문에 고수준의 안전도를 유지하기 위한 차세대 암호 알고리즘으로 각광 받고 있다[2].

타원곡선 암호는 이미 차세대 암호 기법으로 인정받고 있으며, HP, Motorola, NTT, Fujitsu 등 세계 주요 IT 업체들이 타원곡선 암호기술을 기반으로 한 새 표준 암호 기술을 개발 중이다[3]. 또한 외국에서는 타원곡선 암호와 관련된 연구 및 소프트웨어, 하드웨어 구현이 다수 이루어지고 있다. 그러나 국내에서는 아직 타원곡선 암호에 대한 연구가 미비한 실정이며, 개발 역시 소프트웨어를 사용한 구현 사례만이 극소수 있을뿐, 하드웨어 구현 사례는 없다. 따라서 본 연구에서는 타원곡선이 정의된 기저체 상의 연산을 고속으로 수행할 수 있는 전용 프로세서의 개발을 통해 IC Card, 이동 통신 단말기 등, 낮은 사양의 하드웨어에서도 강력한 보안이 필요한 여러 분야로의 응용을 가능하게 하고, 나아가 국내 암호 프로세서의 IP 구축을 목적으로 하고 있다.

먼저 타원곡선 암호의 수학적인 배경에 대해서 설명한다. 다음으로 구현의 핵심이 되는 기저체 연산을 분석하고, 체 원소의 표현법에 따른 연산 방법의 차이에 대해서 고찰한다. 이러한 분석을 통해 연산을 위한 효율적인 하드웨어 구조를 제안한 뒤, 최종적으로 도출된 이론을 바탕으로 프로세서를 설계하는 과정 및 구현 결과에 대해서 기술한다.

2. Mathematical Background

타원곡선을 암호시스템에 적용하기 위해서는 우선 적절한 타원곡선을 선택하고 그에 따른 타원곡선 군을 구해야 한다. 기저체 상의 타원곡선을 나타내기 위해서 여러 가지 좌표계를 사용할 수 있으나, Affine 좌표계나 Projective 좌표계를 주로 사용한다. Affine 좌표계를 사용하면 점 연산 시 곱셈과 덧셈의 횟수가 줄어들지만 역원을 구하는 과정이 필요하다. 반면 Projective 좌표계를 사용하면 역원 연산은 없으나 곱셈과 덧셈의 횟수가 증가한다. 타원곡선을 표현함에 있어서 좌표계의 선택은 역원을 얼마나 효율적으로 구할 수 있는가에 달려 있다. 역원을 구하는 과정 역시 여러 단계의 곱셈 및 덧셈으로 이루어져 있기 때문에 전체적인 연산량을 비교 검토한

후에 좌표계를 선택하는 것이 바람직하다. 유한체 상에서 역원을 구하는 효율적인 알고리즘이 최근 다수 발표되었고 본 연구에서도 역원을 구하는 효율적인 하드웨어 알고리즘을 제시하고 있으므로, Affine 좌표계에 타원곡선을 나타내는 방법을 선택하였다.

타원곡선 군을 얻는 방법은 다음과 같다. K를 임의의 체라 하자. 그러면 아래의 non-homogeneous(affine) Weierstrass 방정식을 만족하는 해 $(x, y) \in K^2$와 무한원점(point at infinity) O는 K상의 타원곡선을 정의한다.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad 단,\ a_i \in K$$

이때 체 K의 표수가 2가 아니라면 $y^2 = x^3 + ax^2 + bx + c$로 변형할 수 있다(만약 표수 $K > 3$이면 $y^2 = x^3 + bx + c$로 변형할 수 있다). 그리고 체 K의 표수가 2인 경우에 이 식은 아래 식 중 어느 하나로 변환할 수 있다.

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

또는

$$y^2 + xy = x^3 + a_2x^2 + a_6.$$

만약 E가 이 곡선상에 있는 점들의 집합이라면, 덧셈연산을 잘 정의함으로써 집합 $E \cup \{O\}$는 가환군(abelian group)이 될 수 있으며 이것을 $E(K)$로 나타낸다. 이 가환군은 항등원인 무한원점 O와 함께 타원곡선군이 된다.

이제 타원곡선 상에 있는 점들간의 가법 연산을 정의한다. 다음과 같이 가법 연산을 정의하면 타원곡선 상의 점 집합은 가환군을 이룬다.

정의 E를 실수상의 타원곡선 이라 하고, P와 Q를 타원곡선 E상에 있는 2개의 점이라 한다. 여기에서 아래와 같은 방법에 의해 P의 음 $-P$와 합 $P+Q$를 정의한다.

(1) 만약 P가 무한원점 O이라면, $-P$를 O라 하고, $P+Q$를 Q라 정의한다. 즉, 무한원점 O가 여기에서 생각하고 있는 점의 군에 있어서 가법의 단위원인 zero원에 상당한다고 생각할 수 있다. 이것으로부터 뒤에 기술하는 부분에서는 두 점 P와 Q의 어느 것도 무한원점이 아니라고 가정한다.

(2) $-P$는 P와 동일한 x 좌표값을 가지며, P의 y 좌표의 음의 값을 가진다. 즉, $-(x, y) = (x, -y)$라 한다. 점 (x, y)가 곡선상의 점이라면 점 $(x, -y)$도 곡선상의 점이 되는 것은 명백하다.

(3) 만약 2점 P와 Q가 서로 다른 x좌표값을 가질 때, 직선 $l = \overline{PQ}$ 와 곡선은 반드시 또 하나의 교점 R을 가지는 것은 명백하다(단, 직선이 점 P에 있어서 곡선과 접하지 않는 것으로 한다. 만약 점 P에서 접하고 있을 경우에는 $R = P$가 되고, 점 Q에서 접하고 있을 때에는 $R = Q$로 한다). 이때 $P+Q$를 $-R$, 즉 제 3번째 교점의 x축에 대한 대칭 상 이라고 정의한다.

(4) 만약 $Q = -P$일 때(즉, Q가 P에 대해서 동일한 x 좌표값을 가지고 음인 y좌표값을 가짐), $P+Q = O$(무한원점)라 정의한다.

(5) 나머지 생각할 수 있는 경우는 $P = Q$일 때이다. 이 경우 l을 곡선상의 점 P에서의 접선이라 하고, R을 곡선과 접선 l의 점접 이외의 유일한 교점이라 하면, $P+Q = -R$이라고 정의한다(만약 접선이 곡선과 2중으로 접하고 있는, 즉 P가 변곡점(point of inflection)인 경우에는 R을 P라 한다).

이와 같은 정의를 바탕으로, K의 표수가 2인 경우에 $P+Q$의 좌표 값을 유도하면 다음과 같다. 유도과정은 [4]를 참조한다.

$$\begin{aligned} x_3 &= \left(\frac{y_1+y_2}{x_1+x_2}\right)^2 + \frac{y_1+y_2}{x_1+x_2} + x_1 + x_2 + a_2 && P \neq Q \\ &= x_1^2 + \frac{a_6}{x_1^2} && P = Q \\ y_3 &= \left(\frac{y_1+y_2}{x_1+x_2}\right)(x_1+x_2) + x_3 + y_1 && P \neq Q \\ &= x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)x_3 + x_3 && P = Q \end{aligned}$$

유도된 $P+Q$의 좌표 값에 대한 식은, K의 표수가 2이고 타원곡선이 nonsupersingular curve인 경우에 대한 것이다. 본 연구에서는 하드웨어로의 구현을 목적으로 하고 있기 때문에 이후로는 기저체 K를 표수가 2인 $GF(2^m)$형태로 생각한다.

3. Optimal Normal Bases in $GF(2^m)$

타원곡선을 이용한 암호 시스템을 설계할 경우 가장 기본이 되는 연산은 타원곡선 상에 있는 점들간의 가법연산이다. 그런데 유한체 상의 타원곡선 위에 있는 점들이 군을 형성할 수 있도록 정의된 가법연산은 2장에서 유도한 바와 같이 기저체 연산, 즉 타원곡선이 정의된 유한체의 연산으로 이루어져 있다. 그래서 타원곡선을 이용한 암호 시스템의 성능, 즉 암·복호화 속도는 기저체 연산기의 성능에 의해 결정된다. 유한체 상의 연산은, 근본적으로는 동일하지만, 체의 원소 표현법에 따라 세부적인 계산법이 달라진다. 체의 원소 표현법은 크게 polynomial basis representation과 normal basis representation이 있는데, 하드웨어 구현시에는 비트 단위의 연산이 간단해 지는, 표수 2인 체 상의 normal basis representation이 보다 적합하다. 구체적인 사항은 다음과 같다.

1. 만약 F_{2^m}이 type I ONB[1]만을 가진다면 $f(x) =$

$x^m+x^{m-1}+\cdots+x^2+x+1$ 라 한다. 그렇지 않고, F_{2^m}이 type II ONB[1]를 가진다면 다음의 점화식을 사용해 $f(x)=f_m(x)$를 계산한다.

$$\begin{aligned} f_0(x) &= 1, \\ f_1(x) &= x+1, \\ f_{i+1}(x) &= xf_i(x)+f_{i-1}(x), \quad i\geq 1. \end{aligned}$$

$f(x)$는 계수가 F_2의 원소이고 차수가 m인 다항식이다. 다항식의 집합 $\{x, x^2, x^{2^2}, \ldots, x^{2^{m-1}}\}$은 F_2상에 있는 F_{2^m}의 기저를 형성한다. 이 기저를 normal basis(정규기저)라 한다.

2. 다항식 x^{2^i}을 mod $f(x)$하여 생성된 binary vector를 i번째 열로 해서 $m\times m$ 행렬 A를 구성한다. 단, $0\leq i\leq m-1$ 이다. A의 각 엔트리는 F_2의 원소로 이루어져 있다.

3. A의 역행렬 A^{-1}를 구한다.

4. 다항식 $x\cdot x^{2^i}$을 mod $f(x)$하여 생성된 binary vector를 v라 하자. $v\cdot A^{-1}$를 i번째 열로 해서 $m\times m$ 행렬 T를 구성한다.

5. $\lambda_{ij}=T(j-i,-i)$라 한다. 단, $0\leq i,j\leq m-1$ 이다. 여기에서 $T(i,j)$는 T의 (i,j)엔트리를 나타낸다. 각각의 λ_{ij}는 F_2의 원소로 이루어져 있다. 단 하나의 j에 대해서 $\lambda_{0j}=1$가 성립하며, i가 $1\leq i\leq m-1$인 경우에는 정확히 2개의 j에 대해서 $\lambda_{ij}=1$이 성립한다. 그러므로 행렬 T에 있는 m^2개의 엔트리 중에서 $2m-1$개의 엔트리만이 1이며, 나머지는 모두 0이다. 이러한 이유 때문에 이 normal basis는 특별히 optimail normal basis(ONB)라 부른다.

이렇게 ONB representation을 사용하면 유한체 F_{2^m}에 있는 두 원소간의 가법연산(Addition)은 간단히 비트별 XOR 가 되며, 제곱연산(Squaring)은 Rotation이 된다. 또한 $a=(a_{m-1}a_{m-2}\cdots a_0)$와 $b=(b_{m-1}b_{m-2}\cdots b_0)$를 F_{2^m}의 원소라 하고 $a\cdot b=c=(c_0c_1c_2\cdots c_{m-1})$라 하면, 승법연산(Multiplication)은 다음과 같이 나타낼 수 있다.

$$c_k = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_{i+k}b_{j+k}\lambda_{ij}, \qquad 0\leq k\leq m-1.$$

단, 여기서 각 첨자는 모듈로 m을 취한다.

4. Multiplicative Inversion over ONB

임의의 체 K에서 정의된 승법연산에 대한 역원은 다음과 같이 나타낼 수 있다.

$$a\cdot a^{-1}=1. \qquad \text{단, } a\in K$$

따라서 a가 F_{2^m}의 원소일 때에도 이 식은 성립한다. 단, 이 경우 승법연산은 전술한 정의에 따르며, 승법연산에 대한 항등원 "1"은 m-tuple $(11\cdots1)$이 된다. 위의 식에 페르마 정리를 적용하면,

$$\begin{aligned} a\cdot a^{-1} &\equiv a^{2^m-1} \quad \bmod 2^m \\ a^{-1} &\equiv a^{2^m-2} \quad \bmod 2^m. \end{aligned}$$

이 된다. 지수만 살펴보면 아래식과 같이 된다.

$$-1=2^m-2$$

이 식의 우변을 적당히 인수분해 할 수 있다면 역원연산은 제곱과 승산으로 계산될 수 있다. 인수분해 과정은 다음과 같다.

$$\begin{aligned} 2^m-2 &= 2(2^{m-1}-1) \\ &= 2(2^{\frac{m-1}{2}}-1)(2^{\frac{m-1}{2}}+1), \qquad \text{if } m-1 \text{ is even} \\ &= 2(2(2^{\frac{m-2}{2}}-1)(2^{\frac{m-2}{2}}+1)+1), \text{ if } m-1 \text{ is odd} \end{aligned}$$

이 인수분해 과정은 [5]에 따른 것이며, 각각의 항에 대해서 이러한 인수분해는 반복적으로 적용가능 하다. 본 논문에서는 하드웨어 구현을 통해 연산속도를 높이는 목적으로 하고 있으므로 모든 항을 병렬로 계산하는 것이 이상적이나, 이렇게 할 경우 하드웨어 면적 및 지연이 지나치게 커져 구현에 적합하지 않다. 그래서 위의 식을 변형하여 병렬로 처리하기에 적합한 단위 항을 유도하고, 이 단위 항을 계산할 수 있는 연산기를 설계한다. 설계한 연산기를 반복적으로 사용하면 역원연산을 수행할 수 있다. 역원 연산기의 하드웨어 구현을 위해 유도한 식은 다음과 같다.

$$\begin{aligned} U_n &= (2^{s_n}+1)\cdot 2^{s_n \bmod 2}+(s_n \bmod 2) \\ U_{n-1} &= U_n\cdot(2^{s_{n-1}}+1)\cdot 2^{s_{n-1}\bmod 2}+(s_{n-1} \bmod 2) \\ &\vdots \\ U_0 &= U_1\cdot 2 \\ \text{단, } n &= \lfloor \log_2 m-1 \rfloor, s_i=\lfloor \frac{m-1}{2^i} \rfloor \text{ for } 1\leq i\leq n \end{aligned}$$

위와 같이 단위 항을 유도하면 역원연산의 결과는 $a^{-1}=a^{U_0}$ 가 된다. 각각의 단위 항 $U_x(0\leq x\leq n)$는 병렬로 수행할 연산을 의미하며 회로로 구현시 2개의 multiplication 회로, 3개의 rotation 회로, 3개의 MUX, 1개의 register로 구성된다. 각각의 비트 사이즈는 m에 의해 결정되며, 역원을 계산하는 데 걸리는 시간은 U_x 항이 n개 이므로 총 n-clock이 소요된다. 따라서 기저체가 $F_{2^{155}}$인 경우 역원은 7-clock에 계산될 수 있다.

5. Implementation onto FPGA

유도한 식을 바탕으로 설계한 승법연산기와 역원 연산기의 구조를 그림 1과 그림 2에 각각 나타내었다.

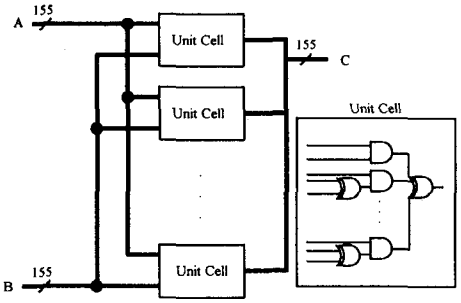

그림 1 Multiplication Logic over $F_{2^{155}}$

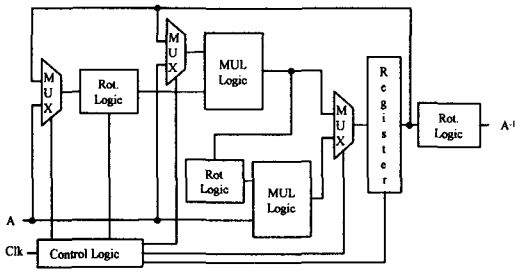

그림 2 Inversion Logic over $F_{2^{155}}$

설계한 연산기의 구조를 VHDL로 기술한 뒤, Xilinx Foundation Series 2.1i 환경에서 VIRTEX V1000FG680에 타겟팅 하여 합성하였다. 타이밍 시뮬레이션 결과를 그림 3에 나타내었다.

[5] 및 [6]에서 구현된 소프트웨어에 근거하여 올바른 연산결과를 출력함을 확인하였으며, Implementation 결과 면적은 커졌으나 속도면에서는 월등하였다.

6. Conclusion

본 논문에서는 타원곡선 암호 시스템을 구현하기 위한 일환으로 타원곡선 상의 점을 고속으로 연산 할 수 있는 전용의 기저체 연산기를 하드웨어로 구현하였다. 기저체 연산의 면밀한 분석을 통해 속도면에서 최적화된 연산기의 구조를 제안했으며, 이를 VHDL로 설계하고 VIRTEX V1000FG680에 타겟팅하여 합성한 후 타이밍 시뮬레이션을 통해 그 기능을 검증하였다. 그 결과 기존의 연구[7]에 비해 속도가, 승법연산은 약150배, 역원연산은 약 480배 향상되었으며 면적은 8.7배가 되었다. FPGA로 타겟팅 했기 때문에 최적의 속도와 최소의 면적은 아니지만, 그 구조가 규칙적이기 때문에 차후 VLSI로의 적용이 용이하다. 구현한 전용의 연산기는 타원곡선을 이용한 암호 시스템 설계를 위한 VLSI 구현에 기초자료로 활용되거나, 직접 coprocessor 형식으로 임베드되어 사용될 수 있을 것이다. 디자인한 연산기는 제11회 MPW 설계 공모전에 선정되어 현재 삼성 0.5um SOG 공정으로 칩 제작 중에 있다.

References

[1] Working Draft IEEE P1363, Part 6.

[2] http://www.kordic.re.kr/

[3] http://www.etimesi.com/

[4] Neal Koblitz, "A Course in Number Theory and Cryptography", Springer-Verlag pub., 1994.

[5] Michael Rosing, "Implementing Elliptic Curve Cryptography", Manning pub., 1999.

[6] Certicom ECC Whitepaper, 1999.

[7] Sarwono Sutikno, Ronny Effendi, Andy Surya, "Design and Implementation of Arithmetic Processor $F_{2^{155}}$ for Elliptic Curve Cryptosystems", IEEE APCCAS, 1998, pp. 647-650.

[8] Robert J., McEliece, "Finite Fields for Computer Scientists and Engineers", Kluwer Academic Publishers, 1989.

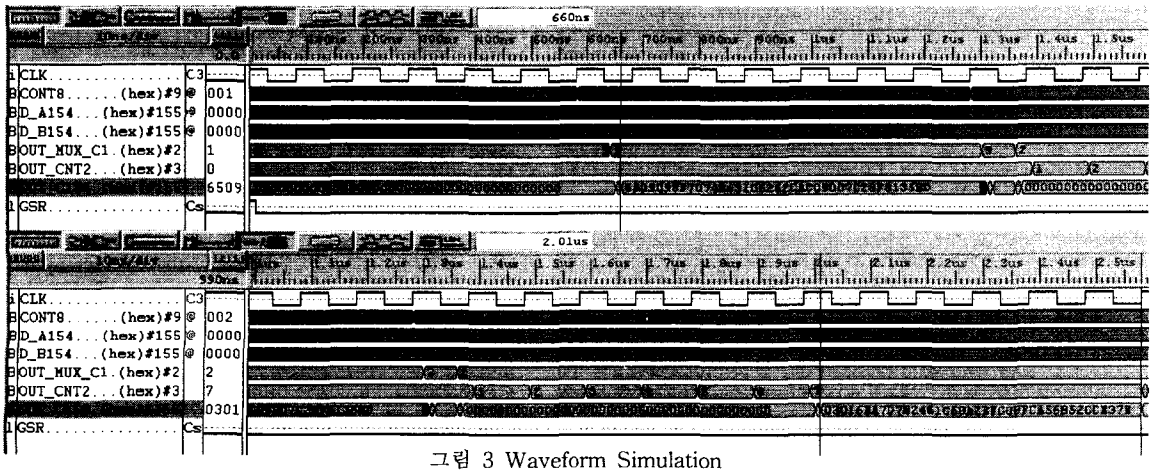

그림 3 Waveform Simulation