IC 카드를 위한 polynomial 기반의 타원곡선 암호시스템 연산기 설계

최 용 제, 김 호 원, 김 무 섭, 박 영 수
한국전자통신연구원
전화 : 042-860-1327 팩스 : 042-860-5611

Design of Elliptic Curve Cryptographic Coprocessor over binary fields for the IC card

YongJe Choi, HoWon Kim, MooSeop Kim, and YoungSoo Park
Electronics and Telecommunications Research Institute
E-mail : choiyj@etri.re.kr

Abstract

This paper describes the design of elliptic curve cryptographic (ECC) coprocessor over binary fields for the IC card. This coprocessor is implemented by the shift-and-add algorithm for the field multiplication algorithm. And the modified almost inverse algorithm(MAIA) is selected for the inverse multiplication algorithm. These two algorithms is merged to minimize the hardware size. Scalar multiplication is performed by the binary Non Adjacent Format(NAF) method. The ECC we have implemented is defined over the field GF(2^{163}), which is a SEC-2 recommendation[7].

I. 서론

타원곡선 암호시스템(Elliptic Curve Cryptosystem : ECC)은 1985년에 Neal Koblitz[3]와 Victor Miller[4]에 의해서 각각 독립적으로 제안된 암호시스템이다. 이 암호시스템은 타원곡선 상의 점들의 이산 대수의 어려움에 안전성의 근간을 두고 있으며, 기존의 공개키 암호시스템들 보다 비트당 안전성이 높은 암호시스템으로 알려져 있다. 비트당 안전성이 높기 때문에 타원곡선 암호시스템은 Diffe-Hellman 키 교환, ElGamal, Schnorr, ECDSA, DSS, Massey-Omura와 같은 디지털 서명 프로토콜이나 암호화 프로토콜을 다른 공개키 암호시스템에 비해 상대적으로 작은 크기의 키 값으로도 충분한 안정성을 보장한다[8].

본 논문에서는 IC 카드를 위한 polynomial 기반의 타원곡선 암호시스템 연산기를 설계하였다. 일반적으로 타원곡선 암호시스템은 작은 크기의 키 값으로도 높은 안전성을 보장하기 때문에 소프트웨어적인 구현만으로도 만족할 만한 성능을 얻을 수 있다. 하지만 IC 카드에서는 메모리 용량의 한계와 연산 능력의 제한 때문에 소프트웨어적인 구현만으로는 충분한 성능을 기대하기가 힘들다[5]. 본 논문에서는 이러한 IC 카드에서 타원곡선 암호시스템 구현이 용이한 타원곡선 암호 프로세서를 설계하고자 한다.

II. 타원곡선 암호시스템

일반적으로, 공개키 암호시스템으로는 소인수 분해의 어려움에 근거한 시스템(RSA)[6]과 이산대수 문제의 어려움에 근거한 시스템(DSA), 그리고 타원곡선상의 이산대수 문제에 근거한 시스템(ECC)이 주로 사용된다. 이중 타원곡선 암호시스템 비트당 안전도가 가장 높은 암호 시스템으로 작은 사이즈의 키 값만으로도 높은 안전성을 보장한다. [표 1]은 타원곡선 암호시스템(ECC), RSA, DSA의 비트당 안전성을 비교한 표이다[1].

[표 1] RSA, DSA, ECC 안전성 비교

Time to break in MIPS years	RSA/DSA key size	ECC key size	RSA/ECC key size ratio
10^{9}	512	106	5 : 1
10^{8}	768	132	6 : 1
10^{11}	1,024	160	7 : 1
10^{20}	2,048	210	10 : 1
10^{78}	21,000	600	35 : 1

모듈러 지수승 연산이 RSA 암호시스템의 성능을 좌우하듯이 타원곡선 암호시스템의 성능은 스칼라 곱셈 연산에 의하여 좌우된다. 스칼라 곱셈 연산은 임의의 랜덤수 k와 타원 곡선 위의 한점 P의 곱셈 연산으로 정의되며, 타원곡선 위의 점 P의 k번 덧셈연산으로 계산된다. 이때 타원곡선의 덧셈연산은 결과값이 다시 타원곡선 위의 점이 되도록 [알고리즘 1]과 같이 정의 되어야 한다. (단, Polynomial 기반 유한체를 위한 타원곡선식은 $y^2 + xy = x^3 + ax^2 + b$ 과 같이 주어지며, P_1과 P_2은 이 타원곡선 상의 존재한다.)

[알고리즘 1] Point Addition Equation

Input : $P_1 = (x_1 , y_1)$, $P_2 = (x_2 , y_2)$.
Output : $P_3 = P_1 + P_2 = (x_3 , y_3)$.

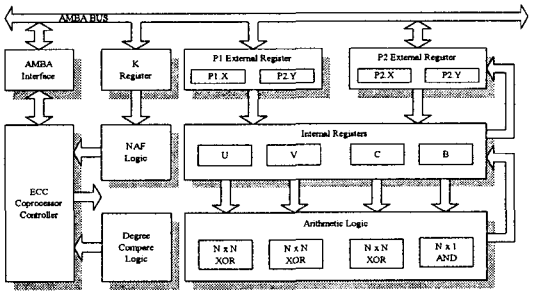
1. If $P_1 = P_2$ (doubling)
   $x_3 = \lambda^2 + \lambda + a$,
   $y_3 = x_1^2 + \lambda(\lambda + 1) x_3$
   where ($\lambda = x_1 + y_1 / x_1$)
2. Else if $P_1 \neq P_2$ (point addition)
   $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$,
   $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$
   where ($\lambda = (y_2 + y_1) / (x_2 + x_1)$)
3. Return (x_3 , y_3)

III. 타원곡선 암호 프로세서 설계

본 논문에서는 SEC2[7]에서 제안된 163비트 polynomial basis 타원곡선 암호시스템을 위한 암호 프로세서를 설계하였다. 163비트 타원곡선 암호시스템은 [표 1]에서 보는 바와 같이 RSA 1024비트 이상의 안정성을 보장하게 된다. 본 타원곡선 암호시스템의 파라미터는 [표 2]와 같다. 이와 같이 정의된 타원곡선 암호시스템을 위하여 [그림 1]과 같은 구조를 가지는 암호프로세서를 설계하였다. 이 암호프로세서는 타원곡선의 scalar 곱셈(kP 연산)과 타원곡선 덧셈 연산을 수행한다. (엄밀히 말하면 scalar 곱셈연산은 타원곡선 덧셈 연산들의 조합이지만, 두 가지 연산의 입출력이 다르므로 서로 구분 지었다.) 실제 상위 프로토콜에서는 이러한 두 가지 연산만을 필요로 하므로, 연산에 필요한 유한체 곱셈, 역승산, 덧셈, 제곱과 같은 연산은 내부적으로만 수행하도록 되어있다.

[표 2] GF(2^{163}) 타원곡선 암호시스템 파라미터

Reduction Polynomial $f(x)$	$x^{163} + x^7 + x^6 + x^3 + 1$
Parameter a	07 B6882CAA EFA84F95 54FF8428 BD88E246 D2782AE2
Parameter b	07 13612DCD DCB40AAB 946BDA29 CA91F73A F958AFD9
Order of a base point n	03 FFFFFFFF FFFFFFFF FFFF48AA B689C29C A710279B
x coordinate of base point	03 69979697 AB438977 89566789 567F787A 7876A654
y coordinate of base point	435EDB42 EFAFB298 9D51FEFC E3C80988 F41FF883



[그림 1] 타원곡선 암호프로세서 구조

타원곡선 덧셈연산을 위한 [알고리즘 1]에서 가장 주요한 연산은 유한체 곱셈연산과 역승산 연산이다. 본 타원곡선 암호프로세서에서는 곱셈연산은 shift-and-add method를 이용하여 구현하였으며, 역승산 연산은 Modified Almost Inverse Algorithm(MAIA)으로 구현하였다. [알고리즘 1]에서 알 수 있듯이 타원곡선 덧셈연산을 위해서는 유한체 덧셈 연산과 제곱 연산도 필요하지만, polynomial 기반 유한체의 특성상 유한체 덧셈연산은 두 수의 xor 연산으로 쉽게 구현될 수 있다. (실제로 알고리즘 1에서 존재하는 덧셈 연산들은 제어기에서 [그림 1]의 내부 XOR 연산기들과 레지스터들을 적절히 이용하여 수행된다.) 그리고 유한체 제곱 연산은 연산 자체는 유한체 곱셈연산보다 매우 단순하지만 하드웨어적인 구현에서는 별도의 유한체 제곱 연산기를 설계하는 것과 유한체 곱셈연산기의 두 입력에 같은 값을 두 번 입력하여 연산하나 별 차이가 없게된다. 타원곡선 salar 곱셈 연산을 위해서는 binary NAF method가 이용되었다.

3.1 유한체 곱셈 연산

Polynomial 기반 유한체의 곱셈 연산 알고리즘으로는 shift-and-add method, comb method, comb method with windows, partial(parallel) method 등 다양한 알고리즘들이 존재한다[2]. 이중 comb method와 comb method with windows는 쉬프트 연산을 줄이거나, 필요한 연산을 미리 수행하여 메모리에 저장하는 방식으로 연산 속도를 증가시키는 방법으로 소프트웨어적인 구현에서 효율적으로 사용된다.

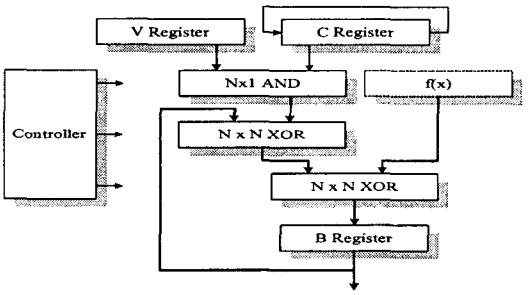
본 타원곡선 암호프로세서에서 구현알고리즘으로 선택한 shift-and-add method는 가장 일반적인 유한체 곱셈 연산 알고리즘이다. 이는 $a \cdot b \bmod f(x)$가 $(a_{m1} x^{m1} b + \ldots + a_2 x^2 b + a_1 x b + a_0 b) \bmod f(x) = (a_{m1} x^{m1} b \bmod f(x)) + \ldots + (a_2 x^2 b \bmod f(x)) + (a_1 x b \bmod f(x)) + (a_0 b \bmod f(x))$ 과 같이 연산될 수 있음을 이용한 알고리즘으로 덧셈 연산(알고리즘 2의 2.1)과 동시에 reduction 연산(알고리즘 2의 2.2)을 수행할 수 있다. 이러한 구조는 하드웨어로의 구현이 매우 용이하며, 비트 길이 만큼의 연산 시간만이 소요된다. (만약 163비트의 타원곡선 암호시스템이라면 곱셈연산은 163 클럭이 필요하게 된다. 단, 입출력 시간은 제외되었다.)

알고리즘 2. Shift-and-add field multiplication

Input : Binary Polynomials $a(x)$ and $b(x)$.
Output : $c(x) = a(x) \cdot b(x) \bmod f(x)$.
1. $c \leftarrow 0$.
2. For i from $m-1$ to 1 do
   2.1 If $a_i = 1$ then $c \leftarrow c + b$.
   2.2 $c \leftarrow c \cdot x \bmod f(x)$.
3. If $a_0 = 1$ then $c \leftarrow c + b$.
4. Return (c).

이러한 구조의 유한체 곱셈 연산기는 뒤에서 언급될 역승산기와 쉽게 병합이 가능하며, 임의의 타원곡선을 모두 수용할 수 있도록 확장이 용이하다는 장점이 있다.

본 타원곡선 암호프로세서는 곱셈 연산을 위해서 [그림 1]의 암호프로세서 구조도에서 v, c, b 레지스터들과 두 개의 NxN XOR 연산기와 Nx1 AND 연산기가 사용된다. 곱셈 연산시 동작 블록도는 [그림 2]와 같다. ([그림 2]의 블록도는 다른 유한체 연산을 위한 mux 신호는 생략되었다.) 앞에서 언급된 바와 같이 본 하드웨어는 파라미터들이 고정되어 있어서 reduction 함수 $f(x)$ 값을 암호프로세서에 이미 내장하고 있지만, 암호프로세서가 동작하기 전에 함수 값을 직접 받아들이도록 하면 일정 비트 길이 내의 임의의 타원곡선 암호시스템을 수용할 수 있는 구조로 쉽게 확장할 수 있다.



[그림 2] 유한체 곱셈기 동작 블록도

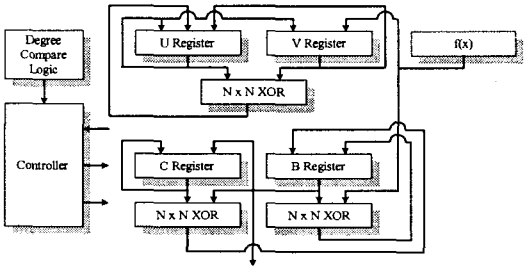
3.2 유한체 역승산

Polynomial 기반 유한체의 역승산 연산 알고리즘으로는 Extended Euclidean Algorithm (EEA)[9]과 Almost Inverse Algorithm (AIA)[8]등이 있다. 소프트웨어적인 구현에서는 두 가지 알고리즘이 유사한 성능을 보인다. 하지만 하드웨어적인 구현에서는 한 비트씩 연산을 수행하는 AIA가 더 용이하다. 더욱이 알고리즘 3의 Modified Almost Inverse Algorithm (MAIA)은 reduction 연산이 따로 필요 없어 매우 효율적이다.

알고리즘 3. Modified Almost Inverse Algorithm

Input : $a \in F_2^m$, $a \neq 0$.
Output : $a^{-1} \bmod f(x)$.
1. $b \leftarrow 1$, $c \leftarrow 0$, $u \leftarrow a$, $v \leftarrow f$, $k \leftarrow 0$.
2. While x divides u do :
   2.1 $u \leftarrow u / x$.
   2.2 If x divides b then $b \leftarrow b / x$;
   else $b \leftarrow (b + f) / x$.
3. If $u = 1$ the return (b).
4. If $\deg(u) < \deg(v)$ then : u ↔ v, b ↔ c.
5. $u \leftarrow u + v$, $b \leftarrow b + c$.
6. Goto step 2.

본 타원곡선 암호프로세서는 역승산 연산을 위해서 [그림 1]의 암호프로세서 구조도에서 u, v, c, b 레지스터들과 3개의 NxN XOR 연산기, 그리고 degree compare logic이 이용된다. 역승산기의 동작 블록도는 [그림 3]과 같다.



[그림 3] 유한체 역승산기 동작 블록도

3.3 타원곡선 scalar 곱셈연산

만약 $k = 1111_2 = 15$이면 이는 $(10000 - 1)_2 = 16 - 1$로 다시 표현할 수 있다. 만약 "-"에 대한 연산을 정의할 수 있다면 이런 방식의 표현 기법은 타원곡선 scalar 곱셈연산의 연산 횟수를 줄일 수 있다. (이와 같은 표현을 Non Adjacent Format(NAF)라 한다.) 다행히 타원곡선 상에서 $-P = (x, x + y)$로 정의되며, 이러한 NAF방식을 이용한 scalar 곱셈 알고리즘은 다음과 같다.

알고리즘 4. Binary NAF method

Input : $\mathrm{NAF}(k) = \sum_{i=0}^{t-1} k_i \cdot 2^i$, $P \in E(F_2^m)$.

Output : kP.

1. $Q \leftarrow O$.
2. For i from $t - 1$ downto 0 do
   2.1 $Q \leftarrow 2Q$.
   2.2 If $k_i = 1$ then $Q \leftarrow Q + P$.
   2.3 If $k_i = 1$ then $Q \leftarrow Q - P$.
3. Return (Q).

본 암호프로세서 NAF 로직에서 NAF 변환을 수행하며, 알고리즘 4의 연산 스케줄링은 제어기에서 담당한다. 실제 NAF 로직은 매우 단순한 형태로 구현이 되었으며, 이와 같은 방법을 이용하여 163비트의 nP(P : base point, n : P의 order) 계산 시, 20MHz 클럭에서 연산시간을 11.9msec으로 단축할 수 있었다. 이는 단순 binary method로 15.6msec이 소요된다.

Ⅳ. 타원곡선 암호프로세서 성능 검증

본 타원곡선 암호프로세서는 ECDSA를 구현하여 성능을 비교·검증하였다. Micro code는 ARM 프로세서에서 기본적인 알고리즘들을 이용하여 소프트웨어로만 구현되었을 때의 결과이다. 암호프로세서는 Xillinx Virtex-1000 FPGA를 이용하여 구현되었다. 결과는 다음과 같다.

[표 3] Performance of the ECDSA

	Micro Code (ARM7M) (30MHz)	Crypto coprocessor (20MHz)
Scalar multiplication	1.004 sec	12.9 msec
ECDSA Sig. Generate.	1.032 sec	13.8 msec
ECDSA Sig. Verify	2.255 sec	27.3 msec

Ⅴ. 결론

본 논문에서는 SEC2에서 제안된 163비트 polynomial basis 타원곡선 암호시스템을 위한 암호 프로세서를 설계하였다. 본 타원곡선 암호프로세서는 단순 입출력 인터페이스 소프트웨어만으로 타원곡선 scalar 곱셈 연산과 덧셈 연산을 수행할 수 있다. 본 암호프로세서는 xillinx FPGA에서 20MHz로 동작하며, HYNDAI 0.5m CMOS 공정으로 AMBA 인터페이스까지 포함하여 24k gates의 면적을 가진다.

본 타원곡선 암호시스템은 여러 가지 구현 알고리즘을 비교 분석하여 IC카드에 적합한 구조를 택하였다. 하지만, 더욱 최적인 구조를 찾기 위해서는 좀더 많은 알고리즘과 구조의 비교 분석이 이루어져야 할 것이다.

참고문헌

[1] Certicom research , *The Elliptic Curve Cryptosystem*, Certicom, April 1997.

[2] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, *Software Implementation of Elliptic Curve Cryptography over Binary Fields*, CHES 2000, page 1-24. 2000.

[3] N. Koblitz, *Elliptic curve cryptosystems*, Mathematics of Computation, number 48, pages 203-209, 1987.

[4] V.S. Miller, *Use of elliptic curve in cryptography*, Advances in Cryptology Proceedings of CRYPTO'85, Springer Verlag Lecture Notes in Computer Science 218, pages 417-426, 1986.

[5] W. Rnakl, W. Effing, Smart Card Handbook, JOHN WILEY & SONS, 1997.

[6] R.L. Rivest, A. Shamir, and L.M. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, volume 21, pages 120-126, February 1978.

[7] Certicom research, "*SEC 2 : Recommended Elliptic Curve Domain Parameters*", October 1999.

[8] Richard Schroeppel, Hilarie Orman, Sean O'Malley, "*Fast Key Exchange with Elliptic Curve Systems*", TR-95-03(Tucson, AZ: University of Arizona, Computer Sciences Department, 1995)

[9] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC press, 1997.