타원곡선 암호를 위한 GF(2^{163}) 스칼라 곱셈기

정상혁* · 신경욱*

*금오공과대학교

A GF(2^{163}) scalar multiplier for elliptic curve cryptography

Sang-Hyeok Jeong* · Kyung-Wook Shin*

* School of Electronic Eng., Kumoh National Institute of Technology

E-mail : jsh20907@kumoh.ac.kr

요 약

본 논문에서는 타원곡선 암호를 위한 스칼라 곱셈기의 설계에 대해 기술한다. 설계된 스칼라 곱셈기는 스마트카드 표준에 기술된 163-비트의 키 길이를 가진다. 유한체 $GF(2^{163})$ 상에서 스칼라 곱셈의 연산량을 줄이기 위해 complementary recoding 방식을 적용한 Non-Adjacent-Format(NAF) 변환 알고리듬을 적용하여 설계하였다. 설계된 스칼라 곱셈기 코어는 0.35-㎛ CMOS 셀 라이브러리로 합성하여 32,768 게이트로 구현되었으며, 150-MHz@3.3-V로 동작한다. 설계된 스칼라 승산기는 스마트카드용 타원곡선 암호 하드웨어 구현을 위한 IP로 사용될 수 있다.

ABSTRACT

This paper describes a scalar multiplier for Elliptic curve cryptography. The scalar multiplier has 163-bits key size which supports the specifications of smart card standard. To reduce the computational complexity of scalar multiplication on finite field $GF(2^{163})$, the Non-Adjacent-Format (NAF) conversion algorithm based on complementary recoding is adopted. The scalar multiplier core synthesized with a 0.35-㎛ CMOS cell library has 32,768 gates and can operate up to 150-MHz@3.3-V. It can be used in hardware design of Elliptic curve cryptography processor for smart card security.

키워드

공개키 암호화, 스칼라 곱셈, 타원곡선 암호, Complementary recoding

Ⅰ. 서 론

인터넷의 대중화와 네트워크 기술의 발달로 유무선 네트워크를 통한 정보교류와 전자금융 및 전자상거래가 활발하게 이루어지고 있다. 따라서 모바일 환경에서 정보보호를 위한 보안기술의 중요성이 증대되고 있으며, 사용자 인증과 서명, 무결성 검증 등을 위한 공개키 암호 알고리듬의 중요성이 더욱 커지고 있다. 대표적인 공개키 암호 알고리듬으로 RSA와 타원곡선 암호 (Elliptic Curve Cryptography; ECC) 알고리듬이 널리 사용되고 있다. 특히 타원곡선 암호 알고리듬은 키 길이의 증가에 따라 안전도가 지수적으로 증가하며, 163-비트의 키 길이를 갖는 ECC는 1024-비트의 키 길이를 갖는 RSA 알고리듬과 유사한 안전도를 가지는 것으로 평가되고 있다[1]. 따라서 ECC는 메모리 용량과 처리 속도가 제한된 모바일 환경에 적합한 차세대 공개키 암호 시스템의 표준으로 자리 잡아 가고 있다.

본 논문에서는 타원곡선 암호 프로세서의 핵심 블록인 스칼라 승산기를 구현하였으며, 스칼라 승산기의 연산량 감소를 통한 효율적인 구현을 위해 complementary recoding (CR) 변환 방식을 적용하여 설계하였다.

Ⅱ. 타원곡선 암호 알고리듬

타원곡선 암호 알고리듬은 타원곡선 이산로그 문제에 근간을 두고 있다. 이는 타원곡선 상의 임의의 한 점 P에 정수 k를 곱한 값이 $Q=kP$일때, 점 Q와 P를 알고 있어도 정수 k를 계산하기 어렵다는 것을 의미한다[2]. 이러한 타원곡선 암호 알고리듬의 핵심적인 연산과정은 $Q=kP$를 구하는 스칼라 곱셈 연산이다. 스칼라 곱셈은 소수 필드(prime field)와 유한체 (finite field) $GF(2^m)$상에서 모두 정의가 가능하다. 그러나 캐리전달이 필요한 소수 필드 대신에, 캐리전달이 없고 하드웨어 구현이 용이한 유한체 상에서의 연산이 일반적으로 사용된다.

III. 타원곡선 스칼라 곱셈

타원곡선 암호 시스템에서 암호화 연산과 복호화 연산은 스칼라 곱셈으로 이루어진다. 타원곡선 스칼라 곱셈을 구현하는 가장 기본적인 방법은 double-and -add 방법이며, 점 두배 연산과 점 덧셈 연산의 반복 과정으로 계산된다. 한편, 점 덧셈과 점 두배 연산은 유한체 상의 덧셈, 곱셈, 나눗셈 연산으로 구성되며, 따라서 타원곡선 스칼라 곱셈을 계산하기 위한 전체적인 연산 구조는 그림 1과 같다.

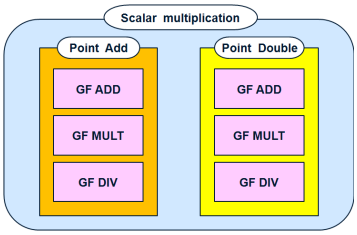


그림 1. 타원곡선 스칼라 곱셈의 연산 구조

m-비트 타원곡선 스칼라 곱셈 알고리듬은 최소 $(m-1)$회의 점 두배 연산과 k의 hamming weight 만큼의 점 덧셈 연산이 필요하다. 점 덧셈 연산을 줄이기 위한 방법으로 Non-Adjacent-Format (NAF) 변환이 제안되었다[3]. 스칼라 k가 NAF 형태로 변환되면 k의 hamming weight가 감소되어, 스칼라 곱셈의 연산량이 줄어들게 된다[4,5]. NAF 변환을 이용한 타원곡선 스칼라 곱셈의 연산과정은 알고리듬 1과 같다.

알고리듬 1. NAF 변환을 이용한 타원곡선 스칼라 곱셈

$NAF(k) = \sum_{i=0}^{m-1} k_i 2^i$
$Q = 0$
for i form m-1 downto 0 begin
  if $k_i = 1$ *then* $Q = Q + P$
  if $k_i = -1$ *then* $Q = Q - P$
  $Q = 2Q$
end

NAF 변환을 이용한 스칼라 곱셈을 위해서는 우선 이진형태로 주어진 k 값을 redundant 형태의 NAF로 변환해야 한다. NAF 형태로 변환할 경우 k의 hamming weight가 감소되어 연산량 이 줄어들게 되지만, 이 경우 -1을 계산하기 위해 점 뺄셈 연산이 추가된다. 점 뺄셈 연산은 기존의 점 덧셈 연산과 점 역원 연산으로 이루어지며 점 역원 연산은 XOR 연산으로 이루어진다. NAF 변환을 위한 여러 가지 알고리듬들이 제안되고 있으며, 본 논문에서는 CR 방법[4,5]을 적용하여 설계하였다. CR은 2003년 Chang에 의해 제안된 알고리듬으로서 이진형태의 k가 주어질 때 이것의 보수를 이용하여 NAF 형태로 변환하며, 알고리듬 2와 같다.

알고리듬 2. Complementary recoding

$k = \sum_{i=0}^{m-1} b_i 2^i \ (b_i \in 0,1)$
$c = \overline{k}$
$s = 2^m - c - 1$
return s

IV. 하드웨어 구현

본 논문에서 설계된 타원곡선 스칼라 곱셈기는 CR 변환을 이용하여 유한체 $GF(2^{163})$ 상에서 $Q = kP$를 계산한다. 내부 구조는 그림 2와 같으며, 입·출력 인터페이스 블록, CR 변환기, 163-비트 유한체 ALU, 그리고 중간 결과값을 저장하기 위한 레지스터와 제어회로 등으로 구성된다. 입·출력 인터페이스 블록은 타원곡선 방정식의 계수 a와 f를 저장하는 레지스터와 데이터의 입·출력을 제어하는 회로로 구성된다. 내부 레지스터 ACC_Reg는 스칼라 곱셈을 위한 데이터인 타원곡선 상의 초기 좌표값 P_x, P_y, 그리고 스칼라 k 값과 함께 중간 결과값을 저장한다. 유한체 ALU 블록은 유한체 $GF(2^{163})$ 상에서의 덧셈, 곱셈, 나눗셈 연산을 수행하는 회로들로 구성된다.

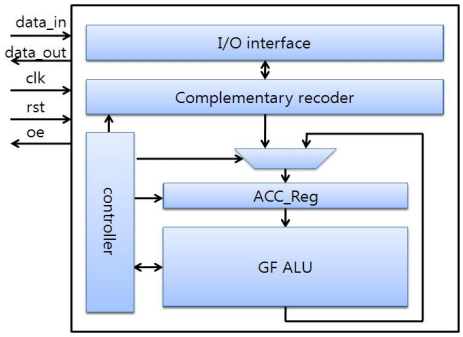


그림 2. 설계된 타원곡선 스칼라 곱셈기의 구조

초기에 타원곡선 방정식의 계수 a와 f, 타원곡선 상의 초기 좌표값 P, 그리고 유한체 연산을 위한 기약다항식의 계수, 그리고 스칼라 k가 data_in 포트를 통해 순차적으로 입력된다. 스칼라 k는 CR 변환기에 의해 NAF로 변환되며, 변환된 NAF(k)의 값에 따라 점 덧셈, 점 두배 연산, 점 역원 연산이 반복적으로 수행되어 스칼라 곱셈이 이루어진다. 한편, 점 덧셈, 점 두배, 점 역원 연산들은 유한체 $GF(2^{163})$ 상에서의 덧셈, 곱셈, 나눗셈 연산들로 구성된다.

4.1 유한체 연산기

유한체 연산기 블록 GF_ALU는 유한체 $GF(2^{163})$ 상에서 덧셈, 곱셈, 나눗셈 연산을 수행하며, 그림 3과 같은 구조로 설계되었다. 유한체 덧셈 연산은 XOR 게이트로 구현되었으며, 유한체 곱셈과 나눗셈은 시스톨릭 어레이(systolic array)의 단일 행으로 구현되었다.

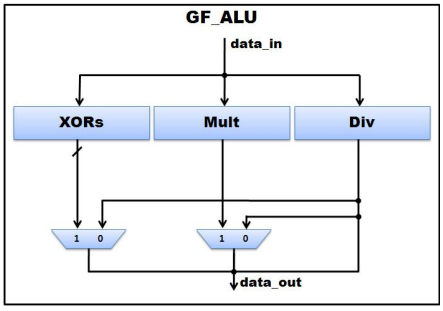
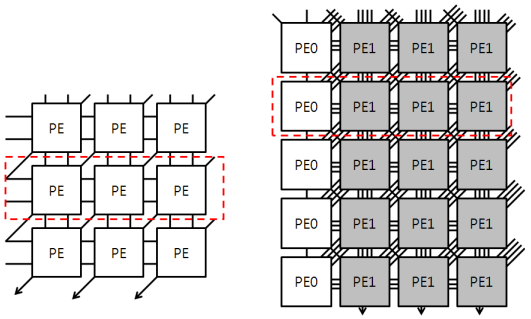


그림 3. GF_ALU의 구조

유한체 $GF(2^m)$ 상의 곱셈기는 그림 4-(a)와 같이 $m \times m$ 크기의 시스톨릭 어레이로 구현될 수 있으며, 나눗셈기는 그림 4-(b)와 같이 $(m+1) \times (2m-1)$ 크기의 시스톨릭 어레이로 구현될 수 있다. 일반적으로, 유한체 $GF(2^m)$ 상의 곱셈과 나눗셈 연산을 위한 완전 병렬 시스톨릭 어레이는 m^2에 비례하는 크기를 가지게 되며, 이를 직접 구현하기 위해서는 매우 큰 하드웨어를 필요로 하게 된다. 본 논문에서는 그림 4의 시스톨릭 어레이에서 단일 행 블록(점선으로 표시된 부분)을 사용하여 반복적인 연산으로 유한체 곱셈과 나눗셈이 계산되는 부분병렬 구조로 구현하였다.

그림 3에서 유한체 곱셈을 연산하는 Mult 블록은 (163열x1행)의 PE 어레이로 구성되며, 163번의 반복 연산으로 유한체 곱셈이 계산된다. 유한체 나눗셈을 연산하는 Div 블록은 1개의 PE0과 (163열×1행)의 PE1 어레이로 구성되며, 325 (=163×2−1)번의 반복 연산으로 유한체 나눗셈이 계산된다 [6].



(a) 유한체 곱셈 (b) 유한체 나눗셈

그림 4. 유한체 연산을 위한 시스톨릭 어레이 ($GF(2^3)$의 경우)

$GF(2^{163})$ 상의 유한체 곱셈 연산은 163 사이클의 반복 연산과 데이터 로딩 과정을 포함해서 총 164 클록이 소요된다. 유한체 나눗셈 연산은 유한체 곱셈에 비해 중간 결과로 출력되는 변수가 매우 많기 때문에 실질적으로 피드백 루프를 이루는 레지스터의 길이는 유한체 나눗셈을 기준으로 결정되어야 한다. 이때, 레지스터 길이는 $GF(2^{163})$ 상에서 $7m+1=1{,}142$ 비트가 필요하다. 또한, $GF(2^{163})$ 상의 유한체 나눗셈 연산은 325 사이클의 반복 연산과 데이터 로딩 과정을 포함해서 총 326 클록이 소요된다.

4.2 Complementary recoder

Complementary recoder는 스칼라 k의 hamming weight를 줄여 스칼라 곱셈의 연산을 간소화시키는 블록이다. 알고리듬 2를 구현하기 위해 그림 5와 같이 $c=\bar{k}$를 만들기 위한 163개의 인버터와 1을 가산하는 Add_one 회로로 구성된다.

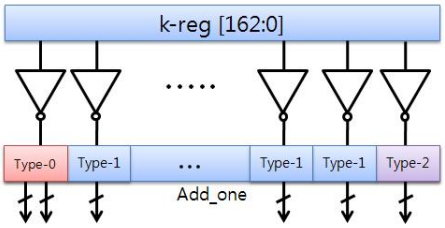
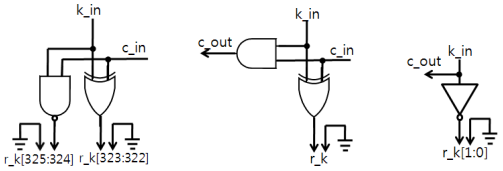


그림 5. Complementary recoder

$\bar{k}+1$을 계산하는 Add_one 블록은 피가수가 1로 고정되어 있으므로, 간략화된 리플캐리 (ripple-carry) 가산기로 구현될 수 있으며, MSB를 계산하는 Type-0 셀, LSB를 계산하는 Type-2 셀, 그리고 나머지 비트들을 계산하는 Type-1 셀이 그림 6과 같이 구현되었다. 한편, Type-0 셀은 MSB에서 발생되는 캐리 출력을 고려하기 위하여 Type-1과 다르게 NAND 게이트가 사용된다.



(a)Type-0 (b)Type-1 (c)Type-2

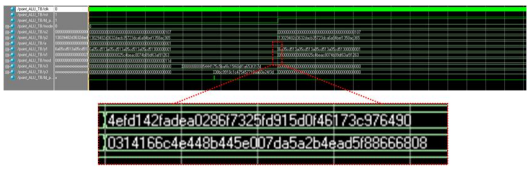
그림 6. Type-0, 1, 2 셀

한편, NAF로 변환된 스칼라 k는 $k_i \in \{-1,0,1\}$인 signed-digit 값을 가지므로, 2-비트의 코드로 표현되어야 한다. CR 알고리듬은 최상위 digit를 제외한 나머지 digit들은 항상 0 또는 -1이 되는 특성을 가지므로, 본 논문에서는 $k_i \in \{-1,0,1\}$를 $-1=10_2$, $0=00_2$, $1=01_2$로 정의하였다.

V. 설계 검증 및 성능 평가

설계된 타원곡선 스칼라 곱셈기 코어는 Verilog HDL을 이용하여 RTL 수준에서 모델링되었으며, ECC 관련 표준 문서에 제시된 테스트 벡터를 이용하여 기능검증을 수행하였다. 시뮬레이션 결과는 그림 7과 같으며, a와 f는 타원곡선을 정의하는 방정식의 계수이고, k는 스칼라 값, P_x, P_y는 타원곡선 상의 초기 좌표값을 나타내며, Q_x, Q_y는 스칼라 곱셈의 결과인 $Q=kP$의 좌표값을 나타낸다. 표준 문서에서 제시

된 결과와 동일한 시뮬레이션 결과가 출력되어 설계된 회로가 정상 동작함을 확인하였다.



a :2 5c4beac8 074b8c2d 9df63af9 1263eb82 29b3c967
f :8 00000000 00000000 00000000 00000000 00000107
k :6 a05cd513 a05cd513 a05cd513 a05cd513 00000001
P_x:1 3029482d 3635dacb 35723dca 6a84bef1 358ac365
P_y:2 479bc536 a5927c38 4795bde0 7325fd91 598cfe32
Q_x:4 efd142fa dea0286f 7325fd91 5d0f4617 3c976490
Q_y:3 14166c4 e448b445 e007da5a 2b4ead5f 88666808

그림 7. 설계된 스칼라 곱셈기의 기능검증 결과

설계된 타원곡선 스칼라 곱셈기 코어를 0.35-㎛ CMOS 셀 라이브러리로 합성한 결과 32,768개의 게이트로 합성되었다. 타이밍 분석 결과, 150-MHz로 동작 가능하여 스칼라 곱셈에 약 1.31-㎳가 소요될 것으로 평가되었다. 표 1은 설계된 스칼라 곱셈기의 성능을 참고문헌의 사례와 비교한 것이며, 그림 8은 문헌[8]의 성능에 대해 정규화된 면적, 연산시간, 면적-시간 곱 성능을 비교한 것이다. 본 논문의 스칼라 곱셈기는 문헌[8]의 경우와 비교하여 게이트 수가 약 36% 많이 소요되지만, 스칼라 곱셈에 소요되는 클록 수와 20MHz 환산 수행시간이 약 29% 감소되어 고속 연산이 가능하며, 면적-시간 곱 성능은 문헌[8]와 유사함을 보이고 있다.

표 1. 스칼라 승산기의 성능 비교

	gates	cycles	20MHz 환산 수행시간 [㎳]
$GF(2^{167})$[7]	20k	526,718	26.34
$GF(2^{163})$[8]	24k	258,000	12.9
본 논문의 경우	32.8k	183,400	9.17

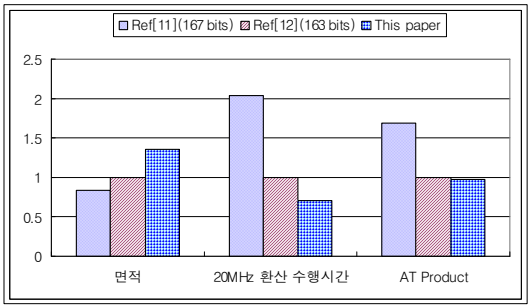


그림 8. 설계된 스칼라 곱셈기의 시간-면적 곱 성능

Ⅵ. 결론

본 연구에서는 차세대 공개키 암호 시스템에 사용되는 타원곡선 암호 시스템을 위한 스칼라 곱셈기를 설계하였다. 스칼라 곱셈을 수행하기 위해 유한체 연산기를 이용해 점 덧셈 연산기와 점 두배 연산기를 구현하였으며, 스칼라 k를 NAF로 변환하여 연산량을 감소시켰다. NAF 변환을 위해 CR 알고리듬을 이용하였으며, $GF(2^{163})$에서 최적화된 회로를 구현하였다. 설계된 타원곡선 스칼라 곱셈기는 32,768개의 게이트로 구현되었으며, 150-MHz@3.3-V로 동작 가능하여 스칼라 곱셈에 약 1.31-㎳가 소요될 것으로 평가되어 스마트카드 보안용 하드웨어 설계에 IP 형태로 사용될 수 있을 것이다.

참고문헌

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

[2] A. Menezes, *Elliptic curve pubic key cryptosystem*, Kluwer Academic Publishers, 1993.

[3] R. Schroeppel, H. Orman, S. O'Malley and O. Spatscheck, "Fast key exchange with elliptic curve systems", *Advances in Cyptology CRYPTO 95, LNCS 963*, pp. 43-56, 1995.

[4] C.C Chang, Y.T. Kuo, and C.H. Lin, "Fast algorithm for common-multiplicand multiplication and exponentiation by performing complements", *Proc. of the 17th Int. Conf. on Advanced Information Networking and Application (AINA'03)*, pp. 807-811, 2003.

[5] P. Balasubramaniam E. Karthikeyan, "Elliptic curve scalar multiplication algorithm using complementary recoding", *Elsevier*, pp. 51-56, 2007.

[6] 김창훈, 권순학, 홍춘표, 유기영, "타원곡선 암호 프로세서의 재구성형 하드웨어 구현을 위한 $GF(2^m)$ 상의 새로운 연산기", 정보과학회 논문지, 제31권 제8호, pp. 453-464, 2004.

[7] G. Orlando, C. Paar, "A Super-Serial Galois Fields Multiplier for FPGAs and its Application to Public-Key Algorithms", *Proceedings of 7th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 232-239, 1999.

[8] 최용제, 김호원, 김무섭, 박영수, "IC 카드를 위한 polynomial 기반의 타원곡선 암호시스템 연산기 설계", 2001년도 대한전자공학회 하계종합학술대회 논문집, 제24권 제1호, pp. 305-308, 2001