스마트카드의 암호화모듈 구현에 적합한 Digit-Serial 유한체 연산기 설계

하진석, 이광엽, *김원종, *장준영, *정교일, *배영환
서경대학교 컴퓨터공학과, *한국전자통신연구원
전화 : 02-940-7240

Design of Digit-serial Circuits for Cryptography Module on Smart cards

Jin-Suk Ha, Kwang-Youb Lee,
*Won-Jong Kim, *June-Young Chung, *Kyo-Il Chung, *Young-Hwan Bae
Dept. of Computer engineering, seokyeong University
*Electronics and Telecommunications Research Institute.
E-mail : kylee@skuniv.ac.kr

Abstract

In this paper, a digit-serial multiplier with a digit size of 32 is proposed, which has more advantages than the 193bit serial LFSR architecture. We give a design example for the irreducible trinomials $x^{193}+x^{15}+1$. In hardware implementations, it is often desirable to use the irreducible trinomial equations. The proposed multiplier is verified with a VHDL description using an elliptic curve addition. The measured results show that the proposed multiplier is 0.3 times smaller than the bit-serial LFSR multiplier.

I. 서론

최근 암호기술과 부호기술이 발달하면서 유한체 연산에 대한 요구가 크게 증가하고 있다. 유한체는 현대수학의 한 분야로 지난 50여년간 활발히 연구되어 왔으며, 주로 switching circuits, digital signal processing, 부호이론 특히, 최근에는 공개키 암호방식에 널리 이용되고 있다.[1]

유한체는 가감승제가 정의되는 유한개의 원소를 가지는 체(field)를 말하며, 가산과 승산은 교환법칙, 결합법칙, 분배법칙이 성립한다. 유한체는 Galois가 발견하여 흔히 Galois Field라 부르며 GF(p)로 표시한다. 여기서 p는 유한체의 원소 수를 말하며 보통 유한체의 크기(order)라고 한다.[2]

본 논문은 한국전자통신연구원과 시스템IC2010 지원을 받음.

본 논문에서는 유한체 연산기의 활용분야로 크게 발전하고 있는 암호분야에 적합한 유한체 연산기를 제안하고자 한다. 암호에서는 암호 공격으로부터 안전성을 높이기 위하여 수백비트의 긴 길의 암호키를 사용하기 때문에 유한체 연산기의 구조도 매우 복잡하다.

본 논문에서는 공개키 가운데 최근 스마트카드등에서 활용도가 높아가고 있는 타원곡선 공개키 암호시스템에 적용하기 적합한 유한체 연산기를 제안한다.

우선, 일반적인 유한체 연산기 구조를 살펴보고 새로운 구조를 제안한다.

II. 유한체의 정의

2,1 확대체의 활용

유한체는 기초체(ground field)와 확대체(extension field)로 나누어진다. 기초체 GF(p)의 p는 1보다 큰 소수로 GF(p)상의 원소는 {0,1,2 ...,P-1}이며, GF(p)를 m차 확대한 확대체 $GF(p^m)$은 GF(p)상의 m차 벡터 공

간으로 표시 가능하다.

본 논문에서는 디지털 회로에 응용이 용이한 GF(2^m)상의 연산회로를 채택하였다. . GF(2^m)상의 원소 수는 2^m개이며 각 원소는 m비트 binary로 표시할 수 있어 디지털 회로 응용에 적당하다.

2.2 Polynomial Basis 구조

원소의 개수가 2^m개인 유한체 GF(2^m)상에서 다항식 표현의 기본이 되는 basis는 NB(Normal Basis), PB(Polynomial Basis), 그리고 DB(dual Basis) 이렇게 크게 3가지로 나눌 수 있다. 표현방식을 보면 NB의 경우 GF(2^m)상에서 $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$이며 PB의 경우 GF($2^m$)상에서 $\{1, \alpha, \cdots, \alpha^{m-1}\}$이다. DB는 GF($2^m$)상에서 field generator P(x)의 근원인 α로 이루어진 PB $\{1, \alpha, \cdots, \alpha^{m-1}\}$를 바탕으로 $\{1, \alpha, \cdots, \alpha_{m-1}\}$로 표현된다.

위 형식에서 볼 때 NB에서 원소 a의 제곱근은 한번의 순회치환으로 얻어지므로 승산의 이점이 있다. 그러나 이 표현에 사용되는 함수 f에 의존한다는 단점이 있다. 또한 DB 표현도 게이트 수는 줄일 수 있으나 속도가 감소한다. 이 표현에 의한 하드웨어 구현시 DB로부터 PB로 변환하는 logic이 포함되어야 한다는 단점을 가진다. 이와 반대로 PB는 회로 구현시 구조가 간단하고 간단한 제어부를 요구한다. 또한 이 구조는 어떠한 field generator P(x)를 선택하더라도 동일한 구조를 사용할 수 있는 이점이 있다.

III. 일반적인 serial구조의 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.[7][8]

그림 1은 LSB(Least Significant Bit)부터 승산하는 회로이다. 그림 1의 승산기 회로에서 A와 Z는 모든 bit들이 병렬적으로 처리가 되므로 키 길이와 상관없이 $A \cdot B_i$ mod P는 한 싸이클에 수행이 된다. 그러므로 순차적으로 곱해지는 B의 bit들에 의해 전체 처리시간이 계산되어진다.

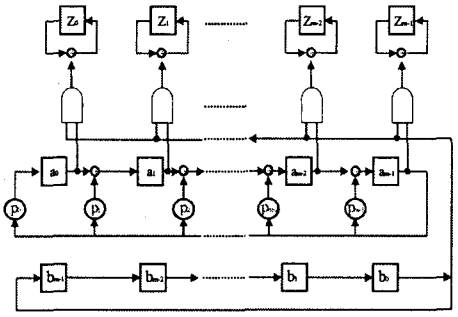

그림 1 순차회로를 이용한 승산기 I

이와같이 LFSR구조의 승산기는 multiplicand bit를 병렬로 연산하는 반면 multiplier는 bit단위의 직렬처리를 한다. 일반적인 LFSR구조의 승산기에서는 multiplicand가 m bits 병렬처리를 하기 때문에 3·m레지스터, m XOR 게이트, m AND 게이트가 필요하다.[3]

그러나 일반적인 LFSR구조의 승산기를 스마트카드와 같이 제한된 면적에서 활용하기 위해서는 보다 축소된 면적의 회로 설계가 요구된다.

Ⅳ 제안된 유한체 승산기의 구조

4.1. 메모리를 활용하는 Digit-serial 회로구조

LFSR구조 승산기에서 m bits길이의 연산 레지스터를 d bits 단위의 digit로 분할하면 레지스터의 수가 m/d 배만큼 축소가 된다.

본 논문에서는 그림 2에서와 같이 d=32bit인 digit-serial 구조의 승산기를 제안한다. 원시다항식 $p(x)=x^{193}+x^{15}+1$인 GF(2^{193})상에서 임의의 두 원소 A, B간의 승산을 목적으로 설계된 승산기 이다.

제안된 승산기는 그림 1의 승산기-I 구조를 바탕으로 하되 한 digit단위인 32bit길이의 레지스터들로 구성된다. 32bit로 분할된 A, B, Z레지스터 이외에 modular p(x)에 의하여 변형된 A레지스터의 32번 shift 결과가 저장되는 레지스터로 구성된다.

4.2. 제안된 구조에서 승산방법

제안된 승산기에서 193bit 승산방법을 설명하면, 그림 4 와 같이 초기상태에서는 multiplicand의 하위 32bit를 A 레지스터에 multiplier의 하위 32bit는 B 레지스터에 저장한다. A레지스터에 직렬로 연결되어 있는 A 레지스터 우측 32비트 레지스터에서는 A 레지스터로부터 shift right되어 빌려나온 bit들을 차례로 저장하게된다.

Multiplier 32bit는 multiplicand와 함께 bit-serial연산

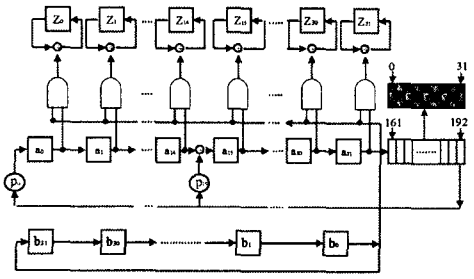

그림 2 m=193, d=32 bit digit-serial 승산기 구조

을 종료한 후, 다음 상위 32bit로 reload된다. 새롭게 reload된 multiplier는 32bit bit-serial연산을 수행한다. 32bit단위로 수행되는 bit-serial연산 결과는 Z레지스터에 누적(accumulation)되고 b_{193} bit의 연산이 완료될 때까지 반복되고 Z레지스터 값을 메모리에 저장한다. 이후 multiplicand의 다음 상위 32bit에 대한 연산을 위와 같은 방법으로 반복한다. 그림 3은 제안된 구조에서 m=193의 유한체 승산의 과정을 나타내고 있다.

A, B, Z레지스터 가운데 A 레지스터는 LFSR구조로 매 클럭마다 순환 shift right동작이 이루어진다. 그 결과 32 shift동작이 완료되면 좌측 digit값이 현재 digit의 32bit 레지스터와 교체된다. A 레지스터에 현재 digit가 저장된다면 좌측 digit를 저장할 별도의 32bit 레지스터가 필요하다. 그림 4에서 a_{192}-a_{161}이 저장되어 있는 레지스터가 좌측 digit레지스터(그림에서는 우측에 나타나있다)이다. A레지스터에는 좌측레지스터 값인 a_{192}-a_{161} 가 옮겨오고 A레지스터 값인 a_{31}-a_0는 좌측 digit레지스터로 옮겨진다. 이때 옮겨진 a_{31}-a_0 는 더 이상 초기값을 유지하지 못하고 modular p(x)에 의하여 변형된다. 따라서 변형된 값은 메모리에서 별도공간에 저장되어야 한다(A' 레지스터). 이 변형된 값은 두 번째 digit인 a_{63}-a_{32}에서 연산이 이루어질 때 좌측 digit의 역할을 한다.

위와 같은 방법으로 digit 승산이 이루어질 때 m=193인 경우 6 digit와 1 bit로 구성된다. 6 digit의 승산은 앞에서 설명한 바와 같이 32 shift를 6번 반복함으로써 수행이 완료된다. 그러나 남은 1 bit를 1 digit로 취급하여 처리하려면 32 cycle이 추가되기 때문에 마지막 digit에서는 남은 1 bit를 포함하여 33bit로 처리되도록 설계하였다.

Multiplier b_{192} bit도 남은 1 bit가 되며 그림 3의 맨 아래 블록에 표시된 것처럼 b_{191}-b_{160} 과 더불어 마지막 digit를 형성하여 33bit digit로 승산을 수행한다.

4.3. 주소발생장치 및 FSM설계

제어회로의 복잡도는 digit-serial 승산기의 장점을 상쇄할 수 있기 때문에 본 논문에서는 제어회로의 FSM(Finite State Machine)과 제어신호를 간략화할 수 있는 메모리 저장방법과 새로운 주소발생장치를 제안한다.

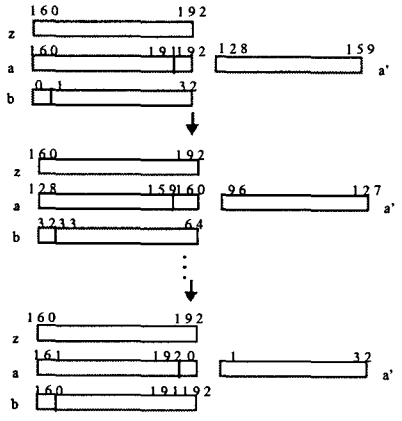

그림 3 제안된구조에서 유한체 승산방법

메모리에 저장되는 데이터는 승산입력데이터 A와 B, 승산결과값 Z, 그리고 modular reduction에 의하여 변형된 T의 영역으로 나누어진다. 그림 4는 메모리에서 A,B,Z,T영역의 배치순서이다. 본 논문에서는 승산기 FSM과 제어신호를 간략화하기 위한 방법으로 그림 4에서와같이 A영역과 T영역이 연속으로 이어지고 그림 5와 같은 구조의 계수기를 제안한다.

첫 번째 *digit*인 a_0-a_{31}의 승산에는 메모리 A1~6와 T7 영역에 저장되어 있는 digit가 사용된다. 두 번째 승산이 되는 digit는 a_{160}-a_{191} 으로 메모리 A2~6와 T7~8 영역의 digit와 함께 승산이 이루어진다. 그림 4에 따라 여섯 번째 *digit*까지 승산이 실행되며 이때 필요한 A의 영역과 T영역이 순서적으로 표시되어 있다. 그림 4 에서와같이 여섯 개의 digit를 승산하는데 필요한 A영역의 digit주소와 T영역의 digit주소는 7-mode counter로써 시작주소(start address)가 1 씩 증가하는 구조의 counter에서 발생된다.

그림 5과 같이 A, T레지스터의 load를 위한 address를 발생하는 counter를 AT counter라하고 그림 7의 주소발생블록도에 나타내었다. 그림 6에서 보듯이 A, T레지스터에 32bit multiplicand digit데이터를 load하기 위해서는 AT counter의 5bit출력값과 32bit start address의 상위 27bit를 연결하여 주소를 만들어낸다.

B레지스터에 32bit multiplier digit데이터를 load하는데는 B counter의 5bit가 같은 방법으로 사용된다. 승산 결과값 Z는 Z counter에 5bit를 더하여 만들고 이 주소값에 store한다.

제안된 AT counter의 장점은 주소발생회로에서 A counter와 T counter를 통합하여 회로 구현시 게이트수를 축소할 수 있고 승산기 제어 FSM에서 A counter와 T counter의 increment를 위한 제어신호수를 반으로 줄일 수 있다. AT counter를 사용함으로써 간략화된 승산기 제어FSM의 상태도를 그림 7에 나타내었다. AT counter를 적용하지 않으면 A counter와 T counter가 각각 독립적으로 존재하여 각각의 제어 상태가 발생하여 AT counter에 비하여 2 state가

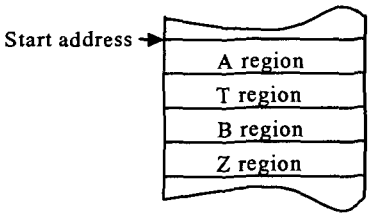

그림 4 메모리에서 데이터 영역

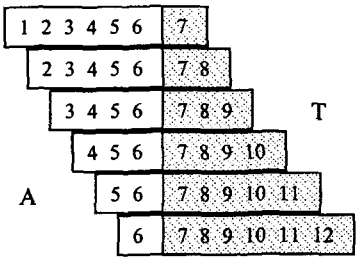

그림 5 AT counter의 상태변화도

늘어난다. 그림 7의 상태도에서 ATR은 AT counter가 지정하는 주소의 digit를 read하는 동작이며 BR은 B영역의 digit의 read동작이다. 그리고 ZW는 승산결과 digit의 저장동작을 TW는 modular reduction으로 변형된 A digit를 T영역에 저장하는 것을 의미한다. P는 reduced polynomial의 계수로 p=1인 부분에서는 A digit의 변형동작이 발생하기 때문에 변형된 값을 T영역에 저장하게되고 p=0인 부분에서는 A digit가 변형되지않고 원래값을 유지하면서 shift가 이루어지기 때문에 T영역에 저장하지 않고 승산이 진행된다.

V. 회로 검증 및 성능결과

설계된 회로는 Visual C++언어로 상위 모델링을 하고 오류를 검증한 후 VHDL모델링을 하였다. 회로 합성은 SYNOPSYS에서 수행하였고 회로 제작은 XILINX FPGA 로 구현되었다. 회로 합성 결과 및 구현 결과는 표 1에 수록하였다.

검증을 위한 데이터는 193비트로 랜덤함수에 의하여 발생하며 회로에서 출력된 곱셈 결과는 C++언어로 작성된 테스트프로그램과 비교하여 검증하였다.

표 1 VHDL과 SYNOPSYS로 구현된 결과

형태	게이트수	cycle 수
193bit LFSR 승산기	약 7000	210 cycle,6.93us@33MHz
Digit-Serial 승산기	약 2000	1200 cycle,39.6us@33MHz

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

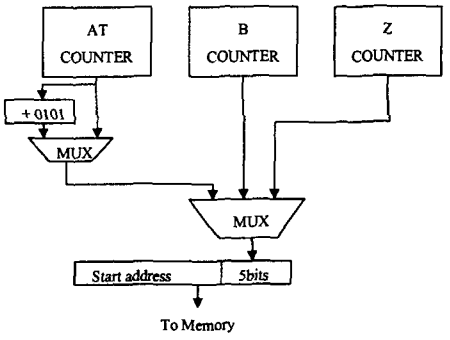

그림 6 주소발생장치의 블록도

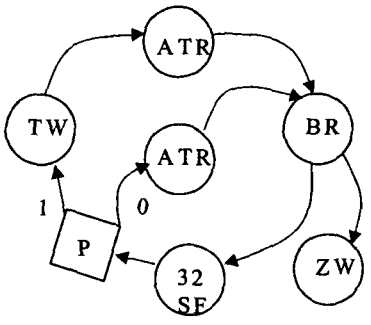

그림 7 승산기 제어FSM의 상태도

IV. 결 론

본 논문에서는 스마트카드내에 탑재되는 암호화프로세서 가운데 타원곡선(ECC)알고리즘 모듈을 구현하는데 필수적인 유한체 승산기를 설계하였다. 제안된 유한체 승산기는 기존의 bit-serial based LFSR구조를 개선한 digit-serial LFSR구조이다. 제안된 구조는 원시다항식이 $p(x)=x^{193}+x^{15}+1$인 다항식기저(polynomial basis) 유한체상에서 타원곡선알고리즘을 사용한 암호화 프로세서에 적용하였다. 검증결과 기존의 방법을 사용한 193bit LFSR 승산기에 비하여 게이트수가 3.5배 축소되었으며 스마트카드에 적용시 사용가능한 실행사이클을 보였다.

참고문헌

[1] 원동호,"유한체 GF(2^m)상의 연산회로 구성에 관한 연구", 성균관대학교 박사학위논문,1988

[2] B. Benjauthrit and I.S.Reed,"Galois switching function and their application," IEEE Trans. Comput., Vol.C-25, pp. 78-96, Jan. 1976.

[3] E.D. Mastrovito,"VLSI Architectures for c omputations in Galois Fields", Dept. of Electrical Eng. ,Linkoping Univ., Sweden, Ph.D. thesis,1991