Improved Massey-Omura Multiplier Design

*박혜영
삼성전자 SmartCard Design
e-mail : *hy1203.park@samsung.com*

*Hye-Young Park
Samsung Electronics, SmartCard Design

Abstract

This paper presents an effective multiplier in $GF(2^m)$ based on programmable cellular automata (PCA) and uses a normal basis. The proposed architecture has the advantage of high regularity and a reduced latency. The proposed architecture can be used in the effectual hardware design of exponentiation, division, inversion architectures.

Ⅰ. 서 론

최근 주목받고 있는 공개키 암호화 시스템은 대개 유한필드 $GF(p)$상이나 $GF(2^m)$상에서 구현되며[1], AB Multiplication 연산, A^2 Squaring 연산, A^m 지수 연산 등을 기본적으로 수행한다.

본 논문에서는 암호화 시스템을 효율적인 하드웨어로 구현하기 위하여 정규 기저(normal base) 방식을 적용한 곱셈기를 제안한다. 정규 기저 곱셈기중 가장 효율적인 Massey-Omura 곱셈기를 개선하여 PCA 구조로 구현함으로써 수행 시간을 줄이고 하드웨어로 구현된 면적을 줄이는 장점을 가지도록 한다.

논문의 구성은 다음과 같다. 2장에서 유한체에 대한 기본적인 개념과 특징, 3장에서는 제안된 곱셈기를 구현한 구조인 PCA 구조를 살펴보고, 4장에서는 개선된 Massey-omura 곱셈기를 제안한다. 마지막으로 5장에서 결론을 내린다.

Ⅱ. 유한체

유한필드 혹은 갈로아 체로 불리는 유한체는 교환, 결합, 분배 법칙에 닫혀 있고, 덧셈, 뺄셈, 곱셈, 나눗셈 연산이 가능한 유한한 원소를 가지는 집합이다. 유한체의 위수(order)는 유한체의 원소의 개수이고 유한체의 위수인 자연수 q는 소수(prime)이거나 소수의 지수 승이다. 유한체를 표기하는 다양한 방법이 있는데 위수가 소수 q와 양의 정수 n에 대하여 q^n이라고 하면 Fq^n 혹은 $GF(q^n)$로 나타낼 수 있고, q^n개의 원소를 가진 유한체는 본질적으로 하나뿐이다[2].

암호학에서는 일반적으로 두 종류의 유한체가 사용되는데, 소수 유한체(prime finite field)와 이진 유한체(binary finite field)가 그것이다. 본 논문에서는 이 두 가지 중에서 이진 유한체를 고려한다. 이진 유한체, $GF(2^m)$에서 원소들을 표기하기 위해서 다항식 기저 표기법(polynomial basis representation), 정규 기저 표기법(normal basis representation), 이원 기저 표기법(dual basis representation)등이 있다.

Ⅲ. PCA

CA는 규칙적으로 상호 연결된 많은 셀들로 구성되어 있는 유한 머신이다. 각각의 셀들은 적용된 법칙과 자신과 연결된 이웃의 현재 상태 값에 따라 새로운 값으로 갱

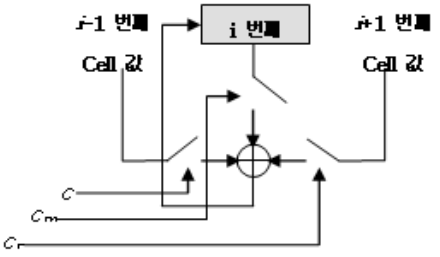

Figure 1. Programmable Cellular Automata

신된다. PCA는 셀마다 입력 값을 컨트롤하여 동일한 셀

에 클럭마다 다른 법칙을 적용 할 수 있는 CA를 말한다.

Ⅳ. 개선된 Massey-Omura 곱셈기

개선된 Massey-Omura 곱셈 알고리즘에서 사용된 곱셈기는 사용된 기약 다항식에 따라 다르게 구성되어 진다. 본 논문에서는 최적 정규 기저뿐만 아니라 일반적인 정규 기저에서도 적용 가능한 알고리즘을 제안한다.

알고리즘1. 개선된 Massey-Omura 곱셈기

Input : $A=\{a_0, a_1, ..., a_{m-2}, a_{m-1}\}$,

$B=\{b_0, b_1, ..., b_{m-2}, b_{m-1}\}$, $f(x)$

Output : $C=\{c_0, c_1, ..., c_m, c_{m-1}\}$

Step 1 : Set

$C=\{c_0, ..., c_{m-1}\}=\{0,...,0\}$

Step 2 : For k from 0 to m-1 do

Step 3 : $temp=A\times b_k$

Step 4 : $C=C+f(temp)$

Step 5 : $C=\sqrt{C}$

Step 6 : $A=\sqrt{A}$

위에서 입력 값은 $GF(2^m)$상의 임의의 원소 A, B를 LSB(Least Significant Bit) 형태로 표현하여 사용한다.

이제 이 알고리즘을 에 적용한 예를 살펴보도록 하자. 먼저 아래와 같은 다항식을 기약다항식을 선택한다. m이 4이므로 $m+1$은 소수(prime)가 되고, 2가 에서 primitive가 되어 Type Ⅰ의 최적 정규 기저상의 곱셈기를 구현 할 수 있다.

$P(x)=1+x+x^2+x^3+x^4$

$A=a_0\beta+a_1\beta^2+a_2\beta^{2^2}+a_3\beta^{2^3}$

$B=b_0\beta+b_1\beta^2+b_2\beta^{2^2}+b_3\beta^{2^3}$

입력 값 A, B를 위와 같이 나타내면 C의 각 항들의 계수는 다음과 같다.

$c_0=a_0b_2+a_1b_2+a_1b_3+a_2b_0+a_2b_1+a_3b_1+a_3b_3$

$c_1=a_0b_0+a_0b_2+a_1b_3+a_2b_0+a_2b_3+a_3b_1+a_3b_2$

$c_2=a_0b_2+a_0b_3+a_1b_1+a_1b_3+a_2b_0+a_3b_0+a_3b_1$

$c_3=a_0b_1+a_0b_2+a_1b_0+a_1b_3+a_2b_0+a_2b_2+a_3b_1$

이 결과 값을 바탕으로 $GF(2^4)$에서의 곱셈 알고리즘의 $f(x)$를 정하고 이를 토대로 PCA 구조의 곱셈기를 구현한다.

$c_0=a_2b_0+(a_2+a_3)b_1+(a_0+a_1)b_2+(a_1+a_3)b_3$위의 정리된 식을 PCA 구조를 이용하여 구현하면 Figure 2과 같다.

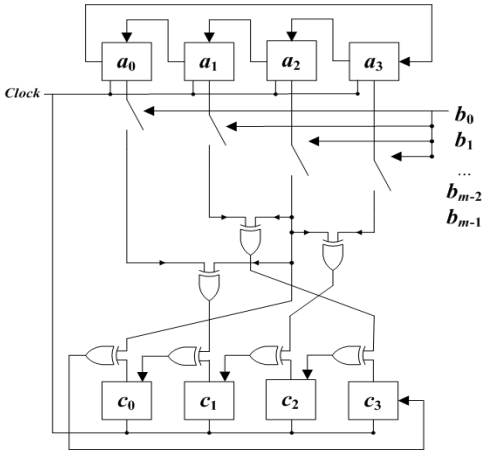

Figure 2. 개선된 Massey-Omura 곱셈기

Ⅴ. 결 론

8개의 레지스터와 7개의 XOR 게이트, 그리고 4개의 SWITCH로 구성되어진 곱셈기이며 Time delay이는 T_A+2T_X이다. 각 클럭에서 C의 값에 저장되는 값은 Time delay 만큼의 시간동안 늦추어진 뒤 저장된다.

제안한 곱셈기는 기존의 parallel 구조의 곱셈기보다 공간 복잡도면에서 뛰어나고, serial 구조의 곱셈기보다 시간 복잡도면에서 뛰어난 특성을 보이며 Diffie-Hellman Key Exchange Protocol이나 Elgamal Cryptosystem, 타원 암호 시스템(Elliptic Curve Cryptosystem, ECC)등의 공개키 암호화 시스템에서의 곱셈이나 나눗셈 연산을 위해 이용되어질 수 있다.

참고문헌

[1] E. R. Berlekamp, "Bit-serial Reed-Solomon encoders," IEEE Trans. IT-28, vol. 6, pp. 869-874, 1982.

[2] H. Wu and M. A. Hansan, "Low complexity bit-parallel multipliers for a class of finite fields," IEEE Trans. on Computers, vol. 47, no. 8, pp. 883-887, Aug. 1998.