한국산업정보학회논문지 제9권 제4호(2004. 12)

AB^2 연산을 위한 세미시스톨릭 구조 설계

(Design of Semi-Systolic Architecture for AB^2 Operation)

이 진 호* ,김 현 성*
(Jin-Ho Lee Hyun-Sung Kim)

요 약 본 논문에서는 GF(2^m)상에서 AB^2 연산을 위한 세미시스톨릭 구조를 제안한다. 먼저 기존의 세미시스톨릭 구조를 통하여 문제점을 제시하고, 이러한 문제점을 해결하기 위한 AOP (All One Polynomial)에 기반 한 새로운 AB^2 알고리즘을 제안하고 이를 위한 새로운 구조를 제안한다. 본 논문에서 제안한 구조는 기존의 구조들보다 효율적인 구성을 가진다. 제안된 구조는 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 수 있다.

핵심주제어 : 암호화 프로세서, 유한필드, AOP, 모듈러 곱셈기

Abstract This paper presents a new semi-systolic architecture for AB^2 operation. First of all, the previous architecture proposed by Lee et al. is analysed and then we present a new algorithm and it's architecture for AB^2 operation based on AOP (all one polynomial) to solve the shortcomings in the architecture. Proposed architecture has an efficient configuration than other previous architectures. It is useful for implementing the exponentiation architecture, which is the core operation in public-key cryptosystems.

Key Words : Crypto-processor, Finite fields, All one polynomial, Modular multiplier

I. 서 론

암호학(Cryptography), 디지털 신호 처리(Digital Signal Processing) 및 에러 교정 코드(Error-correcting Codes)의 응용에서 유한필드(Finite fields or Galois fields, GF) 연산은 아주 중요하다 [1-9]. 이러한 유한체 중에서 특별한 관심을 가지는 유한체는 GF(2^m)이다. 유한필드 GF(2^m)은 2^m개의 원소를 가지고 각각의 원소들은 0과 1의 비트-스트링으로 구성된다. 이러한 속성 때문에 갈로아 필드 연산의 하드웨어 구현에 유한필드 GF(2^m)이 적당하다[9].

여러 가지 구조를 기반으로 다양한 연산기가 제안되었다. 많은 연구에서 구조 복잡도를 줄이기 위해서 특별화 된 기약다항식을 이용한 곱셈기들이 제안되었다.

논문[10]에서 Jain등은 표준기저 상에서 모듈러 곱셈을 위한 세미시스톨릭 어레이 구조를 제안하였다. 그리고, 여러 가지 구조에 기반 한 모듈러 곱셈기들이 제안되었다. 그러나 이들 구조는 시스템의 복잡도 때문에 암호화시스템의 구성에는 효과적이지 못했다. 이러한 시스템의 복잡도를 줄이기 위해서 Itoh와 Tsujii는 AOP(All One Polynomial)에 기초한 곱셈기와 기약 다항식 ESP (Equally Spaced Polynomial)에 기초한 곱셈기를 설계하였다[11]. 그리고, Kim은 LFSR구조에 기반한 AOP를 이용한 다양한 연산기를 제안하였다

* 경일대학교 컴퓨터공학부
(Dept. of Computer Engineering, Kyungil University)

[12]. 또한, 논문 [13]에서 이등은 AB^2 연산을 위한 세미시스톨릭 어레이 구조를 AOP를 이용하여 제안하였다. 그러나 이등의 구조는 AOP의 속성을 최대한 활용하지 못하였다.

본 논문에서는 GF(2m)상에서 기약 다항식 AOP의 속성을 이용한 AB^2을 위한 병렬 세미시스톨릭 어레이 구조를 제안한다. 2-입력 XOR게이트의 딜레이(delay)를 D_{XOR2}라 할 때, 제안된 병렬 세미시스톨릭 어레이 구조의 전체 지연시간은 m+1을 갖고, 각 셀 당 D_{XOR2}의 임계경로를 갖는다. 본 논문에서 제안된 두 구조는 기존에 제안된 이등의 구조보다 AOP의 속성을 최대한 활용할 수 있었고, 더 간단하고 효율적인 구조복잡도를 가진다.

본 논문은 2장에서 본 연구와 관련된 기본 연구에 대해서 살펴본다. 특히 기존에 이등에 의해 제안된 세미시스톨릭 곱셈기를 분석한다. 3장에서는 2장에서 제시한 이등의 구조를 효율적으로 개선하기 위한 새로운 세미시스톨릭 곱셈기를 제시하고, 4장에서 기존의 구조와 본 논문에서 제안한 구조를 비교 및 분석을 한다. 그리고 마지막으로 5장에서 결론을 맺는다.

II. 관련연구

2.1 MSB 우선 지수 알고리즘

유한 체 (Finite Field)는 갈로아 체 (Galois Field, GF)라고도 불리며, 암호이론이나 부호이론에서 주로 사용되는 원소의 개수가 유한인 체를 말한다[9]. 다항식 $f(x)$의 근을 a라 하자. $GF(2^m)$상에서 $f(x)$를 $f(x)=f_m x^m+ f_{m-1}x^{m-1}+\cdots+f_1x+f_0$라 할 때, $f_i=1(i=0,1,\cdots,m)$인, 즉, 다항식의 항이 모두 1인 $f(x)$를 AOP (All One Polynomial)라고 한다. 다항식 AOP에서 m+1이 소수이고 2가 모듈러 m+1에 대해 원시 근이 되는 다항식을 기약 다항식이라 한다. 100보다 작은 m에 대해서 m이 2, 4, 10, 12, 18, 28, 36, 52, 58, 60, 66, 82 일 때 기약 다항식으로서의 AOP를 만족한다 [13]. 위의 AOP $f(x)$의 근 a에 의해 생성된 집합 { 1, a, ..., a^{m-2}, a^{m-1} }은 유한필드 $GF(2^m)$의 표준기저가 되고 유한필드 $GF(2^m)$상의 한 원소 a는 $a=a_{m-1}a^{m-1}+a_{m-2}a^{m-2}+\cdots+a_1a+a_0$로 표현된다.

기약다항식 AOP는 기저를 한 차원 확장했을 때 모듈러 (Modular)로서 효율적인 속성을 가진다. 표준기저에서 확장된 기저를 {1, a, ..., a^{m-2}, a^{m-1}, a^m }이라 하면, 확장된 기저 상에서 유한필드 $GF(2^m)$의 원소 A는 $A=A_ma^m+A_{m-1}a^{m-1}+A_{m-2}a^{m-2}+\cdots+A_1a+A_0$ (여기서, $A_m=0$, $A_i=a_i$, $0 \le i \le m-1$)로 표현된다. 여기서, $F(x)=x^m+ x^{m-1}+\cdots+x+1$를 m차의 기약 다항식 AOP라하고 a를 $F(x)$의 근이라 하자. 즉, $F(a)=a^m+a^{m-1}+\cdots+a+1=0$ 이다. 그러면 $F(a)=0$을 $a^m=a^{m-1}+\cdots+a+1$로 나타낼 수 있고 양변에 a를 곱하고 정리하면 다음 방정식을 만족한다.

$$a^{m+1}=1 \qquad (1)$$

대부분의 공개키 암호 시스템의 구현에 있어서 $GF(p)$나 $GF(2^m)$ 상에서 지수 연산이 필요하다[12]. 효율적인 지수 연산을 위해서는 지수의 처리 방식에 따라서 LSB (Least Significant Bit)와 MSB (Most Significant Bit) 우선 방식의 바이너리 메소드(Binary method)가 있다[12]. 본 논문에서는 MSB 방식의 알고리즘을 위한 기본 구조 제안에 그 목적이 있다. MSB 우선 알고리즘은 다음과 같다.

[알고리즘1] MSB 우선 지수 알고리즘
입력 : A, E, $f(x)$
출력 : $C=A^E \bmod f(x)$
단계1 : if (e_{m-1}==1) $C=A$ else $C=a^0$
단계2 : for $i=m-2$ to 0
단계3 : if (e_i==1) $C=AC^2 \bmod f(x)$
else $C=a^0C^2 \bmod f(x)$

알고리즘1의 단계3에서 $AC^2 \bmod f(x)$연산이 필요하다.

2.2 논문[13]의 세미시스톨릭 곱셈기

논문[13]에서 이등은 AB^2 연산을 위한 세미시스톨릭 어레이 구조를 AOP를 이용하여 제안하였다. 다음 알고리즘 2는 이등의 구조에서 사용된 알고리즘이다.

[알고리즘2] 논문[13]의 AB^2 곱셈 알고리즘
입력 : $A=(A_m,A_{m-1},\cdots,A_1,A_0),B=(B_m,B_{m-1},\cdots,B_1,B_0)$

출력 : $R=AB^2 \bmod P$
초기값 : $R^{m+1}=(R_m,R_{m-1},\cdots,R_1,R_0)=(0,0,...0,0)$
단계1 : for $i=m$ to 0
단계2 : for $j=m$ to 0
단계3 : $R^i_j=R^{i-1}_{<j-2>} + A_jB_i$

알고리즘2에서 모듈러 P는 식(1)에서 얻어진 AOP의 속성이 적용된 $P=d^{m+}+1$ 이다. 그리고 단계 3에서 $<x>$는 $x \bmod m+1$을 의미한다.

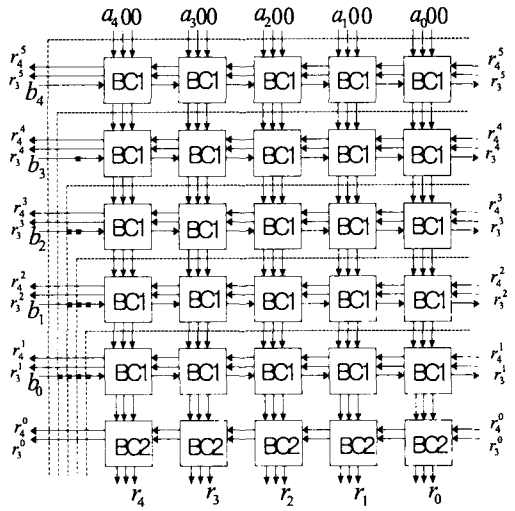
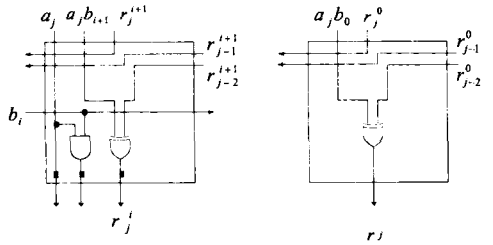


그림 1. 논문[13]의 세미시스톨릭 구조

이등은 알고리즘2를 기반으로 그림2의 기본구조를 갖는 그림 1의 세미시스톨릭 어레이 구조를 제안하였다.



(a) BC1 (b) BC2

그림 2. 그림1의 기본 구조

그림 1의 구조는 기존의 다른 구조에 비해서 상당히 시간 및 공간 복잡도를 향상시킬 수 있었다. 그러나 그림 2의 기본 구조들에서 본 바와 같이 데이터의 흐름처리를 위한 복잡한 선의 구성(wiring)을 갖는다.

III. 새로운 세미시스톨릭 AB^2 구조

본 장에서는 그림 1의 복잡한 선의 구성을 효율화 시키기 위한 새로운 세미시스톨릭 어레이 구조를 제안한다. 이러한 구조를 제안하기 위한 AOP를 이용한 모듈러 AB^2 알고리즘을 살펴보고, 이를위한 효율적인 선의 구성을 갖는 구조를 설계한다.

3.1 모듈러 AB^2 곱셈 알고리즘

본 절에서는 AOP의 속성이 적용된 제안된 AB^2 곱셈 알고리즘을 살펴본다. 이 알고리즘을 이해하기 위해서는 AOP 상에서 모듈러 제곱 연산에 대한 이해가 우선되어야 한다.

모듈러 제곱 연산, B^2은 모듈러 AOP의 속성에 의해서 계수의 재배치에 의해 다음과 같이 계산된다[12].

$$B^2=(B_md^m+B_{m-1}d^{m-1}+\cdots+B_1a+B_0)^2 \bmod (d^{m+1}+1) \quad (2)$$
$$=(B_md^{2m}+B_{m-1}d^{2m-2}+\cdots+B_1d^2+B_0) \bmod (d^{m+1}+1)$$
$$=B_{m/2}d^m+B_md^{m-1}+...+B_1d^2+B_{m/2+1}a+B_0$$

예를들어 GF(2^4)상의 한 원소 $B = B_4d^4+B_3d^3+B_2d^2+B_1a+B_0$의 B^2 연산은 다음과 같다(여기서, $a^{m+1}=1$의 AOP 속성이 모듈러로 이용된다).

$$d^5=1,\ d^6=a,\ d^7=d^2,\ d^8=d^3$$
$$B^2=(B_4d^4+B_3d^3+B_2d^2+B_1a+B_0)^2 \bmod (d^5+1)$$
$$=B_2d^4+B_4d^3+B_1d^2+B_3a+B_0$$

본 논문에서는 그림 1의 복잡한 선의 구성을 효율적으로 단순화하기 위해서 AOP의 모듈러 제곱 연산의 속성을 이용하여 알고리즘2를 알고리즘3과 같이 변형하였다.

[알고리즘3] 새로운 AB^2 곱셈 알고리즘
입력 : $A=(A_m,A_{m-1},\cdots,A_1,A_0),B=(B_m,B_{m-1},\cdots,B_1,B_0)$
출력 : $R=AB^2 \bmod P$
초기값 : $R^{m+1}=(R_m,R_{m-1},\cdots,R_1,R_0)=(0,0,...0,0)$

단계1 : $B'=Squaring(B)$
단계2 : for $i=m$ to 0
단계3 : for $j=m$ to 0
단계4 : $R^i_j=R^{i+1}_{<j-1>} + A_jB'_i$

알고리즘2와의 차이를 나타내기 위해서 알고리즘3에서는 추가되거나 변경된 내용은 굵은 글씨로 처리하였다. 알고리즘3의 단계1에서 $Squaring(B)$는 수식(2)에서 보여준 바와 같이, 계수의 재배치에 의한 모듈러 제곱 연산이다. 이렇게 알고리즘3은 모듈러 제곱 연산을 먼저 수행하고, 이 결과를 이용하여 AB^2 연산을 수행함으로서 단계4에서 $R^i_{<j-2>}$가 아닌 $R^i_{<j-1>}$이 사용된다.

3.2 세미시스톨릭 구조

본 절에서는 유한체 GF(2^4)에서의 원소 A, B를 이용한 AB^2 연산을 위한 세미시스톨릭 구조를 제안한다. 각 원소는 다음과 같이 표현된다.

$$A=A_4\alpha^4+A_3\alpha^3+A_2\alpha^2+A_1\alpha^1+A_0\alpha^0$$
$$B=B_4\alpha^4+B_3\alpha^3+B_2\alpha^2+B_1\alpha^1+B_0\alpha^0$$

구조 설계의 효율적인 이해를 위해서 알고리즘3의 처리과정을 살펴보면 다음과 같다. 먼저 단계1의 처리를 위해서 다음과 같은 연산을 수행한다.

$$B'=Squaring(B) \quad (3)$$
$$=B_2\alpha^4+B_4\alpha^3+B_1\alpha^2+B_3\alpha^1+B_0\alpha^0$$

즉, 알고리즘3의 단계4에서 사용되는 B'은 B를 입력받아 모듈러 제곱 연산을 수행한 후 $B'_4=B_2$, $B'_3=B_4$, $B'_2=B_1$, $B'_1=B_3$, $B'_0=B_0$ 로 초기화 된다. 모듈러 제곱을 위한 구조는 그림 3과 같다.

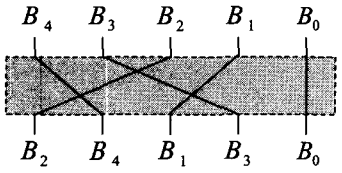


그림 3. 모듈러 제곱기

단계1을 수행한 후, 단계2부터 단계4까지의 연산을 통하여 다음 연산을 수행한다.

$$R^4=R^5_{Cir\_Left}+A_4(B'_4\alpha^4+B'_3\alpha^3+B'_2\alpha^2+B'_1\alpha^1+B'_0\alpha^0) \quad (4)$$
$$R^3=R^4_{Cir\_Left}+A_3(B'_4\alpha^4+B'_3\alpha^3+B'_2\alpha^2+B'_1\alpha^1+B'_0\alpha^0)$$
$$R^2=R^3_{Cir\_Left}+A_2(B'_4\alpha^4+B'_3\alpha^3+B'_2\alpha^2+B'_1\alpha^1+B'_0\alpha^0)$$

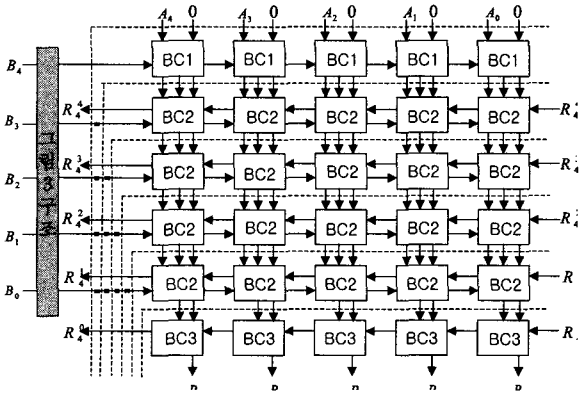


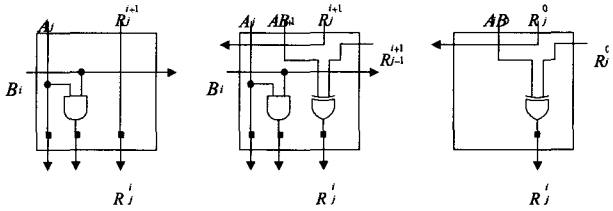
그림 4. 모듈러 AB^2 세미시스톨릭 곱셈기

$$R^1=R^2_{Cir\_Left}+A_1(B'_4\alpha^4+B'_3\alpha^3+B'_2\alpha^2+B'_1\alpha^1+B'_0\alpha^0)$$
$$R^0=R^1_{Cir\_Left}+A_0(B'_4\alpha^4+B'_3\alpha^3+B'_2\alpha^2+B'_1\alpha^1+B'_0\alpha^0)$$

여기서 $R^i_{Cir_Left}$는 R^i를 왼쪽으로 1비트 순환시프트(1-bit circular left shift)하는 연산이고, R^5는 0으로 초기화 된다. 연산이 수행된 후 AB^2 모듈러 곱셈 최종적인 결과는 R^0에 저장된다.

그림 4는 알고리즘3에 기반 한 본 논문에서 제안한 AB^2 연산을 위한 세미시스톨릭 어레이 구조를 보여준다. 그림 5는 그림 4를 위한 기본 구조들을 보여준다. 그림 5 (a)는 첫 번째 행을 위한 구조이고, 그림 5(c)는 마지막 행을 위한 구조이다. 그리고 나머지 구조들은 그림 5(b)의 구조를 이용한다. 그림 4의 구조는 GF(2^m)상에서 m=4에서 뿐만 아니라, 모든 m에 대해서도 확장 가능하다.

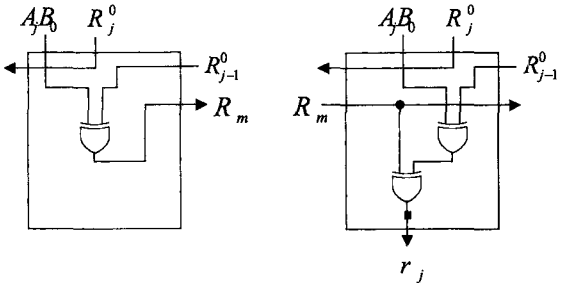
본 논문에서 제시한 구조도 논문[13]의 구조와 마찬가지로 그림4의 구조를 구현하기 위해서 표준기저 보다 확장된 기저 상에서 연산이 이루어졌다.



(a) BC1 (b) BC2 (c) BC3

그림 5. 그림4의 기본 구조들

따라서, 결과 값도 확장된 기저 상에서의 값이다. 이 문제를 해결하기 위해서는 그림4의 연산 후 그 결과를 다시 모듈러 감소 연산을 적용함 으로서 표준기저 상에서의 결과를 얻을 수 있다. 추가적인 모듈러 감소 연산을 위해서는 그림 4의 마지막 행의 구조인 그림 5의 (b) 구조를 다음 그림 6과 같이 변경한다. 그림 6 (a) 구조는 마지막 행의 첫 번째 열의 구조를 보여준다. 첫 번째 열을 제외한 나머지 열은 그림 6 (b)의 구조를 갖는다.



(a) BC4 (b) BC5

그림 6. 추가적인 모듈러 연산을 위한 구조

Ⅳ. 비교 및 분석

본 논문에서 제안한 구조는 Altera MAX+PLUSII를 이용하여 구조에 대한 검증을 수행하였다. 본 논문에서는 모듈러 AB^2 곱셈 연산을 위한 병렬 시스톨릭 어레이 구조를 제안하였다. 논문[10]과 [13]에서 모듈러 역원과 나눗셈을 위한 구조들이 각각 병렬 입출력 세미시스톨릭 구조로 제안되었다. 표 1은 본 논문에서 제안한 구조와 관련된 구조들의 성능 비교 및 분석을 보여준다.

표1에서 2-입력 AND와 XOR게이트의 지연시간(delay)는 각각 D_{AND2}와 D_{XOR2}이고, latch는 1bit이다. 먼저 논문 [10]에서는 각 셀 당 2개씩의 AND와 XOR 게이트가 필요하고, 지연시간으로 m+1, 임계경로는 D_{AND2}+D_{XOR2}속성을 갖는다. 그리고 논문 [13]에서는 모듈러로서 AOP의 속성을 사용하여 셀의 복잡도를 각 셀 당 1개씩의 AND와 XOR 게이트로 줄일 수 있었고, 지연시간으로 m+1, 임계경로는 D_{XOR2}속성을 갖는 구조를 구현하였다. 그러나 그림 1과 그림 2에서 보여준 바와 같이 복잡한 선의 구조를 가졌다.

본 논문에서 제안된 AB^2 곱셈기는 셀 복잡도, 지연시간과 임계경로면에서 기존의 구조보다 더 효율적이다. 따라서 본 논문에서 제안한 시스템을 기반으로 공개키 암호화 시스템의 기본 연산인 지수 연산을 수행한다면, 기존의 구조로 지수 연산을 수행하는 것 보다 더 좋은 결과를 얻을 수 있을 것이다.

표 1. 비트 순차 구조의 비교

구조 / 항목	논문 [10] 구조	논문 [13] 구조	제안한 구조
기약다항식	Generalized	AOP	AOP
셀수	m^2	$(m+1)(m+2)$	$(m+1)(m+2)$
셀복잡도	2-AND 2-XOR 3-latches	1-AND 1-XOR 3-latches	1-AND 1-XOR 3-latches
지연시간	m+1	m+1	m+1
선의구조	복잡	복잡	단순
임계경로	D_{AND2}+D_{XOR2}	D_{XOR2}	D_{XOR2}

Ⅴ. 결 론

본 논문에서는 GF(2^m)상에서 효율적인 지수연산을 수행하기 위한 병렬 세미시스톨릭 어레이 구조의 AB^2 곱셈기를 제안하였다. 논문 [13]에서 세미시스톨릭 어레이 구조에 기반한 AB^2 곱셈기를 제안하였다. 그러나 그 구조는 효율적인 구조 복잡도를 가지지만, 복잡한 선의구조를 가졌다. 본 논문에서는 이러한 기존의 구조의 문제점을 해결하기 위해서 효율적인 AOP 연산에 기반한 새로운 세미시스톨릭 어레이 구조를 제안하였다. 본 논문에서 제안한 구조는 기존의 구조와 동일한 지연시간과 임계경로를 갖지만, 선의구조 면에서 보다 효율적인 속성을 가졌다. 제안된 구조는 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본 구조로 사용될 수 있을 것이다.

참 고 문 헌

[1] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, Cambridge, MA: MIT Press, 1972.

[2] I. S. Reed and T. K. Truong," The use of finite fields to compute convolutions," *IEEE Trans. Inform. Theory*, vol. IT-21, pp.208-213, Mar. 1975.

[3] D. E. R. Denning, *Cryptography and data security*, Reading, MA: Addison-Wesley, 1983.

[4] A. M. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance," in*Adv. Cryptol., Proc. Eurocrypt*84, Paris, France, pp.224-314, Apr. 1984.

[5] W. Diffie and M. Hellman," New Directions in Cryptography," *IEEE Trans. on Info. Theory*, vol. 22, pp.644-654, 1976.

[6] E. R. Berlekamp, *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.

[7] A.J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Boston, MA: Kluwer Academic Publishers, 1993.

[8] R. Lidl, H. Niederreiter, and P. M. Cohn, *Finite Fields*(*Encyclopedia of Mathematics and Its Applications*), Cambridge University Press, 1997.

[9] D. E. Knuth, *The art of Computer Programing. Volume 2: Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 2nd edition, 1997.

[10] S. K. Jain and L. Song, "Efficient Semi-systolic Architectures for Finite Fields Arithmetic," *IEE Trans. on Computers*, vol. C-33, pp. 357-360, 1984.

[11] T. Itoh and S. Tsujii, "Structure of parallel multipliers for a class of fields GF(2^m)," *Info. Comp.*, vol. 83, pp. 21-40, 1989.

[12] H.S. Kim, *Bit-Serial AOP Arithmetic Architecture for Modular Exponentiation*, Ph.D. Thesis, Kyungpook National University, 2002.

[13] 이형목, 김현성, 전준철, 유기영, "GF(2^m)상에서 AB^2을 위한 세미시스톨릭 구조," 정보보호학회논문지, vol. 12, no.2 pp. 45-52, 2002.

이 진 호 (Jin-Ho Lee)

1974년 2월 영남대학교 공학사
1981년 2월 영남대학교 전자계산학과 공학석사
1996년 2월 영남대학교 전자계산학과 공학박사
1979년 3월 ~ 현재 경일대학교 컴퓨터공학과 교수
(관심분야 : 프로그래밍언어, 정보보호)

김 현 성 (Hyun-Sung Kim)

1996년 2월 경일대학교 컴퓨터공학과 공학사
1998년 2월 경북대학교 컴퓨터공학과 공학석사
2002년 2월 경북대학교 컴퓨터공학과 공학박사
2002년 3월 ~ 현재 경일대학교 컴퓨터공학과 교수
(관심분야 : 정보보호, 암호 프로토콜, 암호 프로세서 설계, IDS, PKI)