공개키 암호 시스템을 위한 LFSR 곱셈기 설계

(Design of LFSR Multipliers for Public-key Cryptosystem)

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

요 약 본 논문에서는 GF(2m)상에서 Linear Feedback Shift Register 구조기반의 새로운 구조를 제안한다. 먼저 모듈러 곱셈기와 제곱기를 제안하고, 이를 기반으로 곱셈과 제곱을 동시에 수행할 수 있는 구조를 설계한다. 제안된 구조는 기약다항식으로 모든 계수가 1인 속성의 All One Polynomial 을 이용한다. 제안된 구조는 구조복잡도면에서 기존의 구조들보다 훨씬 효율적이다. 제안된 곱셈기는 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 수 있다.

국문핵심어 : 암호화 프로세서, 유한필드, AOP, 모듈러 곱셈기

Abstract This paper presents new architectures based on the linear feedback shift register architecture over GF(2m). First, we design a modular multiplier and a modular squarer, then propose an architecture by combing the multiplier and the squarer. All architectures use an irreducible AOP (All One Polynomial) as a modulus, which has the properties of all coefficients with '1'. The proposed architectures have lower hardware complexity than previous architectures. They could be. Therefore 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

1. 서 론

유한필드(Finite fields or Galois fields, GF) 연산은 암호학(Cryptography), 디지털 신호 처리(Digital Signal Processing) 및 에러 교정 코드(Error-correcting Codes)의 응용에서 아주 중요하다 [1-9]. 특히, 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. 특히, 모듈러로서 AOP (All One Polynomial)의 특성을 이용한구조 복잡도 면에서 효율적이었다 [11-13]. Itoh와 Tsujii는 효율적인 구조 복잡도를 가진 기약 다항식 AOP에 기초한 곱셈기와 기약 다항식 ESP (Equally Spaced Polynomial)에 기초한 곱셈기를 설계하였다[11]. Fenn et al. 은 GF(2^m)상에서 LFSR (Linear Feedback Shift Register)구조를 이용하는 두 가지 형태의 *AB*곱셈기를 비트순차(Bit-serial) 구조로 설계하였다 [12]. Ha et al. 과 Yoo et al. 은 곱셈기/제곱기의 복합된 형태의 구조를 위한 알고리즘과 이를 위한 병렬 시스톨릭 구조를 제시하였다 [14-15]. 또한, Kim 은 표준기저 상에서 AOP를 이용한 여러가지 연산을 위한 LFSR 구조들을 제시하였다[13]. 지금까지의 연구에서 여러 구조들이 제안되었지만 시간과

* 경일대학교 컴퓨터공학부

공간 복잡도 면에서 보다 효율적인 구조 설계에 관한 꾸준한 연구가 필요하다.

본 논문에서는 GF(2^m)상에서 LFSR 구조기반의 새로운 구조들을 제안한다. 먼저 모듈러 곱셈기와 제곱기를 제안하고, 이를 기반으로 곱셈과 제곱을 동시에 수행할 수 있는 구조를 설계한다. 제안된 구조는 기약 다항식으로 모든 계수가 1인 속성의 AOP를 이용한다. 시뮬레이션 결과 제안된 구조가 구조복잡도면에서 기존의 구조들보다 훨씬 효율적임을 확인할 수 있다. 제안된 곱셈기는 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 유한 체에 대한 기본적인 정의와 기약 다항식으로서의 AOP 속성에 대하여 설명한다. 또한, 공개키 암호 시스템에 대해서 논하고 기본적인연산에 대해서 살펴 본다. 3장에서는 모듈러곱셈기와 제곱기를 설계하고, 곱셈/제곱을 동시에 수행하기 위한 구조를 설계한다. 그리고, 4장에서 시뮬레이션 결과를 제시하고 기존의 구조들과 비교 및 분석을 제시하고, 5장에서 결론을 맺는다.

2. 유한 체와 공개키 암호 시스템

유한 체 (Finite Field)는 갈로아 체 (Galois Field, GF)라고도 불리며, 암호이론이나 부호이론에서 주로 사용되는 원소의 개수가 유한인 체를 말한다[9]. 유한체는 0에 의한 나눗셈을 제외한 사칙연산에 대해서 닫혀있다. 유한필드 GF(2)의 유한 확대체를 GF(2^m)이라 하자. 먼저 유한 확대체 GF(2^m)상의 원소는 표준, 정규, 이원기저의 세 기저에 의해서 표현될 수 있다. 표준 기저를 제외한 다른 두 기저, 즉, 정규기저와 이원기저에서는 연산 전후에 기저의 변환이 필요하다. 본 논문에서는 원소표현에 있어서 연산 전후에 기저의 변환이 필요 없는 표준기저를 이용한다.

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 $m+1$이 소수이고 2가 모듈러 $m+1$에 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 AOP라 하고 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.

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

본 논문에서 제안된 구조들은 AOP를 모듈러로 사용하여 연산을 수행한다.

유한필드 상에서 Diffie-Hellman 키 교환 방식, 디지털 서명 알고리즘과 ElGamal 암호화 방식과 같이 잘 알려진 알고리즘을 응용한 타원 곡선 (Elliptic Curve) 기반의 공개키 암호 시스템의 구현에 있어서 GF(p)나 GF(2^m) 상에서 지수 연산이 필요하다[8]. 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[10]. 본 논문에서는 LSB 방식의 알고리즘을 위한 기본 구조 제안에 그 목적이 있다. LSB 우선 알고리즘은 다음과 같다.

[알고리즘1] LSB 우선 지수 알고리즘

입력 : A, E, $f(x)$
출력 : $C=A^E \bmod f(x)$
단계1 : $T=A$
단계2 : if (e_0==1) $C=T$ else $C=a^0$
단계3 : for i=1 to m-1
단계4 : $T=TT \bmod f(x)$
단계5 : if (e_i==1) $C=CT \bmod f(x)$

알고리즘1의 단계4와 단계5에서 제곱 연산과 곱셈 연산이 각각 필요하다. 즉, 지수연산을 위해서는 제곱기와 곱셈기를 각각 이용하거나, 제곱과 곱셈을 동시에 연산하는 구조 [14-15]를 이용하여 효율적인 연산

을 수행할 수 있다. 본 논문에서는 LSB 우선 지수 연산 알고리즘을 위한 기본 구조로서 AOP를 이용한 모듈러 제곱기와 곱셈기를 각각 제안하고, 두 구조가 결합된 구조를 제안한다.

3. 기본 연산기

본 장에서는 LFSR 구조에 기반 한 기존의 모듈러 곱셈기와 제곱기의 문제점을 분석하고 새로 제안한 곱셈기와 제곱기를 설계하고, 이들 구조를 결합한 구조를 설계한다.

3.1 모듈러 곱셈기

*AB*곱셈기 설계를 위한 GF(2^4)상에서의 곱셈 알고리즘은 그림 1과 같다[13]. 그림1(a)는 모듈러 *AB*곱셈 과정을 보여준다. 곱셈 후 연산의 결과에 모듈러 연산을 적용하면 그림1(b)와 같다. 즉, 곱셈 후 그림1(a)의 왼쪽에 강조된 부분의 값에 따라서 모듈러 감소 연산이 적용되어야 하는 부분이고, $d^{m+1}+1$이 기약다항식으로서 사용되면 모듈러 감소는 그림1(b)에서 보여주는 바와 같이 계산된다. 여기서, 모듈러 감소 연산은 그림1(a)에서 왼쪽 강조된 부분이 그림1(b)에서 보여준

A =				A_4	A_3	A_2	A_1	A_0
× B =				B_4	B_3	B_2	B_1	B_0
				A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0
			A_4B_1	A_3B_1	A_2B_1	A_1B_1	A_0B_1	
		A_4B_2	A_3B_2	A_2B_2	A_1B_2	A_0B_2		
	A_4B_3	A_3B_3	A_2B_3	A_1B_3	A_0B_3			
A_4B_4	A_3B_4	A_2B_4	A_1B_4	A_0B_4				
P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

(a) 모듈러 곱셈

d^4	d^3	d^2	d^1	d^0
A_4B_0	A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_3B_1	A_2B_1	A_1B_1	A_0B_1	A_4B_1
A_2B_2	A_1B_2	A_0B_2	A_4B_2	A_3B_2
A_1B_3	A_0B_3	A_4B_3	A_3B_3	A_2B_3
A_0B_4	A_4B_4	A_3B_4	A_2B_4	A_1B_4
P_4	P_3	P_2	P_1	P_0

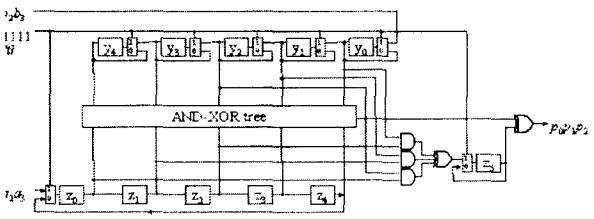
(b) 모듈러 감소 연산이 적용된 곱셈

<그림 1> 확장된 기저상의 곱셈 알고리즘

d^3	d^2	d^1	d^0
P_4	P_4	P_4	P_4
A_3B_0	A_2B_0	A_1B_0	A_0B_0
A_2B_1	A_1B_1	A_0B_1	A_4B_1
A_1B_2	A_0B_2	A_4B_2	A_3B_2
A_0B_3	A_4B_3	A_3B_3	A_2B_3
A_4B_4	A_3B_4	A_2B_4	A_1B_4
p_3	p_2	p_1	p_0

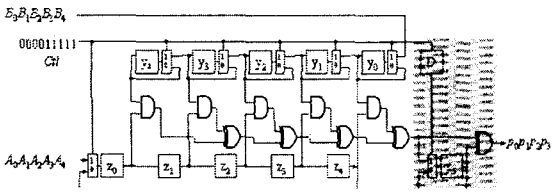
<그림 2> 표준기저상의 곱셈 알고리즘

것처럼 오른쪽으로 치환 됨으로 계산된다. 그러나 그림1(b)의 결과값 역시 확장된 기저상의 연산이므로, 한번의 추가적인모듈러 감소 연산이 필요하며 연산은 그림 2와 같이 수행된다.



<그림 3> Fenn등의 MAOPM

Fenn등의 연구에서는 AOPM에서 그림 3의 MAOPM으로 변환하는 과정에서 추가적인 한번의 모듈러 곱셈을 해결하기 위해서 공간 복잡도를 추가하고 시간 복잡도를 줄이는데 연구의 초점을 맞췄다. 그러나 본 연구에서는 반대로 시간복잡도의 추가로 효율적인 공간 복잡도를 갖는 모듈러 곱셈기를 제안한다. Fenn등의 구조는 필드의 크기에 의존적으로 공간 복잡도가 커지는 단점이 존재한다.



<그림 4> 제안한 곱셈기

그러나 본 논문에서 제안한 곱셈기는 필드의 크기

에 상관없이 기존의 구조에 비해서 단지 2개의 클럭 사이클이 추가로 필요하다. 그림4는 본 논문에서 제안한 모듈러 곱셈기를 보여준다. 그림4의 오른쪽 음영 부분이 그림 2의 알고리즘에서의 추가적인 모듈러 감소를 수행하기 위한 부분이다.

즉, 그림4의 구조는 입력의 마지막 스텝에 모든 레지스터 값이 초기화되고, 그 시점에 모듈러 감소를 위한 결과값 (P_4)이 계산되고 레지스터 Z_5에 저장된다. 이 값은 그 이후의 클럭 사이클의 결과값과 XOR 연산을 통하여 추가적인 모듈러 감소 연산을 수행하는데 이용된다. 곱셈기의 수행에 있어서 입력 상태와 처리 상태의 구별을 위해 하나의 제어 신호가 필요하다. 제어 신호는 입력을 위한 m+1비트의 '1'과 연산을 위한 m비트의 '0'을 필요로 한다. 즉, 본 논문에서 제안한 구조는 모듈러 곱셈 연산 수행을 위해서 AOPM과 같은 전체 $2m$+1개의 클럭사이클을 필요로 한다.

3.2 모듈러 제곱기

모듈러 제곱 연산, B^2은 모듈러 AOP의 속성에 의해서 계수의 재배치에 의해 다음과 같이 계산된다 [16].

$$B^2=(B_m a^m+B_{m-1}a^{m-1}+\cdots+B_1 a+B_0)^2 \mod (a^{m+1}+1) \quad (2)$$
$$=(B_m a^{2m}+B_{m-1}a^{2m-2}+\cdots+B_1 a^2+B_0) \mod (a^{m+1}+1)$$
$$=B_{m/2}a^m+B_m a^{m-1}+\ldots+B_1 a^2+B_{m/2+1}a+B_0$$

예를들어 GF(2^4)상의 한 원소 $B = B_4a^4+B_3a^3+ B_2a^2+B_1a+B_0$의 B^2 연산은 다음과 같다(여기서, $a^{m+1}=1$의 AOP 속성이 모듈러로 이용된다).

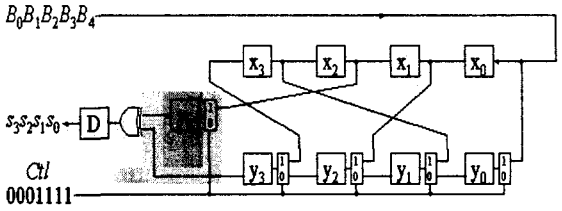
$$a^5=1,\ a^6=a,\ a^7=a^2,\ a^8=a^3$$
$$B^2=(B_4a^4+B_3a^3+B_2a^2+B_1a+B_0)^2 \mod (a^5+1)$$
$$=B_2a^4+B_4a^3+B_1a^2+B_3a+B_0$$

식 2의 B^2의 결과역시 확장된 기저상의 연산이므로, 한번의 추가적인 모듈러 감소 연산이 필요하며 연산은 다음과 그림 5와 같이 수행된다.

a^3	a^2	a^1	a^0
B_2	B_2	B_2	B_2
B_4	B_1	B_3	B_0
s_3	s_2	s_1	s_0

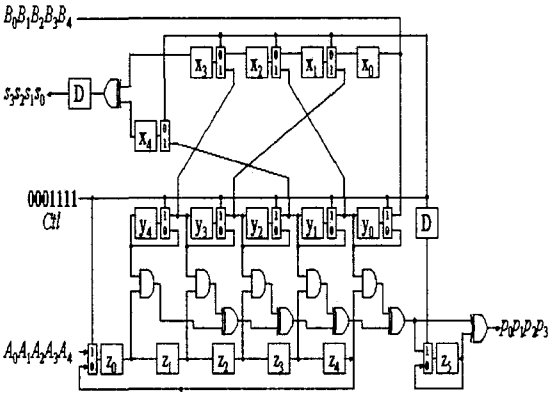
<그림 5> 표준기저상의 제곱 알고리즘

그림 6은 그림 5에서 제시한 알고리즘에 기반 한 GF(2^4)상의 비트순차 B^2연산기를 보여준다. 왼쪽의 음영 부분이 추가적인 모듈러감소 연산을 수행하는 부분이다.



<그림 6> 비트순차 제곱기

제곱기는 원래 $2m$-1 비트의 제어 신호가 필요하다 [14]. 그러나 본 논문에서는 곱셈기와 동기화를 위해서 $2m$+1 제어 신호를 위한 구조로 디자인 하였다.



<그림 7> 결합된 연산기

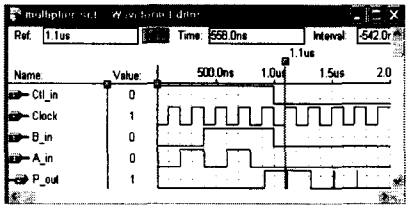
3.3 결합된 구조

LSB우선 지수연산을 위한 구조는 전술한 곱셈기와 제곱기를 결합함으로서 곱셈기와 제곱기를 각각 이용한 구조보다 효율적인 구조 복잡도를 갖는 연산기를 구성할 수 있다. 그림 7은 GF(2^4)상의 LSB 지수연산을 위한 비트순차 연산기를 보여준다.

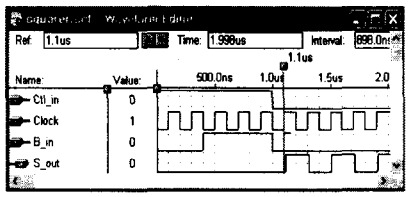
그림 7에서 y레지스터를 기준으로 윗부분은 그림 6에서 제시한 모듈러 제곱기이고, 아래부분은 그림 4에서 제시한 모듈러 곱셈기이다. 이 연산기도 다른 구조와 마찬가지로 수행을 위해서 $2m$+1 비트의 제어 신호가 필요하다.

4. 시뮬레이션 및 분석

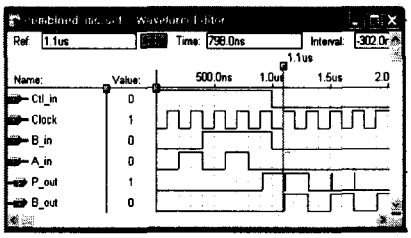
본 장에서는 지금까지 제안된 구조의 시뮬레이션 결과를 제시하고, 제안한 구조와 기존의 구조를 비교 분석한다. 본 논문에서 제안한 구조의 검증을 위해서는 Altera사의 MAX+PLUS를 이용하여 시뮬레이션 하였다. 그림 8은 제안한 각 구조의 시뮬레이션 결과를 보여준다.



(a) 모듈러 곱셈기 시뮬레이션 결과



(b) 모듈러 제곱기 시뮬레이션 결과



(c) 결합된 연산기 시뮬레이션 결과

<그림 8> GF(2^4)상에서의 시뮬레이션 결과

표 1은 본 논문에서 제안한 구조와 기존의 구조에 대한 비교를 자세히 보여준다. 논문 [15]에서는 일반화된 기약 다항식을 이용한 시스톨릭 어레이에 기반한 곱셈/제곱 연산을 동시에 수행하는 구조를 제안하였다. 또한, Kim은 논문 [13]에서 Fenn등의 구조인 MAOPM에 기반 한 곱셈 및 제곱을 동시에 수행하는 구조를 제안하였다. 논문 [15]의 구조와 본 논문에서 제안된 구조를 비교하면 본 논문에서 제안한 구조가 기존의 구조에 비해서 효율적인 구조 복잡도를 가지나 임계경로 면에서는 구조의 특성상 조금 성능이 떨어짐을 확인 할 수 있다. 그러나 시간/공간 복잡도를 계산하면 제안한 구조가 기존의 [15] 구조에 비해서 보다 효율적임을 확인할 수 있다. 또한, Kim의 MFSM구조와 제안된 구조를 비교해보면 제안한 구조는 필드의 크기에 상관없이 단지 2개의 추가적인 클럭 사이클을 필요로 하지만 구조적 복잡도 면에서 현저한 장점을 보임을 확인 할 수 있다.

5. 결 론

본 논문에서는 GF(2^m)상에서 효율적인 지수연산을 수행하기 위한 여러가지 기본 구조를 제안하였다. 먼저 LFSR 구조기반의 새로운 모듈러 곱셈기와 제곱기를 제안하였으며, 이 구조들을 기반으로 결합된 구조를 제안하였다. 결합된 구조는 한번에 모듈러 곱셈 결과와 제곱 결과를 동시에 계산할 수 있는 구조이다. 제안된 구조는 필드의 크기에 상관없이 단지 2개의 추가적인 클럭 사이클이 필요하지만 구조복잡도면에서는 기존의 구조들보다 훨씬 효율적임을 확인할 수 있었다. 제안된 구조는 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 수 있을 것이다.

<표 1> 비트 순차 구조의 비교

항목 \ 구조		논문 [13] 구조	논문 [15] 구조	제안한 구조
기본구조		LFSR	Systolic array	LFSR
기약다항식		AOP	Generalized	AOP
레지스터	REG/Latch	$3(m+1)$	$15m$	$3m+6$
AND	AND	$2m-1$	$3m$	$m+1$
XOR	XOR	$2m-1$	$3m$	$m+2$
MUX	MUX	$2m+2$	$3m$	$2m+3$
Latency	임계경로	1AND+($\log_2$m)XOR	1AND+1XOR	1AND+($\log_2$m)XOR
	Latency	$2m-1$	$3m$	$2m+1$

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. Eurocrypt84*, 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] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Comm. ACM.* vol. 21, pp. 120-126, 1978.

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

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

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

[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] S.T.J. Fenn, M.G. Parker, M. Benaissa, and D. Tayler, "Bit-serial multiplication in GF(2^m) using irreducible all-one opolynomial," *IEE Proc. Comput. Digit. Tech.*, vol. 144, no.6 pp. 391-393, 1997.

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

[14] J.C. Ha and S.J. Moon, "A Common-multiplicand Method to the Montgomery Algorithm for Speeding Up Exponentiation", *Information Processing Letters*, vol. 66, no. 2, pp. 105-107, 1998.

[15] 유기영, 김정준, "유한 필드 GF(2^m)상의 시스톨릭 곱셈기/제곱기 설계", 정보과학회논문지:시스템 및 이론 제28권, 제 5호, pp. 289-300, 2001.

이 진 호 (Jin-Ho Lee)

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

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 : 정보보안, 암호 알고리즘, 암호 프로세서 설계, IDS, PKI)