분할된 AB^2 시스톨릭 모듈러 곱셈기 설계

정회원 이 진 호*, 김 현 성**

Design of Partitioned AB^2 Systolic Modular Multiplier

Jin Ho Lee*, Hyun Sung Kim** *Regular Members*

요 약

AB^2연산은 공개키 암호화 시스템을 위한 효율적인 기본 연산으로 알려져 있고 이를 위한 다양한 하드웨어가 설계되었다. 그러나 이들 구조들은 암호학적 응용에 사용되기에는 구조복잡도가 크다는 문제점이 있었다. 본 논문에서는 GF(2^m)상에서 공간 효율적인 분할된 AB^2 시스톨릭 모듈러 곱셈기를 설계한다. MSB AB^2 모듈러 곱셈 알고리즘으로부터 데이터 의존 그래프를 유도하고 유도된 의존 그래프를 1/3로 분할함으로서 공간 효율적인 분할된 AB^2 시스톨릭 곱셈기를 설계한다. 본 논문에서 제안한 곱셈기는 기존의 곱셈기와 비교하여 2/3정도의 구조 복잡도를 줄일 수 있다. 본 논문에서 제안한 구조는 크기에 제한을 갖는 스마트 카드 등에서 사용될 공개키 암호의 핵심이 되는 지수기의 구현을 위한 효율적인 기본구조로 사용될 수 있을 것이다.

Key Words : cryptoprocessor, finite fields, modular multiplier, public key cryptosystem

ABSTRACT

An AB^2 modular operation is an efficient basic operation for the public key cryptosystems and various systolic architectures for AB^2 modular operation have been proposed. However, these architectures have a shortcoming for cryptographic applications due to their high area complexity. Accordingly, this paper presents an partitioned AB^2 systolic modular multiplier over GF(2^m). A dependency graph from the MSB AB^2 modular multiplication algorithm is partitioned into 1/3 to get an partitioned AB^2 systolic multiplier. The multiplier reduces the area complexity about 2/3 compared with the previous multiplier. The multiplier could be used as a basic building block to implement the modular exponentiation for the public key cryptosystems based on smartcard which has a restricted hardware requirements.

Ⅰ. 서 론

정보화 사회에서 데이터 통신이 급격히 증가하고 있다. 특히, 무선 통신의 보급은 제한된 조건에서 정보 보호의 필요성과 그 중요성은 아주 큰 문제로 대두되고 있다. 암호학(cryptography), 디지털 신호처리(digital signal processing) 및 에러 교정 코드(error-correcting codes)의 응용에서 유한필드(Finite fields or Galois fields, GF) 연산은 아주 중요하다[1-4]. Diffie-Hellman 키교환 프로토콜과 ElGamal과 같은 대부분의 공개키 암호화 시스템에서는 유한필드 상의 모듈러 지수승(modular exponetiation)을 기본연산으로 하고 있다[5-8]. 모듈러 지수승기는 모듈러 AB^2 곱셈(modular multiplication) 연산기를 기본 구조로서 사용한다. 또한, 타원 곡선 암호화 시스템에서는 정수배의 곱셈 연산을 기본으로 하고 있다[8]. 현재 대부분의 유한필드 상의 공개키 암호 알고리즘이 스마트카드에 장착될 때, 스마트카드에 장착된 CPU만을 이용한 구현이 어려워 대부분 별도의 특수프로세서를 장착하여 이용되고 있다. 이렇

* 경일대학교 컴퓨터공학부 (jhlee@kiu.ac.kr),
** 경일대학교 컴퓨터공학부 (kim@kiu.ac.kr)
논문번호 : KICS2005-08-349, 접수일자 : 2005년 8월 29일

게 칩으로 구현 시 칩에 들어가는 게이트의 수가 많아서 면적이 커지는 단점이 있다.

이러한 문제를 해결하기 위해 다양한 암호화 알고리즘의 개발과 효율적인 구현에 대한 연구가 계속 진행되고 있다[8-12]. 지금까지 모듈러 AB^2 곱셈 연산을 위해 개발된 연구 결과들은 다음과 같다. 먼저 Wei는 유한필드상에서 AB^2+C를 계산하기 위한 병렬 시스톨릭 구조를 제안하였다[9]. 그리고 이 구조를 이용하여 역원과 나눗셈 연산을 위한 구조들을 제안하였다[10]. Wang은 Wei의 시스템에 존재하는 양방향 데이터 흐름을 해결하기 위한 단방향 구조를 갖는 시스톨릭 구조를 제안하였다[11]. Kim등은 Wei와 Wang 구조의 구조 복잡도와 시간 복잡도를 향상시키기 위하여 병렬 시스톨릭 구조와 선형 시스톨릭 구조를 각각 제안하였다[12]. 그러나 기존의 구조들은 복잡한 구조 복잡도로 인하여 스마트카드와 같은 하드웨어 제약을 갖는 시스템에 활용되기엔 어려움이 있다. 그래서 제한된 하드웨어 크기에 맞춰서 시스톨릭 어레이의 크기를 줄일 수 있는 분할된 선형 시스톨릭 어레이로 설계할 필요성이 있다.

본 논문에서는 유한필드 상에서 모듈러 곱셈 알고리즘을 위한 효율적인 구조 복잡도를 갖는 분할된 AB^2 선형 시스톨릭 모듈러 곱셈기를 설계한다. 먼저, AB^2 모듈러 곱셈 알고리즘으로부터 데이터 종속 그래프를 유도한다. 유도된 데이터 종속 그래프로부터 1/3 크기로 분할된 선형 시스톨릭 어레이를 설계한다. 제안된 시스톨릭 어레이 VHDL로 프로그램하고, 이를 ALTERA MAX+PLUS II 시뮬레이션 툴을 이용하여 검증한다. 이렇게 분할된 선형 시스톨릭 곱셈기를 설계함으로서 하드웨어의 제약을 해결할 수 있는 새로운 시스톨릭 곱셈기를 유도할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 Kim 등이 제안한 모듈러 곱셈기를 분석한다. 3장에서 분할된 AB^2 선형 시스톨릭 모듈러 곱셈기를 설계하고 4장에서는 설계된 시스톨릭 어레이와 기존의 곱셈기를 비교 분석한다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. Kim등의 AB^2 모듈러 곱셈기

유한체 상에서 Diffie-Hellman 키 교환 방식, 디지털 서명 알고리즘과 ElGamal 암호화 방식과 같은 공개키 암호화시스템(cryptosystem)의 구현에 있어서 GF(p)나 GF(2^m) 상에서 효율적인 지수 연산이 필요하다. 이러한 지수 연산은 모듈러 AB^2 곱셈을 반복함으로서 수행될 수 있다. 유한체 GF(2)의 유한 확대체를 GF(2^m)이라 하자[6,8]. 먼저 유한 확대체 GF(2^m)상의 원소는 다항식, 정규, 이원기저의 세 가지 기저에 의해 표현된다. 본 논문에서는 기저의 변환이 필요 없는 다항식기저에 초점을 맞추었다. 다항식기저 {1, a, a^2,…, a^{m-1}}에서 유한체 GF(2^m) 상의 임의의 원소 A를 나타내면 $A=a_{m-1}a^{m-1}+a_{m-2}a^{m-2}+\cdots+a_1a+a_0$로 나타낼 수 있다. GF(2)상에서 차수 m의 기약 다항식(irreducible polynomial) F는 필드 GF(2^m)을 구성하는데 필요하다. 유한 필드 GF(2^m)의 2^m개 다항식은 GF(2)의 원소를 계수로 가지는 모든 다항식들을 기약 다항식 $F=f_ma^m+f_{m-1}a^{m-1}+f_{m-2}a^{m-2}+\cdots+f_1a+1$로 모듈러 연산을 행한 결과이다. 유한 필드 GF(2^m)에서 비트 문자열 $A=(a_{m-1}\ a_{m-2}\ \ldots\ a_1\ a_0)$는 다항식 $A=a_{m-1}a^{m-1}+a_{m-2}a^{m-2}+\cdots+a_1a+a_0$에 일대일 대응된다. 앞으로 필드 원소 A의 비트 문자열 표현과 다항식 표현은 동치로 간주한다.

Kim등은 논문[12]에서 다음 수식(1)과 같은 AB^2 모듈러 곱셈 알고리즘을 제안하였다.

$$
\begin{aligned}
P=AB^2 \bmod F \qquad (1)\\
=A(b_{m-1}a^{m-1}+b_{m-2}a^{m-2}+\cdots+b_1a+b_0)^2 \bmod F\\
=(Ab_{m-1}a^{m-1}+Ab_{m-2}a^{m-2}+\cdots+Ab_1a+Ab_0)^2 \bmod F\\
=(\ldots(\ldots((Ab_{m-1})a^2 \bmod F+Ab_{m-2})a^2 \bmod F+\cdots\\
\quad +Ab_1)a^2 \bmod F+Ab_0
\end{aligned}
$$

그림 1은 Kim등의 AB^2 모듈러 곱셈 알고리즘에 기반 한 GF(2^4)상의 병렬 시스톨릭 곱셈기를 보여준다.

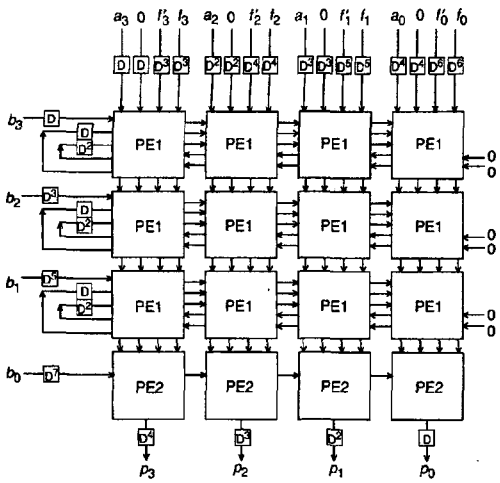


그림 1. Kim등의 병렬 시스톨릭 모듈러 곱셈기

또한, Kim등은 그림 1로부터 수평방향으로 cut-set 시스톨릭 과정을 통해 그림 2와 같은 GF(2^4)상의 선형 시스톨릭 곱셈기를 제시하였다.

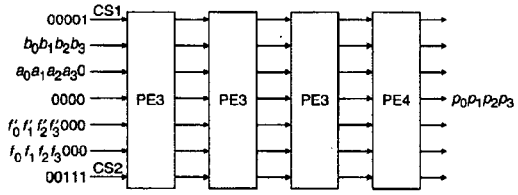
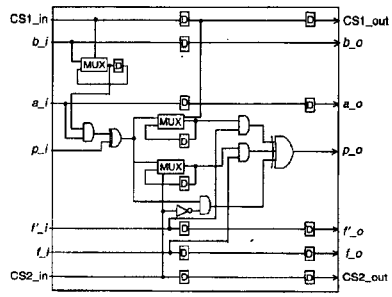
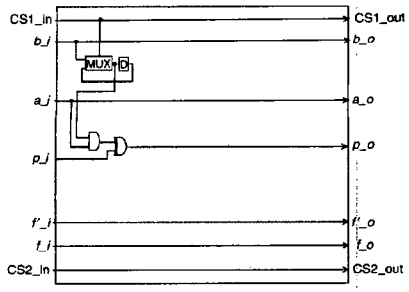


그림 2. Kim등의 선형 시스톨릭 모듈러 곱셈기



(a) PE3 구조



(b) PE4 구조

그림 3. 그림 2의 기본구조

그림 2의 구조는 그림 3 (a)와 (b)의 구조를 기본 구조로 갖는다. 그림 2의 가장왼쪽 PE3은 그림 1의 제일 왼쪽 열의 일을 수행하고, 가장오른쪽 PE4는 그림 1의 제일 오른쪽 열의 일을 수행한다. 기존의 AB^2 모듈러 곱셈 알고리즘과는 달리 Kim등이 제안한 곱셈 알고리즘은 최종적인 모듈러 연산의 복잡도를 단순화시킴으로서 기존 모듈러 곱셈의 구조적/시간적 복잡도를 향상 시킬 수 있었고, 보다 효율적인 구조 구성을 통해서 선형 시스톨릭 곱셈기로 유도할 수 있었다. 그러나 Kim등의 구조 역시 복잡한 구조 복잡도로 인하여 스마트카드와 같은 하드웨어 제약을 갖는 시스템에 활용되기엔 어려움이 있다. 다음 장에서는 이러한 문제를 해결하기 위하여 Kim등의 구조에 기반 한 분할된 선형 시스톨릭 어레이를 설계한다[13].

Ⅲ. 분할된 AB^2 선형 모듈러 곱셈기

본 장에서는 Kim등에 의해서 제시된 새로운 AB^2 모듈러 곱셈 알고리즘으로부터 데이터 의존 그래프(DG, dependency graph)를 유도하고 이러한 데이터 의존 그래프의 처리과정을 효율적으로 재 스케줄링 하여 구조 복잡도가 개선된 분할된 AB^2 시스톨릭 모듈러 곱셈기를 설계한다.

그림 4는 Kim등의 알고리즘으로부터 데이터 의존성을 고려한 각 노드의 수행시간이 스케줄링 된 GF(2^6)상의 데이터 의존 그래프를 보여준다. 이 그래프에 cut-set 시스톨릭 과정[13]을 적용하면 그림 1의 병렬 시스톨릭 곱셈기를 유도할 수 있다.

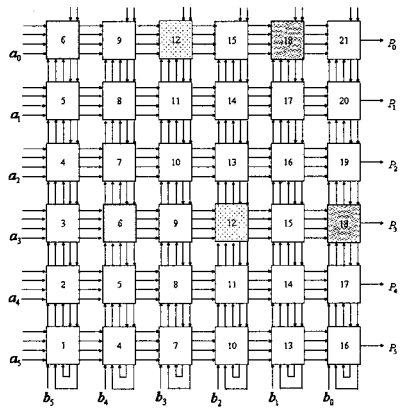


그림 4. 스케줄링이 적용된 DG

그림 4의 데이터 의존 그래프는 그림 1의 병렬 시스톨릭 곱셈기의 시간별 작업을 보다 효율적으로 보이기 위해서 알고리즘의 열 인덱스를 변환한 그래프이다. 그림 5는 그림 4의 이해를 위해서 유도된 GF(2^6)상의 간략화 된 선형 시스톨릭 곱셈기이다.

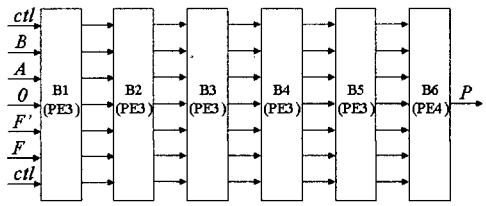


그림 5. GF(2^6)상의 선형 시스톨릭 곱셈기

그림 5의 B1에서 B5까지는 기본구조로 그림 3 (a)의 PE3 구조를 갖는다. 그러나 B6는 그림 3 (b)의 PE4 구조를 갖는다. 그림 5의 각각의 기본구조는 그림 4의 각각의 한 열들의 모든 작업을 수행한다. 즉, B1은 제일 왼쪽의 한 열들의 작업을 시스템 클럭 1부터 6까지 수행하고, B2는 왼쪽에서 두 번째 열들의 작업을 시스템 클럭 4부터 9까지 수행하고, 마지막 B6는 제일 오른쪽 열들의 작업을 시스템 클럭 16부터 21까지 수행한다. 이러한 작업을 수행하는데 있어서 B1은 시스템 클럭 7부터 21까지는 아무런 작업도 하지 않고 유휴상태에 놓임을 확인할 수 있다. 이러한 유휴상태에 놓인 기본구조들을 최대한 활용할 수 있도록 재 스케줄링 할 수 있다면 시스템의 복잡도를 효율적으로 개선할 수 있을 것이다.

그림 6에 제시된 데이터 의존 그래프는 그림 4의 유휴 기본구조를 최대한 활용하기위해서 노드를 1/3 크기로 분할시 새롭게 수행시간이 재 스케줄 된 데이터 의존 그래프를 보여준다. 각각의 기본구조들은 Kim등의 논문에서 제시된 것처럼 그림 3 (a) PE3의 구조를 갖는다. 그림 6의 각각의 기본구조는 프로세서 수행 시 3번의 연산을 수행한다. 특히, 이렇게 데이터 의존 그래프의 재 스케줄링을 위해서는 입력 데이터의 재 스케줄링 또한 고려되어야 한다. 먼저 각 행에 입력되는 데이터 A는 동일한 행의 모든 스케줄링에 동일한 값의 입력이 고려되어야 한다. 그러나, 각 열에 입력되는 B와 각 행의 중간 결과 값인 R은 동일한 기본구조가 이용되더라도 스케줄링 시점에 따라서 다른 값이 사용되어야 한다.

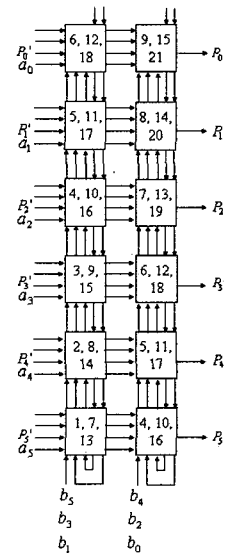
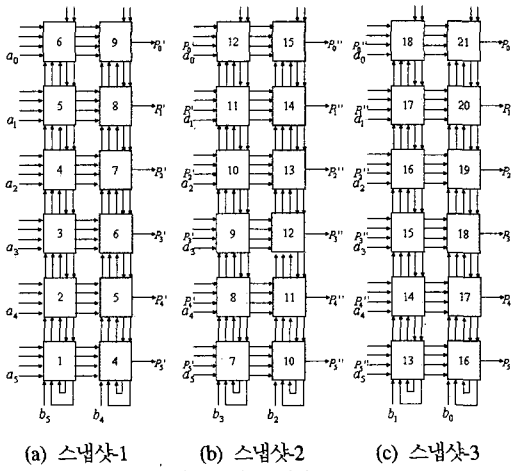


그림 6. 재 스케줄링 된 DG

또 하나 고려해야 할 사항은 이렇게 재 스케줄링 된 데이터 의존 그래프에서 오른쪽 행에 위치하는 각각의 기본구조는 그림 3 (a)의 PE3 구조를 갖는다는 것이다. 그러나 그림 4의 오른쪽 제일 마지막 열에서의 연산은 그림 3 (b)의 PE4 구조의 연산을 수행하여야 한다. 그러나 PE3 구조대신 PE4 구조를 사용하는 것은 결과값에 아무런 영향을 미치지 않는다. PE3과 PE4 연산의 차이는 수식 (1)로부터 명확하게 확인할 수 있다. PE3는 의 수식 (1)의 일반화된 항인 $(Ab_i)d^2 \bmod F$ 연산을 수행하기 위한 구조인 반면 PE4는 수식 (1)의 마지막 항인 $\bmod F$ 연산이 배제된 Ab_0 곱셈 연산만을 수행한다. 즉, Kim등의 알고리즘의 특성상 수식 (1)의 마지막 항의 연산에 있어서는 모듈러 연산이 필요 없다. 그러므로, 모듈러 연산의 특성상 모듈러 연산이 필요하지 않는 값에 모듈러 연산을 취하더라도 결과 값에는 영향을 미치지 않는 속성을 확인할 수 있다. 그러므로, 본 논문에서 제안하는 효율적인 구조복잡도를 갖는 분할된 선형 시스톨릭 곱셈기에서는 모든 기본구조가 동일한 구조를 갖는다.

그림 7은 입력과 처리가 재 스케줄링된 데이터 의존 그래프의 스냅샷(Snapshot)을 보여준다. 이렇게 재 스케줄링된 데이터 의존 그래프로부터 시스톨릭화 과정을 통해서 병렬 시스톨릭 곱셈기와 선형 시스톨릭 곱셈기를 각각 유도할 수 있다. 그러나 본 논문은 효율적인 구조 복잡도를 제시하는 곱셈기 설계에 그 목적이 있으므로 선형 시스톨릭 AB^2 모듈러 곱셈기만을 다룬다.



(a) 스냅샷-1 (b) 스냅샷-2 (c) 스냅샷-3

그림 7. 재 스케줄링된 DG의 스냅샷

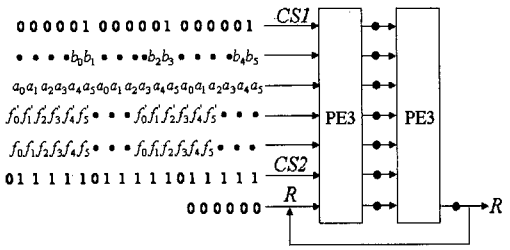


그림 8. 분할된 AB^2 선형 시스톨릭 곱셈기

그림 8은 그림 6의 데이터 의존 그래프로부터 수평방향으로 cut-set 시스톨릭 과정을 통해 유도된 $GF(2^6)$상의 선형 시스톨릭 곱셈기를 보여준다.

그림 8의 곱셈기는 그림 7의 각각의 수행을 위해서 3번의 중복된 데이터 입력이 필요하다. 그러나 연산의 중간 결과값 R은 한번의 입력이 수행된 후 나머지 두 번의 입력은 그림 8의 오른쪽 PE3의 출력 값을 이용한다. 그림 8의 효율적인 표현을 위하여 입력데이터 f'과 f는 한번의 스케쥴링을 위한 데이터를 생략하여 표현하였다. 즉, 입력데이터 f'과 f도 다른 입력처럼 3번의 중복된 데이터의 입력이 필요하다. 이러한 분할은 시스템의 요구사항에 따라서 일반화된 크기로 적용될 수 있을 것이다. 그러나 이렇게 일반화하기 위해서 고려해야 할 사항 각 기본구조의 재 스케쥴링에 따라서 피드백 되는 데이터에 대해서도 다시 스케쥴되는 클럭의 차이 만큼 데이터를 저장하고 원하는 시점에 데이터를 입력하게 할 수 있는 FIFO 구조와 같은 추가적인 장치가 필요하다는 점이다.

Ⅳ. 비교 및 분석

본 논문에서 제안한 구조는 Altera MAX+PLUSII를 이용하여 구조에 대한 검증을 수행하였다. 논문에서 AB^2 모듈러 곱셈을 위한 선형 시스톨릭 어레이가 제안되었다. 이들 두 구조는 거의 비슷한 속성을 가지므로, 본 비교에서는 논문 [12]에 초점을 맞춰서 본 논문에서 제안한 곱셈기와 비교 분석한다. 논문 [11]에서는 병렬 시스톨릭 구조의 속성을 변형한 구조를 제안하였다.

표 1에서 AND와 XOR게이트의 지연시간(delay)은 각각 2-입력 AND와 XOR 게이트이고, latch는 1bit이다. 효율적인 비교를 위해서 4-입력 XOR게이트와 3-입력 XOR게이트는 각각 3 2-입력과 2 2-입력 XOR로 고려한다. 논문 [12]의 구조는 그림 3에서 보여준 바와 같이 두 가지 기본 구조를 갖는다. 그

표 1. 비트 순차 구조의 비교

항목 \ 구조	논문[11] 구조	논문[12] 구조	분할된 구조
셀 수	$m^2/3$	m	$m/3$
셀복잡도	6-AND 6-XOR 17-latches	4-AND 3-XOR 14-latches 3-switch	4-AND 3-XOR 14-latches 3-switch
지연시간	$3m$	$3m$-2	$3m$

러나 효율성을 위하여 전체 구조가 그림 3 (a)로 구성된 것으로 간주한다. 본 논문에서 제안한 분할된 선형 시스톨릭 곱셈기는 논문[12]에서와 비슷한 각 셀 당 구조 복잡도를 갖는다. 그러나 본 논문의 구조가 기존의 구조보다 셀수를 2/3 만큼 줄일 수 있음을 확인할 수 있다.

Ⅴ. 결 론

본 논문에서는 $GF(2^m)$상에서 효율적인 지수연산을 수행하기 위한 분할된 AB^2 선형 시스톨릭 모듈러 곱셈기를 설계하였다. 이를 위하여 먼저 기존의 곱셈 알고리즘으로부터 재 스케쥴링된 데이터 의존 그래프를 유도하고, 시스템의 요구사항에 부합하는 효율적인 구조 복잡도를 얻기 위한 분할된 선형 시스톨릭 곱셈기를 설계하였다. 이러한 분할은 Wang과 Wei등의 곱셈기에도 적용할 수 있다. 표 1에서 보여준 바와 같이 본 논문의 분할된 구조가 기존의 구조의 복잡도를 1/3로 줄일 수 있음을 확인할 수 있었다.

참 고 문 헌

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

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

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

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

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

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

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

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

[9] S. W. Wei, "A systolic power-sum circuit for GF(2^m)," *IEEE Trans. on Computers*, 43, pp. 226-229, 1994.

[10] S. W. Wei, "VLSI architecture for computing exponentiations, multiplicative inverse, and divisions in GF(2^m)," *IEEE Trans. on Circuits and Systems*, 44, pp. 847-855, 1997.

[11] C. L. Wang and J. H. Guo, "New systolic arrays for C+AB^2 inversion, and division in GF(2^m)," *IEEE Trans. on Computers*, 49, pp. 1120-1125, 2000.

[12] N. Y. Kim, H. S. Kim, and K. Y. Yoo, "Computation of AB^2 multiplication in GF(2^m) using low-complexity systolic architecture," *IEE Proc.-Circuits Devices Sys.*, 150(2), pp. 119-123, 2003.

[13] S. Y. Kung, *VLSI array processor*, Prentice-Hall, 1987.

이 진 호(Jin Ho Lee) 정회원

1974년 2월 영남대학교 전자공학과 공학사

1981년 2월 영남대학교 전자계산학과 석사

1997년 2월 영남대학교 전자계산학과 박사

1979년 3월~현재 경일대학교 컴퓨터공학부 교수

<관심분야> 프로그래밍언어, 정보보호

김 현 성(Hyun Sung Kim) 정회원

1996년 2월 경일대학교 컴퓨터공학과 공학사

1998년 2월 경북대학교 컴퓨터공학과 석사

2002년 2월 경북대학교 컴퓨터공학과 박사

2002년 3월~현재 경일대학교 컴퓨터공학부 교수

<관심분야> 정보보호, 암호 프로토콜, 암호 프로세서 설계, IDS, 센서네트워크