네트워크 보안을 위한 다중모드 블록암호시스템의 설계

정회원 서 영 호*, 박 성 호*, 최 성 수**, 종신회원 정 용 진***, 김 동 욱*

Design of Multimode Block Cryptosystem for Network Security

Young-Ho Seo*, Sung-Ho Park*, Sung-Soo Choi**, Yong-Jin Jeong***, Dong-Wook Kim* *Regular Members*

요 약

본 논문에서는 IPsec등의 네트워크 보안 프로토콜을 위해 다중모드를 가지는 블록암호시스템의 구조를 제안하고 ASIC 라이브러리를 이용해서 하드웨어로 구현하였다. 블록 암호시스템의 구성을 위해서 AES, SEED, 그리고 3DES 등의 국내외 표준 블록암호화 알고리즘을 사용하였고 네트워크를 비롯한 유/무선으로 입력되는 데이터를 최소의 대기시간(최소 64클럭, 최대 256클럭)만을 가지면서 실시간으로 데이터를 암호화 혹은 복호화시킬 수 있다. 본 설계는 ECB, CBC, OFB뿐 아니라 최근 많이 사용되는 CTR(Counter) 모드를 지원하고 다중 비트단위(64, 128, 192, 256 비트)의 암/복호화를 수행한다. IPsec등의 네트워크 보안 프로토콜로의 연계를 위해 알고리즘 확장성을 보유한 하드웨어로 구현되었고 여러 암호화 알고리즘의 동시적인 동작이 가능하다. 적절한 하드웨어 공유와 프로그래머블한 특성이 강한 내부 데이터 패스를 통해 자체적인 블럭암호화 모드를 지원하기 때문에 다양한 방식의 암/복호화가 가능하다. 전체적인 동작은 직렬 통신에 의해서 프로그래밍되고 명령어의 디코딩을 통해 생성된 제어신호가 동작을 결정한다. VHDL을 이용해 설계된 하드웨어는 Hynix 0.25um CMOS 공정을 통해 합성되었고 약 10만 게이트의 자원을 사용하였으며, 100MHz 이상의 클럭 주파수에서 안정적으로 동작함을 NC-Verilog에서 확인하였다.

Key Words : hardware design; block cipher; cryptosystem; network security; ASIC; Multimode;

ABSTRACT

In this paper, we proposed an architecture of a cryptosystem with various operating modes for the network security and implemented in hardware using the ASIC library. For configuring a cryptosystem, the standard block ciphers such as AES, SEED and 3DES were included. And the implemented cryptosystem can encrypt and decrypt the data in real time through the wired/wireless network with the minimum latency time (minimum 64 clocks, maximum 256 clocks). It can support CTR mode which is widely used recently as well as the conventional block cipher modes such as ECB, CBC and OFB, and operates in the multi-bit mode (64, 128, 192, and 256 bits). The implemented hardware has the expansion possibility for the other algorithms according to the network security protocol such as IPsec and the included ciphering blocks can be operated simultaneously. The self-ciphering mode and various ciphering mode can be supported by the hardware sharing and the programmable data-path. The global operation is programmed by the serial communication port and the operation is decided by the control signals decoded from the instruction by the host. The designed hardware using VHDL was synthesized with Hynix 0.25um CMOS technology and it used the about 100,000 gates. Also we could assure the stable operation in the timing simulation over 100MHz using NC-verilog.

*광운대학교 전자재료공학과 Digital Design & Test Lab.(ddntlab.kw.ac.kr, design@kw.ac.kr), **한국전기연구원 전기정보망기술 연구그룹, ***광운대학교 전자통신공학과

논문번호 : 030215-0520, 접수일자 : 2003년 5월 20일

※본 연구는 IDEC(IC Design Education Center)의 지원에 의해서 이루어졌습니다.

I. 서 론

정보화 사회가 급진전되면서 인터넷을 이용한 정보통신의 수요가 급격히 증가되었고 개인 컴퓨터간의 정보통신이 활발히 이루어지고 있다. 이와 함께 일반적인 개방형 통신망을 이용한 컴퓨터 통신이 생활 전반에 필수적인 도구로 등장하고 있다. 이와 같이 유·무선을 통한 서로 다른 컴퓨터 통신망 사이의 상호 접속이 빈번해짐에 따라 개인정보와 유료정보에 대한 접근권한과 보호가 중요한 사항으로 대두되었고 정보보호를 포함해서 정보전달 및 저장형태를 정보공학적으로 발전시키기 위한 방안으로 암호학을 이용하고 있다[1]. 단순히 정보 보호라고 하면 암호화와 복호화를 의미하는 암호학의 함수적인 면이 강조되고 있으나, 현대의 통신환경 및 정보전달 형태에서 고도정보화 사회에 걸맞는 전자 송금, 전자 우편, 전자 거래, 홈쇼핑, 전자 현금, 전자투표 등의 실현은 암호학의 프로토콜적인 면이 강조되어야 하고 80년대 중반부터 대두된 암호화 프로토콜은 그 범위가 매우 넓으며 또한 그 역할이 매우 중요하다[2][3][4][5].

네트워크 상의 정보통신 보안을 위한 프로토콜 중에서 대표적인 것 중의 하나인 IPsec(Internet Protocal Security) 프로토콜은 네트워크나 네트워크 통신의 패킷 처리 계층에서의 보안을 위해 지금도 발전되고 있는 표준이다[6][7]. IPSec은 가상 사설망간에 다이얼업 접속을 통한 원격 사용자 접속의 구현에 특히 유용할 것이다. IPSec의 커다란 장점은 개별 사용자 컴퓨터의 변경 없이도 보안에 관한 준비가 처리될 수 있다는 것이다. IPSec은 본질적으로 데이터 송신자의 인증을 허용하는 인증 헤더인 AH (Authentication Header)와 송신자의 인증 및 데이터 암호화를 함께 지원하는 ESP (Encapsulating Security Payload) 등 두 종류의 보안 서비스를 제공한다. 이러한 각 서비스에 관련된 명확한 정보는 IP 패킷 헤더의 뒤를 잇는 헤더 속의 패킷에 삽입되고 ISAKMP/Oakley 프로토콜과 같은 별개의 키 프로토콜들이 선택될 수 있다[8][9]. 또한 고속 인터넷의 확산에 따른 기존의 보안 제품으로 인한 속도 저하문제가 대두됨에 따라서 기존의 소프트웨어(software, S/W) 기반의 보안제품에서 하드웨어(hardware, H/W) 기반의 보안제품으로 시장의 경향성이 변화되고 있고 관련된 H/W IC 제품이 출시되고 있다[10].

암호 알고리즘들을 소프트웨어로 구현할 경우, 쉽게 구현이 가능하며 적은 양의 데이터를 암호화할 경우에는 빠른 속도로 처리할 수 있다. 그러나 전자상거래나 무선인터넷보안시스템에서의 사용과 같은 실제 암호알고리즘을 응용할 때 경우에 따라 동시에 수십 건에서 수백 건의 암호화 작업을 수행해야 하며, 만약 이 때 소프트웨어로 암호화를 처리할 경우 시스템 과부하에 의해 처리속도가 현격히 떨어지게 된다. 이는 유·무선 인터넷을 이용한 적용에 암호알고리즘을 소프트웨어로 처리하는 것이 부적절함을 의미한다. 또한 해킹 등에 의한 암호알고리즘의 불법 접근 및 분석도 가능하여 암호알고리즘 자체의 안전 문제도 고려해야 한다. 그리고 이러한 암호시스템을 다른 시스템으로 쉽게 이식하고 설치 및 유지/보수가 용이하도록 암호시스템은 단일 칩으로 구현하는 것이 바람직하다. 따라서 빠른 속도의 암호화와 보다 안전한 암호화 처리를 위해 암호 알고리즘의 하드웨어적인 단일 칩화가 절실히 요구되고 있다. 현재 국내에서는 침입탐지시스템(IDS)이나 무선인터넷보안시스템의 고속 처리를 위해 컴퓨터에 탑재하던 소프트웨어를 하드웨어 전용장비로 개선하는 연구, 개발이 급속히 진전되고 있다. 최근 기업들의 전산시스템이 대용량 네트워크로 급전되는 환경에서 고대역 속도를 무리없이 지원하기 위해서는 보안기능을 하드웨어화하는 것이 필수적으로 되고 있다. 또한 ETRI 등 여러 연구 기관에서 암호화 알고리즘을 고속처리를 할 수 있는 ASIC 칩을 구현하고자 연구, 개발을 활발히 하고 있다[11][12][13][14].

본 논문에서는 독립된 시스템에 코 프로세서(co-processor)로서 고속의 암호화기능을 제공하고 S/W 기반의 보안기기가 가지는 약점을 보안할 수 있는 네트워크 보안을 위한 H/W 암호시스템을 설계하고자 한다. IPsec등의 네트워크 보안 프로토콜을 위해 다중모드를 가지는 블록암호시스템의 구조를 제안하고 ASIC 라이브러리를 이용해서 H/W로 구현하고자 한다. SEED, DES, 그리고 AES와 같은 암호 시스템의 경우에 이미 개별적인 구현은 많이 진행되어서 단일 칩이나 IP로써 다양한 상용 제품과 연구가 이미 발표되었다. 따라서 본 논문에서는 이들 각각의 블록 암호화 알고리즘에 대해서 100MHz 이상의 동작 속도를 요구하는 일반적인 충분조건을 만족시키면서 각각의 블록 암호 시스템들에 대한 내부적인 특성과 차별성보다는 IPsec 등과 같은 네트워크 프로토콜을 구현할 때 하드웨어적인 코 프로세서로서의 동작 속도를 높이는데 사용될 수 있고 이 경우 다양한 동작적인 특성을 위해서 프로그래머블한 특성을 가지는 하드웨어를 구현한다는데 중점을 두었

다. 즉, SEED, DES, 그리고 AES와 같은 자체 제작한 IP들을 가지고 특정 응용분야를 위한 하드웨어를 구현했다는 것에 중점을 두는 것이다.

2장에서는 IPsec 등의 네트워크 보안을 위한 블록암호 알고리즘의 사용에 대해서 살펴보고 3장에서는 구현된 H/W에 대해서 설명한다. 4장에서는 구현된 H/W의 동작에 대해 살펴보고 5장에서는 구현 결과 및 검증 결과를 보인다. 마지막으로 6장에서 결론을 맺으면서 논문을 마무리 한다.

Ⅱ. 네트워크보안과블록암호알고리즘

유·무선을 이용한 인터넷 사용의 증가에 따라 해킹 혹은 크래킹 문제에 대한 해결책이 중요시되고 있고, 유료화된 서비스를 제공하는 업체와 서비스 사용자간의 데이터 전송에 대한 보안요구가 가장 하위의 IP에 요구되고 있다. 인터넷상에서 기본 프로토콜인 IP에서 보안 서비스를 제공할 수 있도록 IETF에서 IPsec 표준과 이에 관련된 다른 표준들을 제안하였다. 이들 표준들은 인증프로토콜, 암호화프로토콜, 보안설정에 관련된 프로토콜, 보안설정과 보안정책 관리에 관련된 데이터베이스, 암호 및 인증용 암호알고리즘으로 구성되어 있다.

IPsec은 네트워크 계층에 보안 서비스를 제공해주는 것으로서 암호화와 인증의 암호학적 보안 서비스를 IP 패킷단위로 수행하고 현재 상용화되어 사용되는 IPv4 표준과 차세대 인터넷 프로토콜로 사용될 IPv6에 모두 보안 서비스를 제공한다[6][7]. IP 레벨에서 제공되는 6가지의 보안 서비스를 제공하기 위해 사용되는 프로토콜은 인증용 AH와 ESP로 구성된다. AH는 비연결형 무결성과 IP 데이터그램(datagram)들을 위한 데이터 근원인증(data origin authentication)을 제공하기 위해 사용되며 재연 공격에 대한 보호를 한다. AH는 단독으로 사용될 수도 있고 ESP와 함께 사용되어 터널모드를 구성할 수 있다. ESP 역시 단독으로 사용이 가능하고 AH와 조합을 이룰 수도 있는데 터널모드를 구성할 경우 AH에 의해서 보안서비스를 제공하면 ESP는 비밀성, 즉 암호화 서비스를 제공한다[8][9]. 이러한 IPsec의 암호화 서비스를 제공하기 위해 여러 블록 암호화 알고리즘이 사용되는데 대표적으로 AES, SEED, DES, 3중 DES, IDEA, 그리고 RC6 등이 있다[6][7]. 비밀성 서비스 제공 시 대용량의 데이터를 고속으로 처리해야하므로 하드웨어로의 구현이 요구되고 있고 구현 시 다양한 환경에 대한 요구사항을 만족시키는 적응성과 유연성을 가져야 한다. 즉, 사용자 혹은 전송 주체에 의해서 다양한 블록 암호화 알고리즘이 선택될 수 있고 선택된 알고리즘들에 여러 블록 암호화 모드를 적용하여 많은 동작적인 요구사항을 만족시켜야 한다. 이러한 블록 암호화 알고리즘과 블록 암호화 모드의 조합을 통한 다중모드로의 동작은 IPsec과 같은 프로토콜의 구현을 위해서는 필수적이라 할 수 있다. 그림 1에 암호화 동작이 요구되는 데이터를 ESP 모드를 중심으로 나타냈다. 그림 1에 나타난 것과 같이 전송모드에서는 ESP 모드에 대한 헤더(ESP hdr)에 의한 인증과 함께 ESP trailer(trl)까지 암호화가 적용되고 터널모드에서는 IP 헤더(IP hdr)까지 암호화 적용범위에 포함되어 기밀성을 유지한다. 따라서 이러한 암호화 과정을 수행하기 위해 헤더와 프로토콜에 대한 처리 과정이 암호화와 동시에 실시간으로 이루어져야 하고 따라서 전체적인 동작이 네트워크에 부하를 주어서는 안된다. 본 논문에서는 프로토콜에 대한 분석과 헤더들의 부가과정은 해당 시스템의 호스트 프로세서가지는 몫으로 가정하고 암호화를 수행하기 위한 적절한 제어 환경을 호스트에게 부여하는 것에 초점을 맞춘다.

본 논문에서 설계된 회로는 IPSec을 기반으로 하는 유·무선 네트워크 환경에서 고속의 데이터를 처리하기 위한 전용 코 프로세서 및 시스템 구현을 위한 H/W를 제안한다. 그림 2에 나타난 것과 같이 구현된 H/W는 다양한 환경에서 적용이 가능한 전용 프로세서로 Stand-alone VPN (Virtual Private Network), Firewall Integrated VPN, 그리고 Router Integrated VPN 등의 장비에 장착되어 네트워크에 지연을 주지 않고 독립적인 시스템의 암호화 기능을 담당할 수 있다. 그리고 ADSL, HomePNA 등의 개인 초고속 인터넷 장비내의 VPN Card 등에 적용되어 기밀성이 유지된 통신환경을 제공할 수 있다.

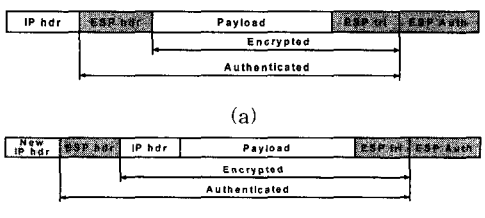


그림 1. IPsec의 ESP 모드에서 암호화 대상 (a)전송모드 (b)터널모드
Fig. 1. Encryption item in ESP mode of IPsec (a)transport mode (b)tunnel mode

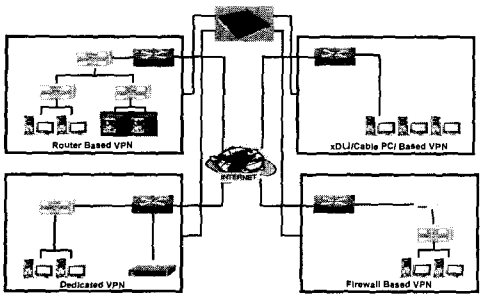


그림 2. 네트워크 보안을 위한 암호 H/W의 적용
Fig. 2. Application of cipher H/W for network security

Ⅲ. 제안된 H/W 구조

본 장에서는 전체적인 H/W의 구조를 설명하고 내부적으로 중요한 구조를 가지는 H/W 구성요소에 대해 구조를 나타내고 기본적인 동작을 설명한다.

1. 전체적인 H/W 구조

본 논문에서 제안되는 전체적인 H/W의 구조를 그림 3에 나타냈다. 그림 1에서 보였듯이 전체적인 H/W는 데이터 패스부(Data-Path part)와 제어부(Control part)로 구성되는데 데이터 패스부는 블록 암호화 및 복호화를 수행하는 부분(*Cipher*), 네트워크 환경과 인터페이스를 위한 입력 및 출력부(*Input/Output serial buffer*), 초기벡터를 저장하는 레지스터 및 카운터(*IV/Counter*), 블록 암호 모드를 위한 블록들로 구성된다. 블록 암호시스템은 AES[15], SEED[16], 그리고 3중 DES[17] 등의 국내외 표준 블록암호화 알고리즘으로 구성되었다. 또한 입출력 동작의 테스트와 추후 여타 알고리즘의 삽입을 위한 가상 암호화시스템(*Virtual Cipher*)도 내장되어 있다. 그리고 전체 H/W를 제어하는 제어기는 외부 호스트로부터 동작을 프로그래밍받고 저장하여 이를 디코딩하는 명령어 레지스터(*Instruction Register*)와 디코더(*Decoder*), 동작 순서를 결정하는 유한상태 기계블록(*FSMs*)과 상태 신호를 이용해서 제어 신호를 생성하는 제어신호 발생기(*Control Signal Generator*), 마지막으로 내부적인 칩의 상태를 외부로 알려주는 상태 레지스터(*Status Register*)로 구성된다.

H/W의 기본적인 동작모드에서 네트워크를 비롯한 유·무선으로 입력되는 데이터를 최소의 대기시간(최소 128클럭, 최대 256클럭)만을 가지면서 실시간으로 데이터를 암호화 혹은 복호화시킬 수 있는 특징이 있다. CFB를 제외한 모든 블럭암호화 모드(ECB, CBC, OFB)를 지원하고 최근 많이 사용되는 CTR(Counter) 모드도 지원한다. AES에 의해 128, 192, 256 비트의 키 길이를 가지는 암호화 동작이 가능하고 단일 DES, 2중 DES, 그리고 3중 DES 등의 다양한 DES 동작이 프로그래밍에 의해서 가능하다. IPsec등의 네트워크 보안 프로토콜로의 연계를 위해 알고리즘 확장성을 보유한 H/W로 구현되었고 여러 암호화 알고리즘의 동시적인 동작을 통해 H/W 사용률을 최대로 끌어올리는데, 이때는 네트워크 암호화 모드와 직접 암호화 모드가 동시에 사용된다. 네트워크 모드의 경우는 직렬 버퍼를 사용해야 하므로 한번에 하나의 암호화 시스템만을 사용할 수 있지만 네트워크 모드와 동시에 직접 입력 포트를 이용하여 네트워크 모드환경에서 사용되고 있는 암호 알고리즘을 제외한 다른 암호 알고리즘의 동작이 가능하다. 뿐만 아니라 네트워크 모드를 통한 암호 알고리즘의 동작과 함께 두개의 암호 알고리즘의 동시 동작이 가능하다. 그러나 이 경우 같은 시간에 동작을 시작할 수는 없고 입력 버퍼 혹은 출력 버퍼에서 충돌이 없도록 몇 클럭의 여유를 두고 동작시켜야 한다. 적절한 H/W 공유와 프로그래머블한 특성이 강한 내부 데이터 패스를 통해 자체적인 블럭암호화 모드가 지원 가능하기 때문에 다양한 방식의 암/복호화가 가능하다. 전체적인 동작은 직렬 통신에 의해서 프로그래밍되는데 총 프로그래밍에 17클럭을 사용하고 프로그래밍된 명령어의 디코딩을 통해 생성된 제어신호에 의해 전체적인 동작이 결정된다.

직렬 입력 버퍼를 통해 직렬로 입력된 데이터는 최대 128 비트 단위로 버퍼링되어 입력 버퍼(*Input buffer*) 혹은 초기벡터버퍼/카운터(*IV/Counter*)에 저장된다. CTR 모드의 경우 초기벡터를 카운팅하여 사용하므로 초기벡터버퍼는 카운터를 내장한다. AES가 256 비트 키를 사용하는 암호화 동작을 할 경우 초기벡터버퍼와 입력 버퍼에 각각 128 비트의 데이터를 저장하여 사용한다. 입력 버퍼에 저장된 데이터는 각 블록 모드 암호화 동작을 위해 저장 버퍼들(*Stage1/2 storage buffer*)을 통해 시간적인 버퍼링을 수행하여 데이터의 소실없이 암호화를 수행한다.

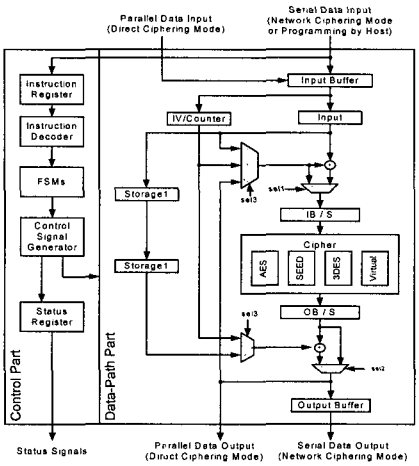


그림 3. 전체적인 H/W 구조
Fig. 3. Global H/W block diagram

2. AES의 H/W 구조

1998년을 기점으로 표준 기한이 만료된 DES를 대체할 블럭암호의 필요성에 따라, NIST(National Institute of Standards and Technology)에서는 향후 정부와 상업계에서 사용할 수 있는 강한 암호화 알고리즘 표준으로 AES(Advanced Encryption Standard)의 개발을 추진하였다. NIST는 3중 DES보다 더 효율적이고 안전하며 로얄티가 없어야 하는 등을 만족하는 알고리즘을 공모하고, 3년여에 걸쳐 15개의 후보 알고리즘을 공개적으로 평가하여, 2000년 10월 2일 최종 AES 알고리즘을 선정하여 발표하였다. AES에 채택된 블럭암호는 Daemem과 Rijmen에 의해 개발되고 RIJNDAEL로 명명된 알고리즘으로 DES와 3중 DES를 대신해서 새로운 업계 표준으로 자리잡아 가고 있다[15].

AES 암호 알고리즘은 입출력이 블록단위로 동작하는 알고리즘으로 기본 단위는 바이트(byte) 단위이고 내부적으로 2차원 배열 형태로 구성된다. AES의 암호화는 AddRoundKey(), SubBytes(), ShiftRows(), MixColumns(), 그리고 AddroundKey()으로 구성되고 마지막 라운드는 3개의 변환 있는데 SubBytes(), ShiftRows(), AddRoundKey()로 구성된다[15].

구현된 H/W에서는 데이터패스부와 제어부로 나누어져 있는데 데이터패스부는 라운드 키를 생성하는 키발생기(*Key Scheduler*), 생성된 라운키를 저장하는 키 저장레지스터(*Key Register*), 모듈로 연산을 테이블화하여 저장하고 있는 모듈로 롬(*Modulo ROM*), 암호화 연산을 수행하는 암호화기(*Cipher*)로 구성되며, 제어부는 키발생(*KeyScheduler*) 제어부, 암호화(*Cipher*) 제어부, 그리고 AES를 전체적으로 제어하는 전체 제어부로 구성된다. 암호화와 복호화를 위한 H/W는 자원 공유를 통해 통합된 구조로 설계되었고 비선형 함수의 모듈로 연산에 해당하는 부분은 모두 룩업 테이블(*Look-Up Table, LUT*)을 이용하여 설계함으로써 고속 동작을 할 수 있다. 구현된 AES는 128, 192, 그리고 256 비트의 키 길이를 모두 수용할 수 있도록 설계되었고 상황에 따라서 프로그래밍에 의해 쉽게 동작모드를 변환할 수 있다. AES의 구조를 그림 4에 나타냈다.

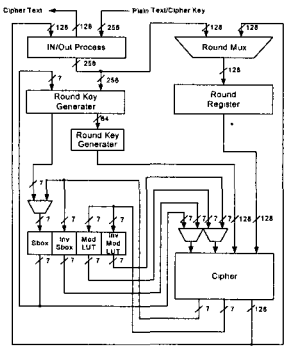


그림 4. AES의 H/W 구조
Fig.4. H/W structure of AES cryptosystem

3. SEED의 H/W 구조

SEED는 암호알고리즘의 중요성이 높아짐에 따라 1998년도에 대한민국 표준 암호화 알고리즘으로 개발되었다. SEED는 128비트 관용키 블록 암호알고리즘이며 기존의 암호알고리즘에 비해 속도와 효율성 면에서 우수함이 입증되어 현재 국내의 여러 응용분야에서 사용되고 있다[16].

본 논문에서 구현된 SEED의 전체적인 H/W 구조를 그림 5에 나타냈다. SEED는 128비트로 데이터를 처리하고 전체적으로 Feistel구조를 이룬다. 게이트 수를 줄이기 위해 G함수를 F 함수와 키 생성기가 시간적 스케쥴링에 의해 공유하는 구조를 가지고 G 함수 내부에 S-box는 원래 알고리즘에 비해 절반만을 H/W로 구현하고 반복하여 사용하도록 하였다.

SEED H/W는 그림 5에 나타낸 것과 같이 크게 데이터 패스부와 제어부로 구성하였다. 여기서 데이터 패스부는 라운드 키 생성부(*Key generator*), F 함수부(*F-function*), 라운드 키 저장부(*Round key register*), 입/출력 처리부(*In/out data process*), 그리고 라운드 처리부(*Round XOR*과 *round MUX*)로 구성하였고, 제어부는 3개의 유한 상태 기계(*Finite State Machine, FSM*)와 인코더(*Encoder*)로 구성하였다. 데이터 패스부 내의 모든 모듈들은 제어부에서

생성되는 제어 신호로 제어되도록 설계하였다.

제어부는 크게 FSM군과 인코더로 구성하였다. 제어부는 SEED 알고리즘의 동작을 수행하기 위해 데이터 패스부로 제어 신호들을 공급한다. 동작순서는 크게 라운드 키 생성부분과 암호화 혹은 복호화 부분으로 나뉘어 진다. 복호화는 암호화와 동일한 과정을 수행하나, 생성되는 라운드 키만 역순으로 적용된다.

4. 3중 DES의 H/W 구조

DES는 1975년 3월 17일 제안된 알고리즘이 표준 암호알고리즘으로 미연방 정부에 등록되고 안전성이 충분히 검토된 후 1977년 1월 15일 연방 정보처리규격 FIPS-46(Federal information processing standard)으로 등록되었고 DES(Data encryption algorithm)로 약칭화되었다. DES는 평문 64비트를 암호문 64비트로 변환시키는 암호방식으로 64비트의 키를 사용하고 있다. 이 키는 8비트마다 패리티(Parity) 비트를 하나씩 포함하고 있어 DES의 암호화 과정에는 56비트만이 적용된다[17]. 또한 DES의 안전성에 있어서 문제가 제기되자 다중 DES 암호화 방식의 사용을 고려하게 되었는데 이는 서로 다른 키로 DES 암호 방식을 반복 적용하면 암호의 안전도가 향상된다는데 기인하였다. 다중 키(K_1, K_2)를 이용한 2중 DES(Double-DES)와 3중 DES(Triple-DES)에 대한 식을 각각 식 (1)과 식 (2)에 나타냈는데 E_{K1}은 K_1 암호키를 이용하여 암호화를 수행하는 것을 뜻하고 D_{K2}는 K_2 암호키를 이용하여 복호화를 수행함을 의미한다.

$$Ciphertext=(E_{K2}(E_{K1}(M)) \quad (1)$$

$$Ciphertext=(E_{K1}(D_{K2}(E_{K1}(M))) \quad (2)$$

3중 DES는 그림 6에 나타난 단일 DES를 구현한 뒤 그림 7에 보이는 것과 같은 다중 DES (Multiple-DES)를 위한 모듈의 부가로 구현된다. DES의 경우 비교적 간단한 알고리즘으로 SEED와 유사한 Feistal 구조를 가지고 16라운드를 수행하여 암호화 및 복호화를 수행한다. 그림 7에 나타난 것과 같이 DES의 반복적 사용을 통해 2중 DES 및 3중 DES의 동작을 할 수 있고 3중 DES 제어(*3DES Control*)블록이 프로그래밍에 따라서 DES의 다양한 동작을 제어한다. 즉, 그림 7의 *DES* 블록이 그림 6의 단일 DES에 해당하는데 *MUX*를 통한 궤환구조를 형성시켜 DES의 암호화 혹은 복호화된 결과를 재입력 받아 다중 DES를 수행한다. 이때 3중 DES 제어 블록은 DES 블록을 적절히 암/복호화 모드로 동작시키고 그에 해당되는 암호키를 입력하여 준다.

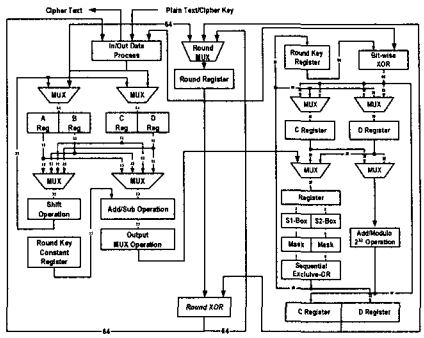


그림 5. SEED의 H/W 구조
Fig.5. H/W structure of SEED cryptosystem

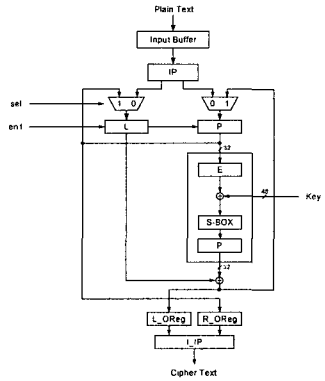


그림 6. 단일 DES의 H/W 구조
Fig.6. H/W structure of single-DES cryptosystem

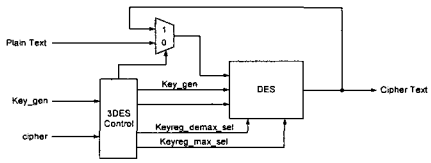


그림 7. 다중 DES의 H/W 구조
Fig. 7. H/W block diagram of multiple-DES

Ⅳ. 제안된 H/W의 동작

본 장에서는 구현된 H/W의 동작을 결정하기 위한 프로그래밍 항목에 대해 살펴보고 이러한 프로그래밍 항목에 따라서 어떠한 동작을 하는지를 설명한다. 또한 네트워크 암호모드에서 직렬 입출력을 위한 H/W 구조에 따른 동작과 일반적인 사용을 위한 직접 암호모드의 동작을 설명하여 전체적인 H/W의 동작적 특징과 유연성에 대해서 살펴본다.

1. 제안된 H/W의 프로그래밍과 동작

구현된 H/W는 표 1에 나타난 항목을 프로그래밍하여 동작한다. "Cipher Mode"의 결정을 통해 암/복호화를 정하고 "Operation Mode"를 통해서 네트워크

프로토콜을 위한 네트워크 암호모드로 사용할 것인지 직접적인 암호알고리즘의 사용을 위한 직접 암호모드를 사용할 것인지를 결정한다. 각각의 프로토콜의 상황과 입력 데이터의 특성에 따라서 적절한 암호화 방식을 선택할 수 있도록 프로그래밍하여 H/W를 유연하게 사용할 수 있다.

프로토콜의 상황에 맞게 프로그래밍되면 H/W는 호스트 프로세서의 지시에 의해서 네트워크를 통해 입력되는 데이터를 암호화 혹은 복호화 시킬 수 있다. 네트워크 암호모드의 경우 직렬로 입력되는 데이터를 입력 특성에 맞추어 프로그래밍된 조건에 따라서 데이터를 버퍼링하고 데이터 크기가 조건에 만족되면 이 데이터를 암호화 혹은 복호화시킨다. 또한 이와 동시에 그 다음의 직렬 데이터를 한 클럭의 손실도 없이 버퍼링하여 다음 동작을 준비한다. 암호화 알고리즘의 선택에 따라서 출력률이 다르지만 제어기에 의해 각각의 프로그래밍 조건에 따라 전체 동작이 입력 버퍼링 시간과 동일하게 자동적으로 맞추어지기 때문에 일정한 출력률을 유지할 수 있다. 이러한 동작적 특성은 일정(최소 128, 최대 256 클럭) 시간의 대기지연시간을 필연적으로 가지지만 이러한 대기지연은 암호알고리즘을 사용하면 피할 수 없는 요소이고 전체 네트워크의 소통에 부하를 주는 것은 아니므로 문제되지 않는다.

2. 모드 구성에 따른 H/W의 동작

앞 절에서 설명한 것과 같이 구현된 H/W는 네트워크 암호모드와 직접 암호모드로 나뉜다. 직접 암호모드의 경우는 특별한 처리없이 직접적으로 암호 알고리즘에 데이터를 입력하여 암호화 과정을 수행하는 것으로 암호 알고리즘들이 단독으로 존재하여 동작하는 것과 같다. 그러나 네트워크 암호모드의 경우는 직렬 입력에 대해 시간적인 손실없이 입력 데이터를 동일한 비트율로 출력해야 하므로 대기 지연시간이 발생하고 이를 위한 버퍼가 요구되었고 그에 따른 하드웨어 구조를 설계하였다. 그림 8에 네트워크 암호모드에서 직렬 입력을 버퍼링하여 암호화를 수행하고 이를 직렬로 출력하는 과정을 나타냈다. 동작의 최초부터 고려하면 먼저 시스템 내의 외부 호스트 프로세서에 의해서 동작과 모드가 프로그래밍되어 H/W의 동작이 구성되고 다음에 직렬 키를 입력 받아서 이를 일정 블록의 크기(그림에서 128 비트)로 버퍼링한다. 버퍼링된 키는 128 클럭이 지난 다음에 암호 알고리즘을 통해 라운드 키를 생성하게 되고 그와 동시에 연속적으로 평문을 입력받으면서 일정 블록(그림에서 128 비트)으로 버퍼링을 수행한다. 128비트의 블록이 형성되면 지정된 블록 암호 알고리즘을 통해 암호화를 수행하고 이와 동시에 입력 버퍼링부는 다음 평문을 입력받는다. 따라서 프로그래밍과 키의 입력을 제외하고 데이터만을 고려할 때 최초의 평문 입력으로부터 최초의 암호문 출력까지는 최대 256 클럭의 대기지연이 발생하고 이후에는 연속적인 직렬 출력을 발생한다.

표 1. H/W 동작을 위한 프로그래밍 항목
Table 1. Programming items for H/W operation

	Programming Item	Number of Bits			
1	Ciphering	0	Operation		
		1	Not-Operation		
2	Cipher Mode	0	Encryption		
		1	Decryption		
3	Operation Mode	0	Network Cipher Mode		
		1	Direct Cipher Mode		
4	Block Cipher Selection (Network Cipher Mode)	00	SEED	10	DES
		01	AES	11	Virtual Cipher
5	Block Cipher Mode	00	ECB	10	OFB
		01	CBC	11	CTR
6	AES Key-Length	00	128	10	256
		01	192	11	N/A
7	DES Mode	00	2-DES	10	3-DES
		01	1-DES	11	N/A
8	Operation Length		8-bit (Maximum 256)		
	Total	17 bits			

V. 설계 및 시뮬레이션 결과

1. 설계 및 합성결과

제안된 블록 암호시스템은 VHDL(VHSIC Hardware Description Language)을 이용하여 H/W로 설계하였다[18]. 설계는 VHDL 하향식(top-down) 설계 기법을 통해서 이루어졌으며 특정 구현 대상에 국한하지 않는 범용적인 설계를 이루고자 오직 IEEE 표준 라이브러리만을 사용하였다. 각각의 모듈들은 RTL (Registar Transfer Level)수준으로 설계되었고, 구조적 수준(structure-level)에서 서로 연결되었다. 설계된 H/W를 검증하기 위해 Synopsys의 Design Complier™로 논리 합성을 수행하였다[19]. 그림 9에는 구현된 H/W의 합성 결과를 RTL 수준에서 보이고 있다. 사용된 암호화 알고리즘들은 일반적 회로에 비해 비교적 데이터 버스가 크기 때문에 합성 시 팬아웃(fanout), 배선 지연(wire delay) 그리고 CTS (Clock Tree Synthesis)등에 주의해야 한다. VHDL을 이용해 설계된 H/W는 Hynix 0.25μm CMOS 공정을 통해 합성되었고 약 10만 게이트의 자원을 사용하였

다. 또한 NC-Verilog를 이용한 타이밍 시뮬레이션을 통해 100MHz 이상에서 안정적으로 동작함을 확인하였다. 표 2에 각각의 H/W 모듈이 사용하는 자원을 나타냈다. 전체적으로 암호 알고리즘들이 큰 H/W 사용률을 보이고 있고 그 중에 AES가 전체에 대해 약 38%를 차지하면서 가장 큰 H/W 자원을 사용한다.

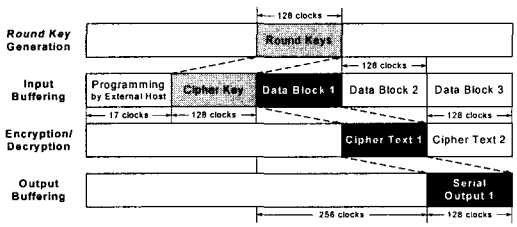


그림 8. 네트워크 모드에서의 H/W 동작순서
Fig. 8. H/W operation sequence in the network mode

표 2. 구현된 H/W의 자원 사용율
Table 2. Resource usage of the designed H/W

Module	Number of Gates
Multiple-DES	23,425 (22.2%)
AES	40,521 (38.2%)
SEED	23,333 (21.9%)
Virtual Cipher	1,404 (1.32%)
I/O Interface	2,941 (2.79%)
Control Part	423 (0.39%)
Block Mode	14,050 (13.2%)
Total	106,097 (100%)

2. 구현된 회로의 시뮬레이션

Synopsys에서 합성된 회로의 시뮬레이션은 NC-Verilog에서 이루어졌다. 그림 10은 전체적인 시뮬레이션 결과를 나타냈는데 합성 후 Vela를 이용해서 추출된 Netlist의 시간적 지연 정보(SDF file)를 사용해서 타이밍 시뮬레이션을 수행하였다. 실제로 DES, SEED, 그리고 AES 각각의 회로에 대한 임계경로가 다르기 때문에 각각에 대한 동작 주파수가 다를 수 있지만 본 논문에서는 P&R 후에 칩 단위로 동기화된 회로에 대해서 패드(Pad) 지연, 배선 지연 등의 모든 물리적인 조건과 제어기의 성능을 모두 포함한 타이밍 시뮬레이션을 통해 얻어진 실제적인 칩의 동작 주파수를 측정하였고 측정결과 100MHz의 동작 주파수 이상에서 안정적으로 동작하였다. 그림 10에서 보는 바와 같이, 데이터가 직렬 입력되면 일정 대기지연시간을 지나고 나서부터는 암호화된 직렬 데이터가 출력되는 것을 볼 수 있다. 처음은 SEED를 이용한 네트워크 암호 모드의 결과를 보이고 있고 가운데 부분은 AES를 사용하여 CFB 블록 암호모드를 적용한 시뮬레이션 결과이다. 결과의 우측 하단은 DES를 이용한 직접 암호모드의 시뮬레이션 결과를 나타낸다. 그림 10의 결과 중 일부를 그림 11과 12에 자세히 보였다. 그림 11는 네트워크 암호모드에서 AES를 사용하는 경우를 나타낸 것으로 CFB 블록 암호모드를 사용하는 시뮬레이션 결과인데, 그림에서 보이듯이 초기 입력 벡터와 키에 대한 처리를 제외하면 256 비트가 지난 후부터 직렬 출력이 발생하는 것을 확인할 수 있다. 그림 12는 직접 암호모드에서 3DES를 선택하여 사용하는 경우의 시뮬레이션 결과를 나타내는 것으로서 대기지연 시간이 요구되지 않으면서 입력과 출력이 곧바로 발생하는 것을 볼 수 있다. 전체적인 시뮬레이션 결과를 관찰하면 각각의 암호 알고리즘의 동작에 대해서 키를 생성할 경우에는 "key_rdy" 신호가 발생하고 암호 혹은 복호화 동작을 할 경우에는 "data_rdy"신호가 발생하여 현재 동작 상태에 있는 암호 알고리즘을 외부로 알려준다. 네트워크 암호모드에서는 결과에서 보이듯이 기본적으로 128 비트단위로 동작이 맞추어져 있고 DES를 사용할 경우는 64 비트 단위 동작을 한다. 즉, 대기지연 시간은 256 클럭이 아니라 128 클럭이 된다.

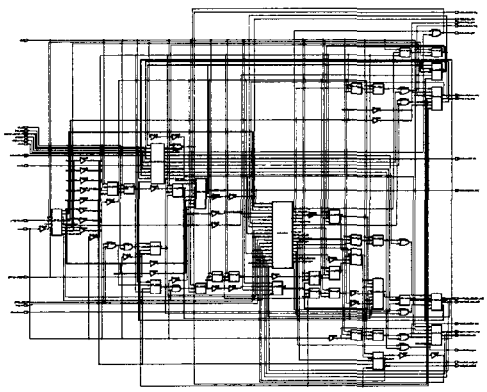


그림 9. Synopsys를 이용한 RTL 수준의 합성도
Fig. 9. RTL synthesis diagram using Synopsys

3. P&R 결과

합성된 회로에 입출력 포트를 위한 패드를 붙인 후 Apollo를 이용하여 P&R(Place and route)을 수행하였고 DRC 및 LVS 과정을 통해서 에러없이 결과를 추출하였다. 전체 칩의 크기는 4000x4000mm이고 설계된 코어(core)의 이용률(utilization)은 57%에 해당한다. 그림 13에 Apollo를 이용하여 P&R을 수행한 결과를 나타내었다.

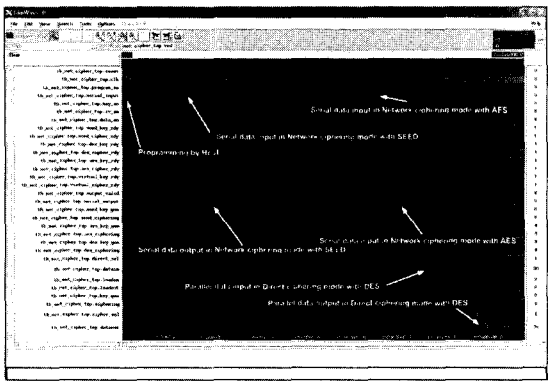


그림 10. 네트워크 암호모드의 전체적 시뮬레이션 결과
Fig. 10. Global simulation result in network ciphering mode

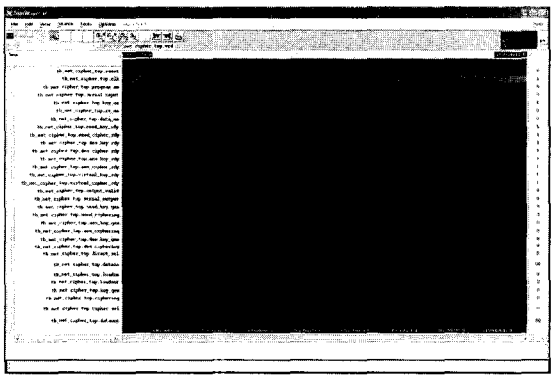


그림 11. 네트워크 암호모드에서 AES를 이용한 OFB 블록 암호모드의 시뮬레이션 결과
Fig. 11. Simulation result of OFB mode with AES in network ciphering mode

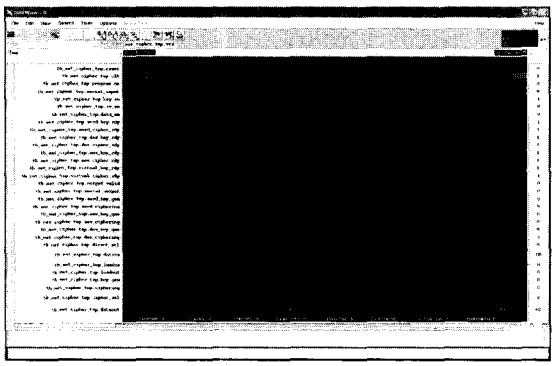


그림 12. 직접 암호모드에서 3DES를 이용한 ECB 블록 암호모드의 시뮬레이션 결과
Fig. 12. Simulation result of ECB mode with 3DES in direct mode

Ⅵ. 결 론

본 논문에서는 IPsec등의 네트워크 보안 프로토콜을 위해 다중모드를 가지는 블록암호시스템의 구조를 제안하고 ASIC 라이브러리를 이용해서 H/W로 구현하였다. 블록 암호시스템의 구성을 위해서 AES, SEED, 그리고 3DES 등의 국내외 표준 블록암호화 알고리즘을 사용하였고 네트워크를 비롯한 유·무선으로 입력되는 데이터를 최소의 대기시간(최소 64클럭, 최대 256클럭)만을 가지면서 실시간으로 데이터를 암/복호화시킬 수 있다. 모든 블럭암호화 모드(ECB, CBC, CFB, OFB)를 지원하고 최근 많이 사용되는 CTR(Counter) 모드도 지원하면서 다중 비트단위(64, 128,192, 256 비트)의 암/복호화를 수행한다. IPsec등의 네트워크 보안 프로토콜로의 연계를 위해 알고리즘 확장성을 보유한 H/W로 구현되었고 여러 암호화 알고리즘의 동시적인 동작이 가능하였다. 적절한 H/W 공유와 프로그래머블한 특성이 강한 내부 데이터 패스를 통해 자체적인 블럭암호화 모드가 지원 가능하기 때문에 다양한 방식의 암/복호화가 가능하다. 전체적인 동작은 직렬 통신에 의해서 프로그래밍되고 명령어의 디코딩을 통해 생성된 제어신호가 동작을 결정한다. VHDL을 이용해 설계된 H/W는 Hynix 0.25μm CMOS 공정을 통해 합성되었고 약 10만 게이트의 자원을 사용하였다. 또한 NC-verilog를 이용한 타이밍 시뮬레이션을 통해 100MHz 이상에서 안정적으로 동작함을 확인하였다.

본 논문에서 설계된 회로는 IPSec을 기반으로 하는 고속 데이터 처리용 보안 전용 프로세서로 Stand-alone VPN (Virtual Private Network), Firewall Integrated VPN, Router Integrated VPN, 그리고 ADSL, HomePNA 등의 개인 초고속 인터넷 장비내의 VPN Card 등에 적용가능할 것으로 보인다.

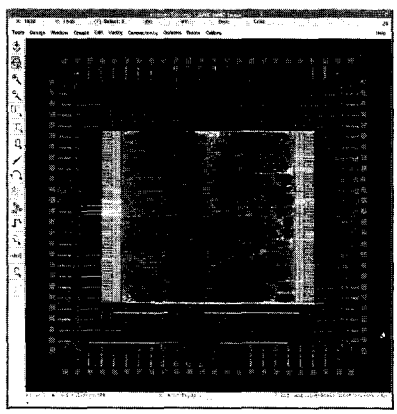


그림 13. Apollo를 이용한 P&R결과
Fig. 13. P&R result using Apollo

참 고 문 헌

[1] Willianm Stallings "Cryptography and Network Security", Prentice Hall, 2003

[2] Sidnie Fiet, "TCP/IP : Architecture, Protocals and IMplenentation With IPv6 and IP Security", Dec. 1998

[3] Jalal Feghhi and Peter Williams, "Digital Certificated/ Applied Internet Security", pp. 127-161, Addison-Wesley

[4] Christopher M. King, Curtis E. Dalton and T. Ertem Osmanoglu, "Security Architecture : Design, Deployment & Operations", RSA press, 2001

[5] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocal", RFC 2408, Nov. 1998

[6] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocal(IPsec)", RFC 2401, Nov 1998

[7] IP Security Protocal(IPsec), IETF, http:// www.ietf.org/html.charters/ipsec-charter.html

[8] S. Kent and R. Atkinson, "IP Authentication Header(AH)", RFC 2402, Nov, 1998

[9] S. Kent and R. Atkinson, "IP Encapsulation Security Payload(ESP)", RFC 2406, Nov 1998

[10] Kamel H. Rahouma, "A Block Cipher Technique For Security of Data and Computer Networks", Internet workshop, IWS 99, pp. 25-31, 1999.

[11] Jiang Anping, Sheng Shimin, Fu Yiling, Liu Yue, and Ji Lijiu, "The Design and Implementation of a Block Cipher ASIC", ASIC, 2001. Proceedings. 4th International Conference on, pp. 344-347 2001

[12] A. Schubert, and W. Anheier, "Efficient VLSI Implementation of Modern Symmetric Block Cipher", Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on, pp. 757-760 vol. 2, 5-8 Sep 1999

[13] Yukio Mitsuyama, Zaldy Andales, Takao Onoye, and Isao Shirakawa, "Burst Mode : A New Acceleration Mode for 128-bit Block Ciphers", Custom Integrated Circuits Conference, 2002. Proceedings of the IEEE, pp. 151-154, 2002

[14] National Institute of Standards and Technology(NIST), Advanced Encryption Standard (AES), National Technical Information Service, Springfield VA 22161, Nov. 2001

[15] 한국정보보호센터, *128비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서*, 12. 1998.

[16] National Bureau of Standards. FIPS PUB 46 : Data Encryption Standard, January, 1987.

[17] I.S 1076-1993, IEEE Standard VHDL Language Reference Manual, IEEE, 1993.

[18] Synopsys Inc, "Guidelines and Practices for Successful Logic Synthesis", August 1997.

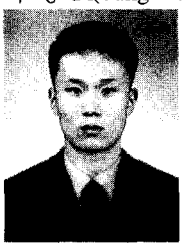
서 영 호(Young-Ho Seo) 정회원

1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사).
2001년 2월 : 광운대학교 대학원졸업(공학석사).
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원.
2001년 3월~현재 : 광운대학교 전자재료공학과 박사과정.
2003년 6월~현재 : 한국전기연구원 연구원

<주관심분야> Image Processing/Compression, 워터마킹, 암호학, FPGA/ASIC 설계

e-mail : design@kw.ac.kr

박 성 호(Sung-Ho Park) 준회원



2000년 3월 : 광운대학교 전자재료공학과 졸업(공학사).
2002년 6월~현재 : 광운대학교 전자재료공학과 석사과정.

<주관심분야> Image Compression, MPEG, FPGA/ASIC 설계
e-mail : psh12280@explore.kw.ac.kr

최 성 수(Sung-Soo Choi) 정회원

1996년 2월 : 경원대학교 전자공학과 졸업(공학사)
1998년 2월 : 광주과학기술원 정보통신공학과 졸업(공학석사)
2003년 2월 : 광주과학기술원 정보통신공학과 졸업(공학박사)
2003년 2월: 한국전기연구원 전기정보망기술연구그룹 연구원

<주관심분야> 저전력, 초고속 유무선 통신이론/시스템설계, 오류정정부호화기설계, UWB전송시스템>
e-mail : sschoi@keri.re.kr

정 용 진(Yong-Jin Jeong) 종신회원

1983년 2월 서울대학교 제어계측공학과 졸업
1983년 3월 ~ 1989년 7월 전자통신 연구소(ETRI)
1991년 5월 미국 UMASS 전자전산공학과(공학석사)
1995년 2월 미국 UMASS 전자전산공학과 박사.
1995년 4월 ~ 1999년 2월 삼성전자 반도체 수석연구원
1999년 3월 ~ 현재 광운대학교 전자공학부 조교수.

<주관심분야> 컴퓨터 연산 알고리즘, ASIC 설계, 무선 통신, 정보보호
e-mail : yjjeong@daisy.ac.kr

김 동 욱(Dong-Wook Kim) 종신회원

1983년 2월 : 한양대학교 전자공학과 졸업(공학사).
1985년 2월 : 한양대학교 대학원 졸업(공학석사).
1991년 9월 : Georgia 공과대학 전기공학과 졸업(공학박사).
1992년 3월~현재 : 광운대학교 전자재료공학과 정교수. 광운대학교 신기술 연구소 연구원.
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원.

<주관심분야> 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication
e-mail : dwkim@daisy.gwu.ac.kr