1998년도 한국음향학회 학술발표대회 논문집 제 17 권 제 1(s) 호

FPGA를 이용한 암호 알고리즘의 구현

The Implementation of Crypto-Algorithm Using FPGA

이상덕*, 이계호**, 한승조*
조선대학교 전자 · 정보통신시스템설계 Lab.*
한국통신 인력개발본부**

Sang Duck Lee, Seung-Jo Han*
Dept. of Electronic and Information Communication Eng. Chosun Univ.*
Korea Telecom. HRDG**
E-mail : sjbhan@mail.chosun.ac.kr

요 약

최근 개인 휴대통신과 컴퓨터 기술의 발달로 유용한 데이터의 질적 · 양적 향상을 가져왔다. 이로 인해 저장중이거나 선로상에서의 전송중인 정보의 보호문제가 중요시되고 있다. 이러한 정보보호 문제가 중요시됨에 따라 정보보호를 위한 직접적인 암호화 방법중의 하나인 IDEA(International Data Encryption Algorithm)의 구현을 제안하고자한다. IDEA는 블록 암호화 방식의 하나로서 64비트 데이터를 암호화하기 위해 128비트의 키를 사용한다. 본 논문에서 암호알고리즘 구현을 위하여 하드웨어 설계언어인 VHDL을 사용하였고, V-System을 이용하여 Simulation을 수행하였다. Coding된 알고리즘은 Synopsys를 사용하여 자동합성하였고, Xilinx사의 FPGA-4025를 Target으로 구현하였다.

I. 서 론

최근 컴퓨터 기술과 정보통신 기술의 발달은 대량의 정보를 신속 정확하게 처리하여 제공함으로써 정보의 효율성, 활용성, 편의성의 증대를 가져왔다. 이로 인해 개인의 컴퓨터를 이용한 정보의 송수신이나 건물내에서의 통신을 이용한 결재, 다른 곳으로의 유용한 데이터의 전송이 손쉽게되었다. 정보사회에서 정보의 신속한 전송 및 보관 또는 비인가자로부터의 정보의 보호가 중요한 문제로 대두되고 있다. 이러한 무형적인 정보가 저장중이거나 통신망을 통해 전송 될 때 불법 침입자로부터 데이터를 보호하고 도청이나 내용의 변조를 막기 위해서는 정보를 암호화하는 것이 절실히 요구된다. 뿐만아니라 통신선로상에서 보관중이거나 전송중인 정보의 집적적인 보호문제도 크게 대두되고 있다.

암호화(encryption)란 데이터를 무의미하게 무작위로 나타나도록 스크램블(scramble)하여 나중에 되찾을 수 있게 보호하는 과정을 말한다. 암호화된 메시지의 수신자는 그 왜곡되어 있는 메시지를 원래의 알기 쉬운 형상대로 해독하거나 제대로 해놓을 수 이어야한다. 이 알고리즘들은 평문 또는 cleartext라 불리는 보호하고자 하는 원래의 정보를 갖고, 그것을 key라 부르는 연산자를 사용하여 스크램블된다. 수신자가 암호문을 다시 복호시키기 위해서는, 사용되어진 암호화 알고리즘을 정확히 알고 있음에도 불구하고, 정확한 key를 갖고 있어야만 원래의 평문으로 복구 시킬수 있다. 키사용 방식은 상대방과의 오직 하나의 메시지 또는 하나의 대화에만 사용될 세션 키 하나를 설정하거나, 아니면 노출을

최소화시킨다는 기본 원칙에 따라 키들을 변경하는 메커니즘을 설정하여야한다.

Ⅱ. IDEA알고리즘의 설명

IDEA(International Data Encryption Algorithm)는 64bit의 블록에서 데이터를 암호화하기 위해 128bit의 키를 사용하는 블록 암호방식이다. IDEA의 메시지 전체의 암호화 방법은 이전의 블록을 암호화한 결과를 다음 암호문의 계산으로 집어넣는 피드백 루프 형태를 사용하여 완료된다. IDEA의 블록 길이는 통계적 분석을 막을 수 있을 만큼 충분히 강력하다고 인식되고있고, 128bit의 키의 사용은 전사적 키 검색을 막을 수 있을 만큼 충분히 길다고 여겨지고 있다.

그림 1. 전체 IDEA의 구조

IDEA의 복호화 처리는 암호화 처리와 유사한 과정을 거친다. 복호화 과정에서는 Plain text 대신 암호문을 입력으로 사용함으로써 원래의 Plain-text가 얻어진다. 그러나 각 라운드에 필요한 서브키는 입력으로 사용된 키의 역원을 사용한다. 복호과정 n번째에서의 처음 네 개의 서브키는 암호과정 (10-n)에서의 처음 4개의 서브키로부터 유도된다. 처음과 네 번째 복호 서브키는 첫 번째와 네 번째 암호 서브키에 대응하는 법 $(2^{16}+1)$에서의 곱셈의 역과 같다. 2번째 라운드를 거쳐 8번째 라운드까지에 대해서, 두 번째와 세 번째 복호 서브키들은 세 번째와 두 번째 암호 서브키에 대응하는 법(2^{16})에서의 덧셈의 역과 같다.

Ⅲ. IDEA 블록의 설계

본 논문에서는 IDEA 암호알고리즘을 구현하기 위해 Top-down 디자인 방식을 채택하였다. 설계흐름도는 아래의 그림과 같다. 설계 소프트웨어로 Synopsys를 사용하여 시스템레벨 합성을 하였고, Xilinx tool을 사용하여 Download 하였다.

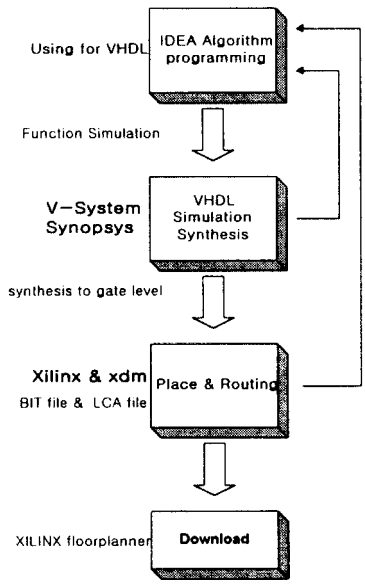

그림 2. 구현절차 순서도

본 논문에서는 IDEA의 블록을 크게 Keyin Blcok, Encryption Block, Key-Round, Control Block의 네 개로 나누었다.

1. Keyin Block

Plain_text, OP, K_in을 입력으로 받아들이어 암호화부에 입력으로 사용될 data를 생성하고, 입력값 64bit를 128bit로 확장시켜 Key Block에 입력값으로 사용한다.

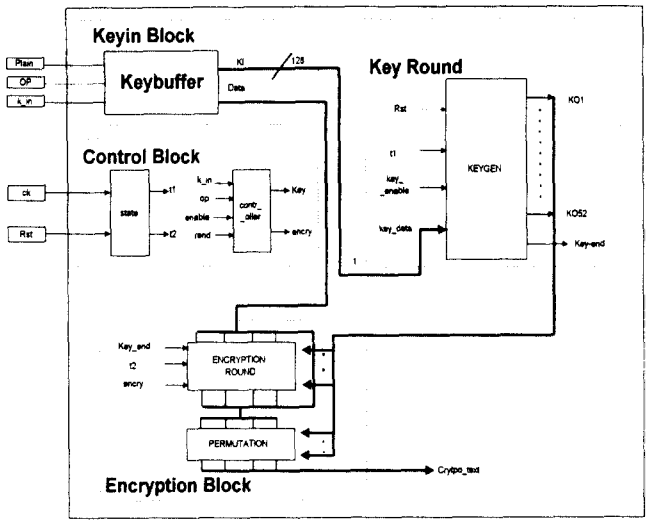

그림 3. IDEA의 블록다이어그램

2. Control Block

control blcok 블록은 암호기 전체를 제어하는 controller와 두 개의 cycle을 가지는 클럭을 발생하는 State로 구성되어있다. 외부에서 클럭을 직접 받아들여 동작을 시키며, 전체적인 동작제어는 enable과 Rst를 통하여 제어한다.

*** State**

IDEA의 Encryption Block과 Key Round는 배타적인 동작을 수행할 필요가 있다. Key Round와 Encryption Block이 동시에 동작하게 되면, 블록에서 소요되는 시간에 의해 서로 동기가 맞지 않고 입력키의 오류를 일으켜 잘못된 연산을 수행할 우려가 있다. 클럭 발생부는 2개의 각기 다른 클럭을 발생시키고, Key Round나 제어부에 사용되어질 클럭은 가장 빠른 클럭을 사용하고, Encryption 부에는 다음 순위의 클럭이 사용된다. 실제 설계에 사용된 클럭 발생기는 외부의 클럭을 받아들이어 분주시키는 분주 회로를 사용하였다.

*** controller**

제어신호를 발생시켜 모든 부분의 유기적인 동작을 시킨다. 외부의 enable, op 등의 제어신호를 입력으로 받아들이며 출력되는 제어신호인 key와 encry는 다른 블록들의 수행 여부를 결정짓는다. 궁극적으로 controller는 암·복호화부와 키 발생부의 우선순위를 정해주는 것이 목적이다.

2. Key Round

IDEA에 사용되어진 52개의 16bit 서브키들은 128bit의 암호키로부터 생성된다. 생성 방식은 처음 8개의 서브키들은 최상위 16bit 키들을 그대로 사용하고, 그 다음의 16bit 키도 차례로 나머지 비트와 대응되어 만들어진다. 128bit key 입력을 받아들이어 16bit 서브키 8개를 생성하고, 그 다음에 사용되어질 키들은 128bit의 키를 25bit 순환 shift 시킴으로써 생성시킨다. 단일 라운드에 사용되는 96개의 서브키 비트는, 처음과 8번째 반복을 제외하고 서로 인접해있지 않고 한 라운드의 서브키와 다른 라운드의 서브키 사이의 간단한 이동관계 조차 없다. encryption시 입력으로 사용되어질 6개의 서브키들은 각 round에서 사용될 키들을 미리 계산하여 출력하게 하였으며, key는 미리 생성되어 encryption에 사용되어야 하므로, 가장 빠른 클럭인 t1을 사용하였으며, key_enable 신호에 의해 동작여부를 결정짓는다.

3. Encryption Block

Encryption Block은 Control Block으로부터 encry 신호를 받게 되면 동작을 수행하게 된다. 각 라운드에서의 출력은 평문과 키의 연산에 의해 유도되며, 8 round의 동작이 끝나면 permutation 과정을 거쳐 암호문을 출력한다. permutation은 암호화 과정과는 무관하나 복호시에 동일 알고리즘을 사용하기 위하여 data의 자리바꿈을 해주는 기능을 한다. encryption 과정은 key 생성될 때까지 대기 상태에 있어야 하므로, t1보다 느린 t2를 사용하였다.

Ⅳ. 구현고찰

통신선로 상에서의 정보의 효율적인 보존을 위하여 직접적인 암호화를 위하여 암호 알고리즘인 IDEA를 VHDL을 사용하여 설계하였고, Xilinx FPGA Chip 4025 Target으로 구현하였다. 전체적인 알고리즘 전체를 구현하였고, 입력측의 단자수를 줄이기 위하여 암호화를 요하는 데이터를 확장시켜 키의 입력으로 사용하였다. 칩의 속도 개선을 위하여 키 생성부와 암호화부를 동시에 동작하게 하였으며, delay문제는 서로 다른 주기의 클럭을 사용하여 키 생성부에 우선순위를 두었다.

그림 7은 IDEA의 전체 블록의 합성결과이다. 합성시 tool은 Synopsys를 사용하였으며, Simulation은 V-system을 사용하여 검증하였다.

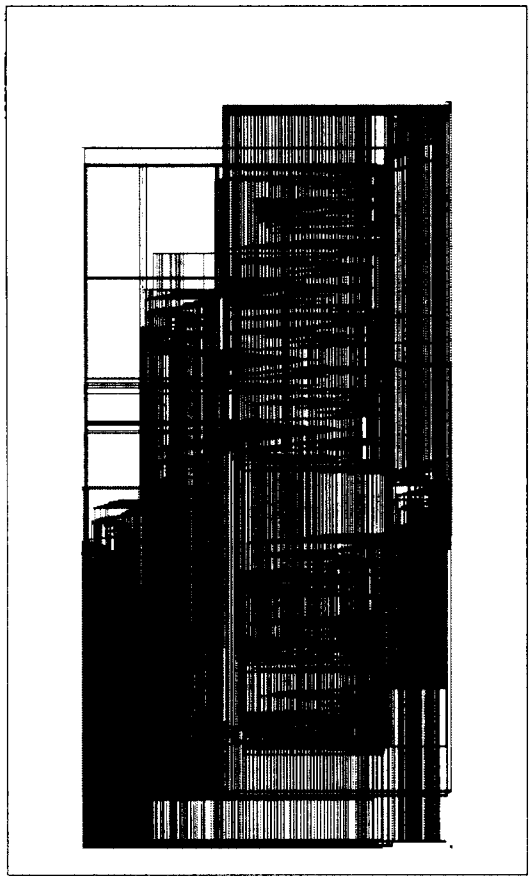

그림 4 IDEA의 전체 합성 결과

V. 결론

본 논문에서는 IDEA 암호화 알고리즘을 Xilinx FPGA를 이용한 하드웨어 구현을 목적으로 VHDL을 사용하여 Coding 하였다. 구현물은 FPGA-4025를 대상으로 하여 합성하였다. 표에서와 같이 Encryption block은 연산자 수가 많아서 가장 많은 시간과 면적을 소요하였다. 향후 Encryption block의 Area와 delay를 최소화시키는 문제가 연구과제로 남아있다. IDEA는 암·복호화과정에서 Key를 생성할 때 비교기, 가·감산기 등이 많이 사용되므로 Cell 숫자를 줄여야 할 것이다. 따라서 이 부분을 미리 계산된 data를 저장한 ROM으로 대치시키면 속도면 에서 매우 향상 될 것이라고 고려되어진다. 또한 서브키 생성과 encryption시 서로 다른 cycle의 두 클럭을 가지고 수행시켰는데, 동일한 클럭을 사용하여 약간의 delay를 두고 연산을 시키면 더욱 속도를 개선시킬 수 있으리라고 기대된다.

	area	time
keyin	23	3.78
controlblock	15	3.25
keyround	247	10.7
encryptionblock	2105	31.78
total	2390	49.51

표 Chip의 area와 time 분석

<참 고 문 헌>

1. J. B. Kam, & S. E. Tavares, "Structured Design of Substitution Permutation Encryption Network," IEEE Transactions on Compiters, Vol.28, no.10, pp.747 753m 1989.

2. Schneirer Bruce, Applied Cryptography, John Wiley & Sons, Inc., pp.219-296.

3. H. H. Evertse, "Liner Structures in Block Ciphers," Advabeces in Cryptology-EUROCRYPT '87, proceedings. Berlin:Springer-Verlag, pp.249-266, 1987.

4. G. I. Davida, D. J. Linton, C. P. Szelag &D. L. Well, "Data base security," IEEE Transactions on Software Engineering, Vol.SE-4, no.6, pp.531-533, 1978.

5. R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Commu. ACM, Vol.21(12), 1978, pp.993-999.

6. Baker, Richard H. Computer Security Hnadbook. New York : McGraw-Hill, Inc., 1991.

7. Man Young Rhee, "Cryptography and Secure Communications", 1993.8. Rivest, R., "The MD5 Message-Digest algorithm", July, 1991.

9. The Great Lakes Data Systems, "Enhanced Cable Billing Systems", 1993.

10. Horster P., Knoblock H.-J., "Cryptograhpic Protocols and Network Security", Security and Control : From Small systems to Large, Proceedings of IFIP/Sec 1992.