제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

IPsec 암호 알고리즘의 통합 설계

김진범, 송문빈, 정연모
경희대학교 전자공학과
e-mail : gojinjingo@hotmail.com

Integrated Design for IPsec Cryptography Algorithms

Jinbeom Kim, Moonvin Song, Yunmo Chung
Dept. of Electronic Engineering, Kyung Hee University

요 약

IPsec(Internet Protocol Security protocol)에서는 기본적으로 4개의 암호 알고리즘(3-DES, AES, MD5, SHA-1)을 사용하고 있다. 본 논문은 4개의 암호 알고리즘을 효율적으로 통합 및 하드웨어로 설계하였으며 검증하였다. 그 결과 알고리즘을 각각 합친 경우보다 하드웨어에서의 크기를 줄일 수 있다.

1. 서론

암호 알고리즘은 정보화 사회에서 정보의 보안을 유지하기 위한 필수 요소이다. 정보통신 기술이 발달함에 따라 정보의 사용처 및 사용량이 날로 증가하고 있다. 이에 비례하여 정보의 보안성 노출의 위험성 또한 증대하고 있다. 인터넷의 빠른 보급과 발전에 의하여 각종 금융 서비스, 전자 상거래와 이를 위한 각종 카드 결제 등이 인터넷상에서 행해지고 있다. 따라서 이와 관련된 각종 범죄들이 속속 들어나고 있다. 이를 대비하기 위한 한 예로서 IPv6에서는 IPsec의 사용이 의무화 되었다. 여기서 IPsec는 인터넷상에서 발생할 수 있는 보안을 강화한 것으로 기본적으로 4개의 암호화 알고리즘을 사용하여야 한다[5,6]. 따라서 IPsec를 사용한 보안시스템의 장비는 수많은 네트워크 장비에 탑재되고 있으며 사용범위가 매우 넓다. 보안을 위한 범용적인 IPsec의 기능을 각 알고리즘 단위로 사용하기 위해서는 구현 및 응용에 있어서 하드웨어적인 일정한 제약이 따르게 된다[2,7].

본 논문은 IPsec에서 의무적으로 사용하는 4 개의 암호 알고리즘인 3-DES, AES, MD-5, 그리고 SHA-1을 통합 설계하였다. 이때 각 구조를 분석하여 유사 기능을 통합 설계함으로써 보안 시스템의 크기를 줄이는 것을 목적으로 한다.

2. IPsec 암호 알고리즘

여기서는 4 개의 IPsec 기본 암호 알고리즘인 3-DES, AES, MD5, SHA-1에 대해서 설명하고자 한다.

3-DES와 AES는 암·복호화가 가능한 암호화 알고리즘이며, MD5와 SHA-1은 해쉬 알고리즘으로서 임의의 길이의 비트 열을 고정된 길이의 출력 값인 해쉬 코드로 압축시키는 알고리즘이다[4,5].

첫 번째로 3-DES를 설명하기 위해서 먼저 DES를 설명한다. DES는 NIST(National Institute of Standards and Technology)에서 제정된 표준 암호화 알고리즘으로 Feistel 구조를 갖는 대표적인 대칭키 암호화 알고리즘 이다. 64비트로 이루어진 평문과 키를 받아들여서 총 16라운드에 걸친 치환 및 확장 단계를 거친 후 결과 값을 출력한다[1,5].

3-DES는 [그림 1]과 같이 3 개의 DES를 연속적으로 사용하는 구조를 가지고 있다.[5]

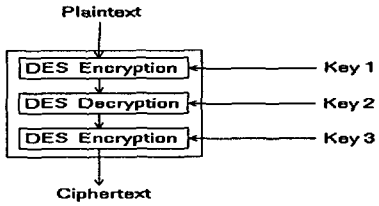

그림 1. 3-DES 구조

그림의 첫 번째 DES는 암호화 과정을 수행하여 결과 값을 두 번째 DES의 입력으로 전달한다. 두 번째 DES는 복호화 과정을 이용한 암호화를 수행하며 그 결과 값을 세 번째 DES의 입력으로 전달한다. 세 번째 DES는 암호화 과정을 수행하여 최종 결과 값을 출력한다[5].

두 번째로 AES 알고리즘을 설명한다. AES 알고리즘은 고정된 128 비트의 데이터에 대하여 128, 192, 그리고 256 비트의 키를 지원하고 있다. 그리고 10, 12, 14 라운드의 연산을 각각 수행한다. DES와는 다르게 non-Feistel 구조를 바탕으로 하며 독립된 역변환이 가능한 블록으로 구성되어 있다[5]. [그림 2]에서 AES 알고리즘의 전체적인 수행 과정을 나타내었다.

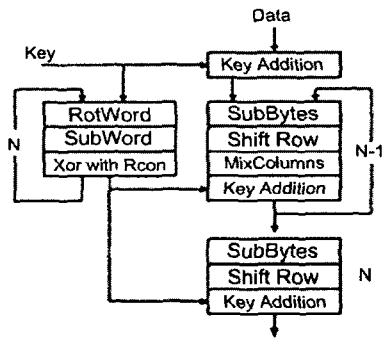

그림 2. AES 알고리즘

암호화 및 키 생성과정에서 모두 N번의 라운드 연산을 수행한다. RotWord와 Shift Row 단계는 Shift 연산을 수행한다. SubWord와 SubBytes는 S-Box를 사용하여 데이터의 치환을 수행한다. Xor with Rcon과 Key Addition은 각각 XOR 과정을 수행하는데 Rcon은 AES에서 미리 정해져 있는 상수이다. MixColumns 단계는 유한체 상에서의 나머지 연산을 수행한다[1,2].

세 번째로 MD5 알고리즘을 설명한다. MD5는 내부적으로 512비트 단위의 메시지 처리를 기본으로 하며 결과 값으로 128비트의 해쉬값을 출력한다. 512비트의 입력은 16개의 32비트 메시지 블록으로 나뉜다. 512비트의 데이터는 4라운드의 단계연산을 거치며 각 라운드는 16단계의 처리과정이 있다. 각 라운드는 [그림 3]과 같은 단계연산을 수행한다[3].

$A = D,\ C = B,\ D = C,$

$B = B+((A+F_R(B,C,D)+X[\rho(j)]+K[j])<<<S))$

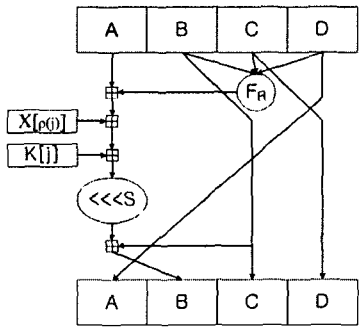

그림 3. MD5의 단계연산

변수 A, B, C, D는 각각 32비트로 구성되어 있으며 그 값이 정해져 있다. F는 각 단계별로 처리되는 함수이다. X는 메시지 블록이며 K는 정해져 있는 상수이다. j는 단계의 수를 나타내며 p는 단계별로 합해지는 메시지 블록을 나타낸다[5].

마지막으로 SHA-1을 설명한다. SHA-1역시 MD5처럼 4개의 라운드로 구성되지만 각 라운드의 단계연산은 MD5와 달리 20번의 반복수행을 필요로 한다. SHA-1의 단계연산은 [그림 4]과 같은 구조로 되어 있다[4,5].

$A = S^5(A)+Fj(B,C,D)+E+X[l(j)]+K[j]$

(S^5 : 5비트 좌측 순환이동)

$B = A,\ C = S^{30}(B),\ D = C,\ E = D$

(S^{30} : 30비트 좌측 순환이동)

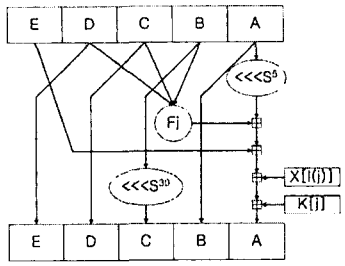

그림 4. SHA-1의 단계연산

A, B, C, D는 MD5의 A, B, C, D와 동일한 값이며 총 5개의 변수를 갖는다[5].

3. 구현 및 결과

여기서는 앞에서 언급한 네 개의 알고리즘을 하드웨어로 통합 설계하는 과정을 설명하고 그 결과를 분석하고자 한다.

[그림 5]는 3-DES를 구현한 블록도이다.

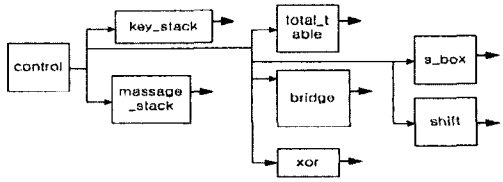

그림 5. 3-DES 구현 블록도

control 블록이 모든 블록을 제어하며 control 블록을 포함한 key_stack, masage_stack, des_xor, shift 블록은 다른 알고리즘의 블록과 공동사용이 가능한 블록으로서 통합 설계시에 공간적 효율을 높일 수 있는 부분이다. total_table 블록은 3-DES에서 사용되는 모든 치환 및 순열 단계를 하나의 블록으로 설계하였다.

AES의 구현은 [그림 6]과 같다.

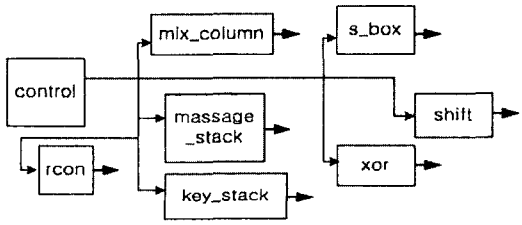

그림 6. AES 구현 블록도

본 블록도 에서도 control 블록과, key_stack, massage_stack, shift, xor 블록은 3-DES와 마찬가지로 통합 설계시 공간적 효율성을 높일 수 있는 부분이다.

3-DES와 AES를 통합 설계한 블록도는 [그림 7]과 같다.

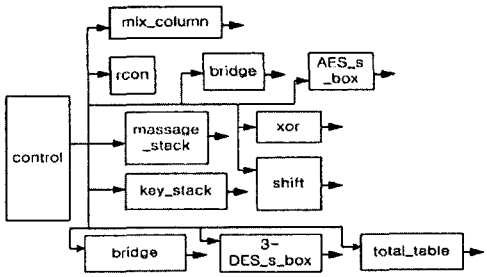

그림 7. 3-DES와 AES 통합 구현 블록도

control 블록은 두 가지 암호화 알고리즘 모두에 해당되는 카운트를 생성해 내는 블록으로서 같은 카운트 신호를 가지고 두 암호화 알고리즘이 공유하게 된다. AES가 3-DES보다 key_stack과 massge_stack의 큰 크기를 필요로 한다. 따라서 AES의 시스템 자원을 공유할 수 있도록 3-DES의 stack을 재설계 하였다. xor 및 shift 블록은 AES의 경우 127 비트 연산을 하게 되며 이는 3-DES보다 크다. 따라서 이 또한 stack과 같이 AES 자원을 활용할 수 있도록 3-DES를 재설계 하였다. 이 이외의 블록들은 고유한 특성을 지닌 블록으로서 통합이 불가능 하다.

MD5와 SHA-1은 매우 유사한 구조를 가지고 있으며 두 알고리즘을 통합 설계한 블록은 [그림 8]과 같다.

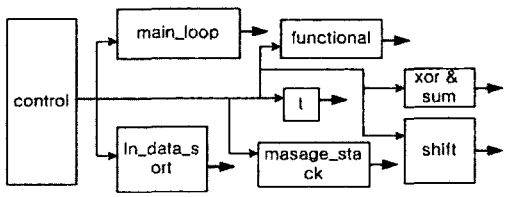

그림 8. MD5와 SHA-1의 통합 구현 블록도

functional 블록과 main_loop, control 블록은 두 알고리즘에서 다소 차이가 나는 기능적인 면을 보이지만 통합 설계가 가능한 부분이다. 이들 블록을 제외한 나머지 블록은 두 알고리즘에서 모두 같다.

main_loop 블록은 MD5가 SHA-1 보다 큰 사이즈를 갖는다. 통합 설계 시 이를 고려하여 MD5의 컴포넌트를 활용할 수 있는 SHA-1을 구현 하였다.

네 개의 암호화 알고리즘을 모두 통합하여 구현하였을 때의 블록도는 [그림 9]이다.

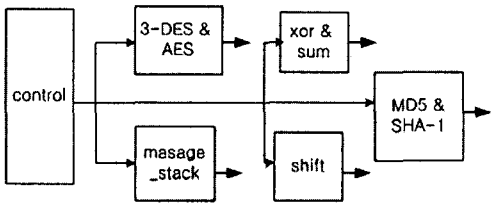

그림 9. 3-DES, AES, MD5, SHA-1
통합 설계 블록도

control 블록과 xor&sum, shift, masage_stack 블록은 네 개의 알고리즘 모두에서 추출하여 통합한 블록이다. 통합이 불가능한 블록은 각각 3-DES & AES 블록과 MD5 & SHA-1 블록에 따로 구현하였다.

지금까지 설계한 블록도의 logic cell 수는 [표 1]에 나타내었다.

표 1. logic cell 수

3-DES	control	k_stack	m_stack	total_ta	bridge
	72	1277	134	497	279
	s_box	shift	xor		**total**
	553	178	48		**3293**
AES	control	rcon	mix_col	m_stack	xor
	56	22	797	273	128
	s_box	shift	k_stack		**total**
	945	303	1465		**4389**
3-DES & AES	control	mix_col	rcon	A_sbox	m_stack
	96 (-32)	797	22	945	328 (-79)
	k_stack	xor	shift	bridge	D_sbox
	1657 (-1085)	128 (-48)	424 (-57)	279	559
	total_ta				**total**
	497				**5867 (-1815)**
MD5	control	main_lp	func	t	xor
	38	3491	261	632	217
	in_data	m_stack	shift		**total**
	528	1072	554		**6913**
SHA-1	control	main_lp	func	t	xor
	33	635	320	66	281
	in_data	m_stack	shift		**total**
	528	3560	66		**5891**
MD5 & SHA-1	control	main_lp	func	t	xor
	58 (-13)	3768 (-358)	459 (-122)	698	378 (-120)
	in_data	m_stack	shift		**total**
	528 (-528)	4007 (-625)	586 (-34)		**10487 (-3317)**
3-DES AES MD5 SHA-1	control	D & A	xo&su	m_stack	shift
	147	4891	413	4068	679
	M & S				**total**
	5458				**15656**

4개 암호 알고리즘 각각의 logic cell 수의 합은 20486(3293+4389+6913+5891)개이다. 본 논문에서 4개의 암호 알고리즘을 통합 설계한 결과 15656개의 logic cell수를 필요로 하였다.

칩은 Altera 사의 EPX20KE1500 칩을 타겟으로 하였으며, Active HDL, Quartus, Modelsim, Symplify을 사용하여 설계 및 검증하였다.

참고문헌

[1] C. Sanchez-Avila, R. Sanchez-Reillo, "The rijndael block ciper(AES proposal): a comparison with DES," *2001 IEEE 35th International Carnahan Conference on Security Technology* pp. 229-234, Oct 2001.

[2] T. Ichikawa et al, "Hardware Evaluation of the AES Finalists," in *Proceeding. 3th AES Candidate Conference*, New York, April 13-13, 2000.

[3] R. Rivest, *The MD5 Message-Digest Algorithm*, RPC 1321, MIT LCS & RSA Data Security, Inc, 1992.

[4] U.S. Department of Commerce, *Secure Hash Standard*, FIFS PUB 180-1, NIST, 1995.

[5] Man Young Rhee, *Internet Security*, Wiley, 2002.

[6] 쓰지이 시게오, *암호와 정보사회*, 한마음사. 2000.

[7] Michael Welschenbach, *(C와 C++로 구현하는) 암호화 알고리즘*, 인포북. 2003.

[8] 정연모 외, *VHDL을 이용한 디지털 설계*, 미래컴. 2003.