논문 2005-42SD-4-4

ARIA 암호 알고리듬의 하드웨어 설계 및 구현

(Design and Implementation of ARIA Cryptic Algorithm)

박 진 섭*, 윤 연 상*, 김 용 대**, 양 상 운**, 장 태 주**, 유 영 갑***

(Jinsub Park, Yeonsang Yun, Young-Dae Kim, Sangwoon Yang, Taejoo Chang, and Younggap You)

요 약

본 논문은 2004년 12월 국내 표준(KS)으로 제정된 ARIA 암호 알고리듬의 하드웨어 구조를 처음으로 제안하고 있다. ARIA 암호 알고리듬은 알려진 공격에 대하여 안전하며, Involution SPN (Substitution Permutation Network)으로써 구조적 효율성도 높다. 1 cycle/round 구조로 갖는 제안된 ARIA 구조는 회로 크기를 줄이기 위해 s-box를 듀얼 포트 롬과 배럴 로테이터를 채택한 고속의 라운드 키 생성기를 포함하고 있다. 제안된 ARIA는 Xilinx VirtexE-1600 FPGA를 사용하여 구현하였고, 1,490 slices와 16 RAM 블록을 사용해서 437 Mbps의 성능을 낸다. 설계된 ARIA 블록을 검증하기 위해서 영상 데이터를 암호화(복호화)하여 통신하는 시스템을 개발하였다. 설계한 ARIA는 IC 카드뿐만 아니라 데이터 저장이나 인터넷 보안 규격(IPSec, TLS)과 같은 많은 데이터를 고속 처리가 필요한 응용에 적용될 수 있다.

Abstract

This paper presents the first hardware design of ARIA that KSA(Korea Standards Association) decided as the block encryption standard at Dec. 2004. The ARIA cryptographic algorithm has an efficient involution SPN (Substitution Permutation Network) and is immune to known attacks. The proposed ARIA design based on 1 cycle/round include a dual port ROM to reduce a size of circuit and a high speed round key generator with barrel rotator. ARIA design proposed is implemented with Xilinx VirtexE-1600 FPGA. Throughput is 437 Mbps using 1,491 slices and 16 RAM blocks. To demonstrate the ARIA system operation, we developed a security system cyphering video data of communication though Internet. ARIA addresses applications with high-throughput like data storage and internet security protocol (IPSec and TLS) as well as IC cards.

Keywords : ARIA, Cryptographic, Security, Hardware architecture, video security system

I. 서 론

최근 인터넷을 통한 현장 감시, 동영상의 채증 및 개인 프라이버시 보호에 관한 연구가 증가하고 있다. 국가보안기술연구소에서는 정보 통신 서비스의 다변화 및 전자 정부 구현 등으로 인한 국가기관과 민간 (G2C) 간 소통 자료에 안전성과 효율성을 제공 할 수 있는 암호 알고리듬으로서 ARIA을 제안하였다[1]. ARIA는 AES의 SPN 구조와 DES, Camellia의 Feistal 구조를 참고로 하는 Involution SPN (Substitution-Permutation Network) 구조이고, 128 비트 블록을 128 비트, 192 비트 그리고 256 비트의 3 종류의 키 사용해 암호화(복호화) 한다[1]. ARIA의 입, 출력 크기와 사용 가능한 키 크기는 미국 표준 블록 암호인 AES (Advanced Encryption Standard)의 입, 출력 크기 및 사용 가능한 키 크기와 동일하다[2]. ARIA는 차분 공격 (Differential crypt-analysis), 선형 공격(Linear crypt- analysis)과 같은 공격 방법 등에 대하여 비교적 안전하다. 2003 년 말 벨기에 루벤 대학에서의 성능평가 결과에서 SEED 나 일본의 암호화 알고리듬인 Camellia보다 데이터 처리속도 등 성능 면에서 2배가량 앞선 것으로 나타났다[3].

* 학생회원, *** 정회원, 충북대학교 정보통신공학과
(Department of Information & Communication Engineering, Chungbuk Nat'l University)
** 정회원, ETRI 부설 국가보안기술연구소
(National Security Research Institute)
※ 본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과이며 충북대학교 컴퓨터정보통신연구소의 일부지원에 의하여 수행되었음.
접수일자: 2004년10월18일, 수정완료일: 2005년3월25일

ARIA 블록 암호화 알고리듬의 성능을 실제 응용 환경에서 평가할 필요가 있다. 이 알고리듬은 아직 하드웨어 구현이 발표되고 있지 아니하다. 실제 운용 환경에서 이 알고리듬의 성능을 제시하는 것이 실용화의 중요한 과제이다. 동급 경쟁 기술인 AES의 여러 연구를 살펴보고 ARIA를 위한 제안된 구조와 비교해보고자 한다. 하드웨어 구현에서 효율성은 처리 속도 및 하드웨어의 면적으로 평가한다. 2000년 128 비트 미국 표준 블록 암호 알고리듬 AES로 채택된 Rijndael은 하드웨어와 소프트웨어 설계에 관한 많은 연구가 이루어졌다. AES 암호 알고리듬을 소프트웨어나 하드웨어로 설계하기 위한 구조에 대한 연구는 초기의 단순한 알고리듬을 하드웨어 기술하면서 성능을 높인 형태에서 최근의 회로 크기에 따른 최적의 구조를 제안하는 연구와 같은 많은 연구가 이루어졌다.

1 라운드 반복 구조를 갖는 AES[6]는 암호기 기능만 수행하면서 542 slices와 10 RAM 블록을 사용하여 1,450 Mbps의 성능을 낸다. AES[7]는 암호기 설계와 암/복호기 설계를 제안하고 있다. 암호기는 1 라운드 반복 구조를 가지면서 125.38 MHz로 동작하여 1,563 Mbps의 성능을 낸다. 회로 크기는 1,857 slices 이다. AES 알고리듬이 암호화 과정과 복호화 과정이 다르기 때문에 암/복호기는 암호기 회로의 두 배 이상의 회로 크기를 요한다. AES[7]의 암/복호기는 5,150 slices로써 암호기보다 약 3배 커졌고, 동작 주파수도 76 MHz로 암호기에 비해 크게 느려졌다. 다른 구조로써 병렬 처리와 라운드 함수를 파이프라인 기법으로 잘라 동작 속도를 증가시키는 구조가 있다. AES[8]은 라운드 함수를 3단과 7단으로 잘라 성능을 비교하였다. 3단의 경우는 회로 크기는 9,406 slices이고 93.5 MHz에서 11,965 Mbps의 성능 보이고 있고, 7단의 경우는 회로 크기는 11,022 slices이고 168.4 MHz에서 21,556 Mbps의 성능을 보이고 있다. 4 사이클/라운드 구조를 갖는 AES[9]는 32 비트단위로 처리하여 163 slices로 208 Mbps의 성능 내고 있다. AES[9]는 면적 대비 성능 (Mbps/slices)으로 우수한 회로로써 임베디드 어플리케이션에 적합한 회로이다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 ARIA 알고리듬을 소개하고 하드웨어 구조를 제안한다. Ⅲ장에서는 제안한 구조로 설계한 ARIA 블록을 FPGA를 이용해 검증하여 결과를 분석하였고, 설계된 ARIA 암호칩을 이용한 영상 암호 시스템을 소개한다. 마지막으로 이를 토대로 Ⅳ장에서는 결론을 맺는다.

Ⅱ. ARIA 하드웨어 설계

본 장에서는 ARIA 알고리듬을 설명하고, 이 알고리듬에 적합한 하드웨어 구조를 제안한다. 암호화(복호화)는 유한 체 안의 곱셈 연산을 사용한 involution 치환과 확산계층을 통해서 이루어진다. ARIA의 키 생성은 하드웨어로 구현할 때 비트 순환 방법과 초기화 과정을 통해서 초고속 회로 설계가 가능하다. 또한 상수 값만을 변경하여 동일한 키 스케줄을 적용하기 때문에 가변 키 크기 구현에 적합하다.

1. 기본 알고리듬

ARIA 알고리즘은 암호화와 복호화를 수행하는 라운드 함수와 키 스케줄러로 구성되어 있다. ARIA 라운드 함수의 기본 구조는 Involution SPN 구조이다. 입, 출력의 크기는 128 비트이고, 키의 크기는 128 비트, 192 비트 그리고 256 비트를 선택할 수 있다. 키의 크기에 따라 12, 14 또는 16 번 라운드 함수를 반복 수행한다. 그림 1과 같이 ARIA의 라운드 함수는 확산계층 대신 라운드 키 덧셈을 수행하는 마지막 라운드를 제외하고는 라운드 키 덧셈(AddRoundKey), 치환 계층(SubstLayer)

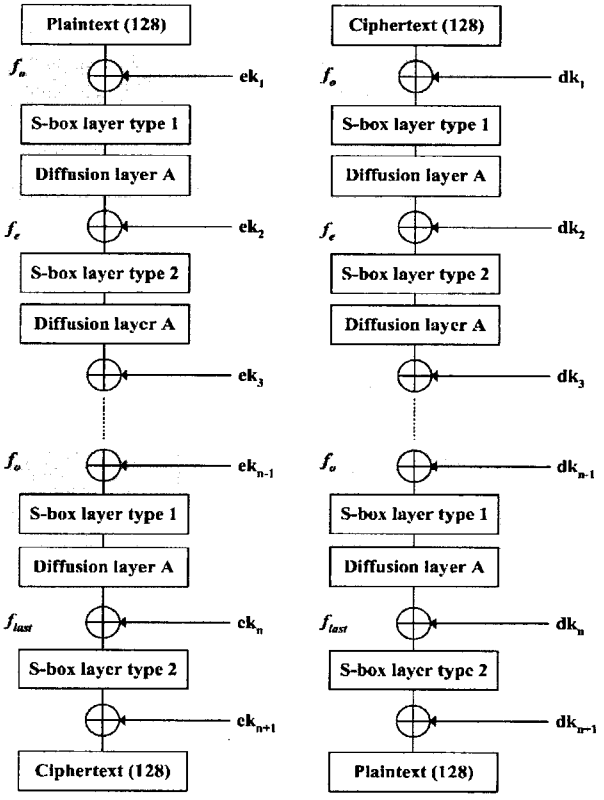


그림 1. 암호화 복호화 과정[4]

Fig. 1. Encryption and Decryption processes.[4]

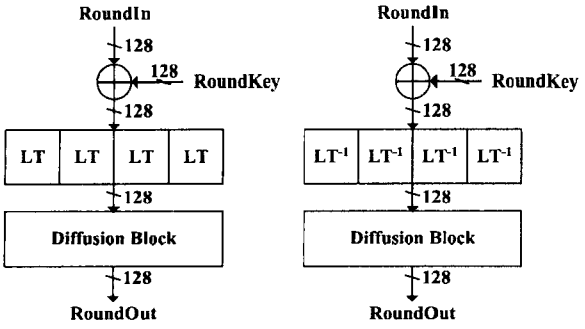


그림 2. 라운드 함수[1] (a) f_o (b) f_e
Fig. 2. Round function.[1] (a) f_o (b) f_e

그리고 확산 계층 (DiffLayer) 세 부분으로 구성되어 있다. 라운드 키 덧셈은 128 비트 라운드키를 라운드 입력 128 비트와 비트별 XOR한다. 치환 계층은 involution 구조를 위해 수정되었다. ARIA는 두 가지의 S-boxes S1, S2을 사용한 두 가지 유형의 치환계층 ((LS, LS, LS, LS), (LS^{-1}, LS^{-1}, LS^{-1}, LS^{-1})을 갖는다. 단, LS는 (S_1, S_2, S_1^{-1}, S_2^{-1})이다. 홀수 라운드 (f_o)의 치환 계층은 (LS, LS, LS, LS)이고, 짝수 라운드 (f_e)의 치환 계층은 (LS^{-1}, LS^{-1}, LS^{-1}, LS^{-1})이다. 라운드 함수는 그림 2와 같다. 확산 계층은 식 (1)과 같은 16x16 involution 이진 행렬을 사용한 바이트 간의 확산 함수로 구성되어 있다.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 0&0&1&1&1&0&0&1&0&0&1&0&0&1&0&1 \\ 0&0&1&1&0&1&1&0&0&0&0&1&1&0&1&0 \\ 1&1&0&0&0&1&1&0&1&0&0&0&0&1&0&1 \\ 1&1&0&0&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \bullet \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} \quad (1)$$

키 스케줄러는 초기화 과정과 라운드 키 생성 과정으로 나뉠 수 있다.초기화 과정에서는 3 라운드 256 비트 Feistel 알고리듬을 이용하여 마스터 키(MK)로 128 비트 값 W_0, W_1, W_2, W_3을 생성한다. 식 (2)과 같이 마스터 키는 KL과 KR로 입력되고 남은 비트는 0으로 채운다.

$$KL||KR = MK||0...0 \quad (2)$$

C_1 = 0x517cc1b727220a94fe13abe8fa9a6ee0
C_2 = 0x6db14acc9e21c820ff28b1d5ef5de2b0
C_3 = 0xdb92371d2126e9700324977504e8c90e

표 1. 초기화 상수[1]
Table 1. Initial constant.[1]

암호키 길이	CK_1	CK_2	CK_3
128-비트	C_1	C_2	C_3
192-비트	C_2	C_3	C_1
256-비트	C_3	C_1	C_2

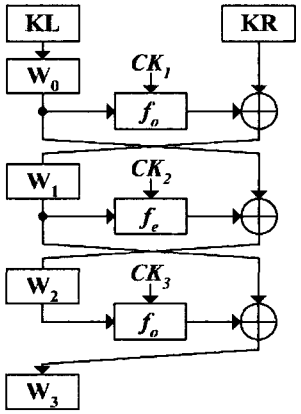


그림 3. 키 초기화 과정[1]
Fig. 3. Key initial processes.[1]

그림 3과 같이 KL과 KR은 라운드 함수의 fo와 fe 함수를 사용하여 Wi 값을 생성한다. Feistal 암호의 라운드 키 CKi는 π^{-1}의 유리수 부분의 128-비트 상수를 이용하여 암호키의 길이에 따라 표1과 같이 결정된다.

라운드 키 생성 블록은 네 개의 Wi 값을 조합하여 라운드 키를 생성한다. 키의 길이에 따라서 라운드 수가 10, 12 그리고 14 로 정의되어 있고 마지막 라운드에서는 KeyAddition을 두 번 하기 때문에, 라운드 키는 각각 11, 13 그리고 15 개를 생성한다. 복호화 할 때와 암호화 할 때의 라운드 키는 다르다. 복호화 라운드 키는 첫 번째와 마지막 라운드 키를 제외한 나머지의 라운드 키를 Diffusion Layer A 에 입력하여 그 결과를 사용한다. 복호화 라운드 키 생성 방법은 식 (3)과 같다.

$$\begin{aligned} dk_1 &= ek_{n+1},\ dk_2 = A(ek_n), \\ dk_3 &= A(ek_{n-1}),\ ..., \\ dk_n &= A(ek_2),\ dk_{n+1} = ek_1 \end{aligned} \quad (3)$$

2. 제안된 ARIA 구조

ARIA 알고리듬은 AES 알고리듬에 비하여 2 사이클의 연산을 더 수행해야 한다. 동일 클록으로 동작한다고 할 때, ARIA는 AES보다 17% 낮은 성능을 보인다. 동일한 성능을 위해서는 동작 주파수를 높일 수 있는 구조가 필요하다. 그림 2는 본 논문에서는 제안하는

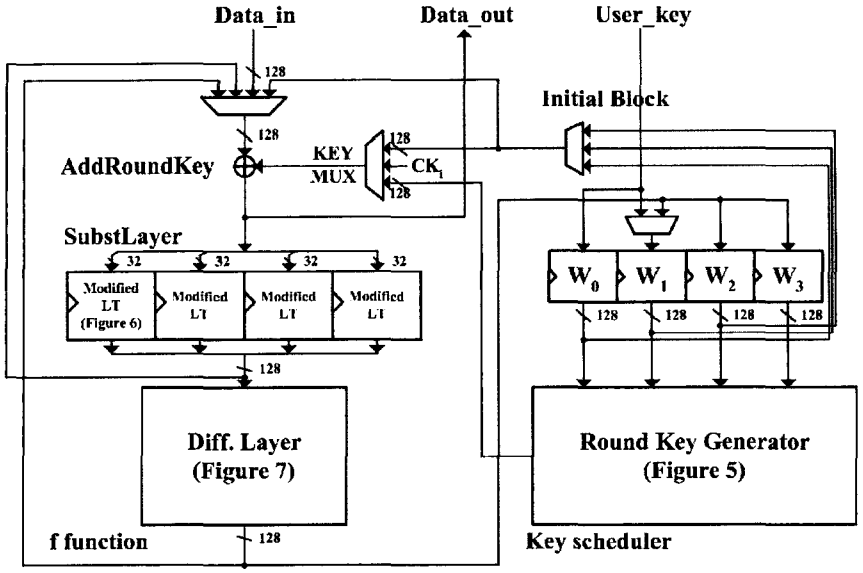


그림 4. ARIA의 블록도
Fig. 4. ARIA block diagram.

ARIA의 하드웨어 구조이다. 라운드 키 생성방식 중 on-the-fly 방식은 선 처리(pre-computation) 방식에 비해 라운드 키를 저장하는 레지스터가 필요치 않아 하드웨어의 크기를 줄일 수 있다.

제안된 키 스케줄러 블록은 초기화 블록과 라운드 키 생성 블록으로 나뉠 수 있다. 키 스케줄러를 초기화 블록은 Feistel 구조이다. Feistel 구조의 f 함수는 라운드 함수가 사용되고, EXOR 연산은 라운드 함수의 Addroundkey layer를 사용할 수 있도록 설계하였다. 그림 4의 KEY MUX는 초기화 블록의 Wi 값과 라운드 키 생성 블록의 라운드 키를 라운드 함수로 전달한다.

라운드 키 생성 블록은 on-the-fly 방식으로 키를 생성하기 위해 배럴 로테이터 (barrel rotator) 구조를 사용하였다. 배럴 로테이터는 곱셈기에서 고속 연산을 위해서 사용하는 배럴 쉬프터 (barrel shifter)를 수정하여 적용한 것이다. 이러한 배럴 로테이터는 한 사이클에 임의의 수만큼 회전이 가능하기 때문에 표 1과 같이 4 라운드를 주기로 매 라운드 마다 일정하게 회전하여 라운드 키를 생성하는 ARIA의 라운드 키 생성 블록에 적합하다. 표 1에서 '≫19' 는 19 만큼 오른 방향 회전을 의미한다.

라운드 키 생성은 4 라운드를 주기로 매 라운드 마다 Wi 의 선택이 바뀌고, 4 라운드 마다 회전하는 수가 바뀐다. 따라서 라운드 수로 회전수와 Wi 선택을 제어할 수 있다. 매 라운드 키를 생성하기 위한 제어신호를 표 2 와 같이 사용하였다. 제어신호는 그레이 코드를 수정한 것으로, 매 라운드마다 1 비트 변하고, 4 라운드를

표 1. 암호화 라운드 키[1]
Table 1. Encryption round key.[1]

$ek_{01} = (W_0) \oplus (W_1 \gg 19)$	$ek_{09} = (W_0) \oplus (W_1 \gg 67)$
$ek_{02} = (W_1) \oplus (W_2 \gg 19)$	$ek_{10} = (W_1) \oplus (W_2 \gg 67)$
$ek_{03} = (W_2) \oplus (W_3 \gg 19)$	$ek_{11} = (W_2) \oplus (W_3 \gg 67)$
$ek_{04} = (W_3) \oplus (W_0 \gg 19)$	$ek_{12} = (W_3) \oplus (W_0 \gg 67)$
$ek_{05} = (W_0) \oplus (W_1 \gg 31)$	$ek_{13} = (W_0) \oplus (W_1 \gg 97)$
$ek_{06} = (W_1) \oplus (W_2 \gg 31)$	$ek_{14} = (W_1) \oplus (W_2 \gg 97)$
$ek_{07} = (W_2) \oplus (W_3 \gg 31)$	$ek_{15} = (W_2) \oplus (W_3 \gg 97)$
$ek_{08} = (W_3) \oplus (W_0 \gg 31)$	$ek_{16} = (W_3) \oplus (W_0 \gg 97)$
	$ek_{17} = (W_0) \oplus (W_1 \gg 109)$

주기로 2 비트 변한다. 제어신호는 라운드 키생성 회로의 MUX와 배럴 로테이터의 선택 신호로 사용된다. 라운드 수의 세 번째, 네 번째 비트에 의해 생성된 배럴 로테이터의 decode 신호는 입력된 벡터의 회전이나 바이패스를 선택한다. 예를 들어, 1 ~ 4 라운드까지는 19 만큼 회전하므로 두 번째 단 이하에서는 바이패스가 선택되어서 최종 출력에서는 19 만큼 회전

된 값이 결정된다. 5 ~ 8 라운드에서는 31 번 회전해야 하므로 처음 19만큼 회전하고 두 번째 단에서 12만큼 더 회전된다. 세 번째 단 이하에서는 바이패스가 선택되므로 31만큼 회전한 값을 출력으로 얻을 수 있다. 본 논문에서는 FPGA로 설계하기 때문에 배럴 로테이터를 직접 구현하지 못하고 그림 6과 같이 4개의 MUX를 직렬로 연결하여 구현하였다. 배럴 로테이터의 구조로 구현한다면, 회로의 크기를 줄일 수 있고, 라운드 키 생성을 빠르게 할 수 있다.

LT 함수와 LT^{-1} 함수를 공유하기 위해 듀얼 포트

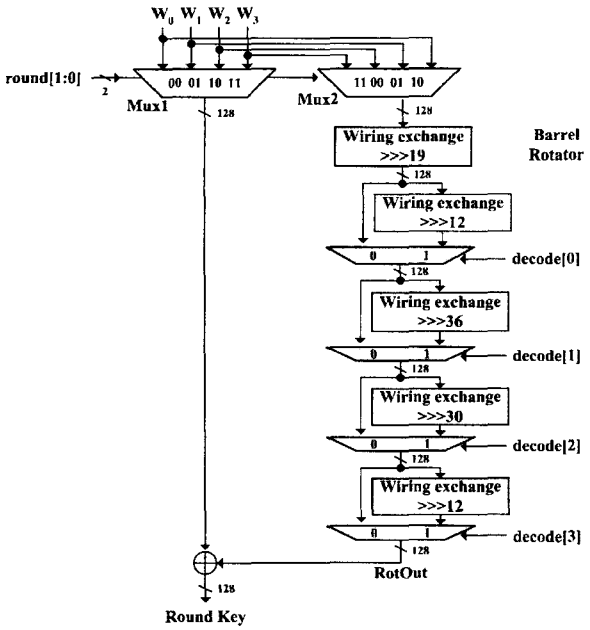
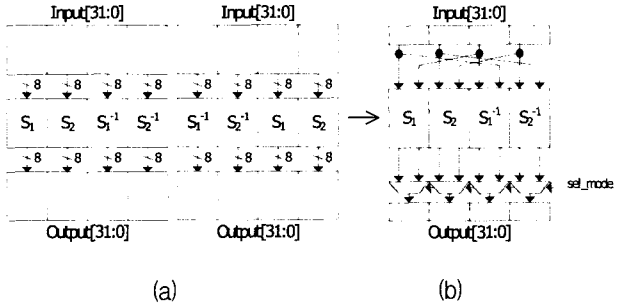


그림 5. FPGA에서 구현된 라운드 키 생성기
Fig. 5. Round key generator on FPGA.

표 2. Control signal table
Table 2. Control signal table.

Round	Control signal					decode [3:0]	Rotation number	Selected W_i	
	4	3	2	1	0			Mux1	Mux2
0	0	0	0	0	0	0000	19	W_0	W_1
1	0	0	0	0	1	0000	19	W_1	W_2
2	0	0	0	1	1	0000	19	W_2	W_3
3	0	0	0	1	0	0000	19	W_3	W_0
4	0	0	1	0	0	0100	31	W_0	W_1
5	0	0	1	0	1	0100	31	W_1	W_2
6	0	0	1	1	1	0100	31	W_2	W_3
7	0	0	1	1	0	0100	31	W_3	W_0
8	0	1	1	0	0	0110	67	W_0	W_1
9	0	1	1	0	1	0110	67	W_1	W_2
10	0	1	1	1	1	0110	67	W_2	W_3
11	0	1	1	1	0	0110	67	W_3	W_0
12	0	1	0	0	0	0111	97	W_0	W_1
13	0	1	0	0	1	0111	97	W_1	W_2
14	0	1	0	1	1	0111	97	W_2	W_3
15	0	1	0	1	0	0111	97	W_3	W_0
16	1	1	0	0	0	1111	109	W_0	W_1



(a) (b)

그림 6. (a) LT & LT^{-1} 블록; (b) 수정된 LT 블록
Fig. 6. (a) LT & LT^{-1} block; (b) Modified LT block.

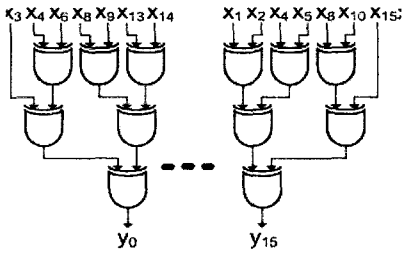


그림 7. 확장 블록
Fig. 7. Diffusion layer.

ROM을 채택하였다. 그림 6과 같이 LT 함수와 LT^{-1} 함수는 S-box의 배열순서만 바꿔줌으로써 쉽게 공유될 수 있다. 수정된 LT 블록의 구조는 그림 6b 와 같이 LT 함수를 듀얼 포트 ROM으로 구현하고 두 출력을 선택하는 멀티플렉서를 추가하였다. LT 함수와 LT^{-1} 함수를 구별하기 위해 sel_mode 신호를 사용하였다.

두 S-box는 설계의 용이성과 고속 처리를 위해 Lookup-table을 사용하였다. Diffusion Layer에서는 16×16 involution 이진 행렬을 그림 7과 같이 EXOR 게이트의 시리얼 연결로 구현하였다. 키 확장은 암/복호화와 독립적으로 동작하기 때문에, 키 확장 과정의 Feistel 구조의 f 함수를 round 함수를 공유할 수 있도록 설계하였다.

III. 성능 평가 및 영상 보안 시스템

본 장에서는 구현한 ARIA 블록의 효율성을 평가한다. 하드웨어 구현에서 효율성은 처리 속도 및 하드웨어의 면적으로 평가한다. 하드웨어 구조는 VHDL로 설계하였고, Xilinx VirtexE-1600 FPGA를 사용하여 검증하였다. ARIA 블록은 1 라운드 반복 구조로 구현하였고, 라운드 키 생성 블록의 배럴 로테이터는 FPGA에서 연속된 MUX 로 구현되었다. S-box는 Xilinx FPGA의 Block RAM으로 구현하였다. 설계된 ARIA 블록의 알고리듬 검증은 국가보안기술연구소에서 제공하는 테스트 벡터를 사용하였다[2].

ARIA 알고리듬은 AES 알고리듬에 비하여 2 사이클의 연산을 더 수행해야 한다. 동일 클록으로 동작한다고 할 때, ARIA는 AES보다 17% 낮은 성능을 보인다. 동일한 성능을 위해서는 동작 주파수를 높일 수 있는 구조가 필요하다. 제안한 ARIA 구조는 Addroundkey layer, Substitution layer 그리고 Diffusion layer를 포함한 1 라운드가 한 클록 사이클에 동작하는 구조로 제안되었다. 1 라운드를 통과하는 데는 약 21ns가 걸린다.

표 4. 성능 평가표
Table 4. Performance evaluation.

	Encryption/ Decryption	Device	Slices	BRAM	output cycle	Frequency (MHz)	Throughput (Mbps)	Throughput/Area (Mbps/slices)
AES[6]	Encryption	Virtex-E 3200	542	10	21	119	1,450	1.65
AES[7]	Encryption	Virtex-E 1000	1,857	-	1/10	125.38	1,604	0.867
	En/De	Virtex-E 1000	5,150	-	1/21	76	463.2	0.089
AES[8]	Encryption	XCV1000E-8	11,022	0	1	168.4	21,556	1.956
AES[9]	En/De	XC3S50-4	166	3	1/44	71.5	208	1.26
OurARIA	En/De	XCV1600E-8	1,491	16	1/12	46.5	496	0.33

ARIA는 AES에 비해 하드웨어 설계가 간단하다. Involution 함수로 이루어진 ARIA는 사용하는 키를 제외하면 암호화 과정과 복호화 과정이 동일하다. 따라서 복호화를 위한 회로가 추가로 필요한 AES에 비해서 작은 회로로 암/복호기를 제작할 수 있다. 제안된 ARIA에서는 회로 크기를 줄이기 위해 s-box에 듀얼 포트 ROM를 채택하였고, 라운드 키 생성기에 배럴 로테이터를 채택하였다. 설계한 ARIA는 1,491 SLICES의 크기로 496 Mbps의 성능을 보이고 있다.

표 4의 성능 평가표에서는 ARIA 회로와 AES 회로들과 면적과 성능을 비교하고 있다. AES[6]과 AES[8]은 암호기의 설계 구조만을 다루고 있기 때문에, 복호화 기능을 추가한다면 AES[7]과 같이 면적과 성능의 차이가 날것이다. 제안된 ARIA는 암/복호화 기능을 모두 가지고 있으면서 위와 같은 성능을 보이고 있다. AES[9]는 면적 대비 성능은 우수하더라고 고성능을 필요로 하는 어플리케이션보다는 소형의 임베디드 어플리케이션에 적당한 회로이다. 본 논문에서 제안된 구조보다는 AES[9]와 같은 32 비트 구조로 ARIA를 다시 설계해서 평가하는 것이 필요하다.

표 5와 같이 ARIA의 라운드 키 생성 불록은 BRAM을 제외한 전체 ARIA slices의 69% 인 1,031 slices이다. 4개의 128 비트 레지스터와 두 개의 4X1 MUX 그리고 배럴 로테이터 때문이다. FPGA에서는 배럴 로테이터가 MUX로 구현 되어 많은 회로 차지한다. 라운드 함수가 차지하는 비율이 낮기 때문에 라운드 함수에 레지스터를 추가하여 동작 주파수를 높이더라고 회로의 크기가 크게 커지지 않음을 예측할 수 있다. FPGA에서 연속된 MUX로 배럴 로테이터를 구현한 ARIA가 표 4와 같은 성능을 보인다는 것은 칩으로 설계하여 배럴 로테이터를 레이아웃 한다면 더욱 높은 성능과 적은 회로를 예측 할 수 있다. 동작 실측을 위한 칩 제작이 현재 진행 중이다.

ARIA 블록을 검증하기 위해서 영상 정보를 암호화

표 5. ARIA 설계의 블록 크기
Table 5. Block size of ARIA design.

Block		Size (slice)	Rate (%)
f function	Addroundkey	78	5.23
	SubstLayer	128	8.58
	DiffLayer	123	8.57
key scheduler	Initial block	384	25.7
	round key generator	647	43.3
controller		131	8.78
total		1,491	100

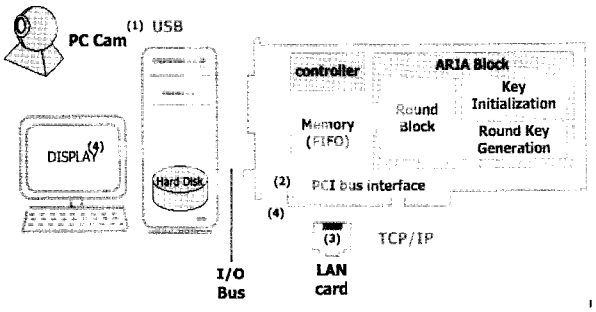


그림 7. 영상 보안 시스템의 블록도
Fig. 7. Block diagram of the video security system.

하여 인터넷 회선을 통해 주고 받을 수 있는 보안 애플리케이션인 영상 보안 시스템을 개발하였다. 시스템을 개발하기 위해 플랫폼 기반의 검증 툴인 Dynalith사의 iPROVE emulator를 사용하였다[11]. 두 컴퓨터의 통신은 TCP/IP를 사용하였고, 전송하는 패킷의 구조는 IPSec의 ESP (Encapsulation Security Payload)의 포맷을 사용하였다. 그림 7은 영상 보안 시스템의 블록도이다. 영상 보안 시스템의 데이터 흐름은 다음과 같다.

1. 사용자 A는 사용자 B와 통신을 연결하고, PC-Cam을 작동 시킨다.
2. 동영상 데이터를 USB를 통해 한 프레임씩 버퍼에 저장하고, PCI를 통해 ARIA 암호 블록으로 입력한다.
3. 암호화가 끝난 데이터는 사용자 B에서 인터넷으로

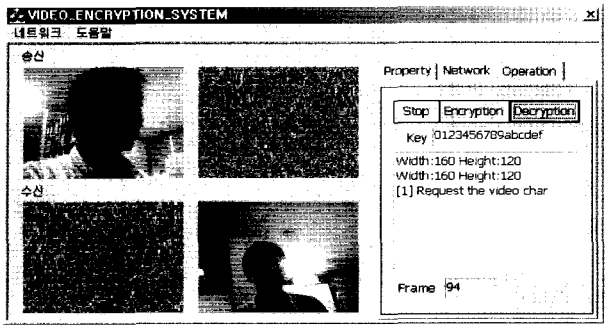


그림 8. 영상 보안 시스템 화면
Fig. 8. Screen capture of the video security system.

전송한다.

4. 사용자 B는 전송받은 데이터를 ARIA 암호 블록으로 복호화하여 화면에 출력한다.

640*480의 해상도를 갖는 영상을 초당 15 프레임으로 전송될 때 실시간으로 암호화하기 위한 대역폭이 약 110Mbps이다. 33MHz 32-비트 PCI 버스의 클록에서 ARIA의 처리율이 341Mbps이므로 실시간으로 처리에 충분하다. 그림 8은 영상 보안 시스템의 시연 장면을 포착한 것이다.

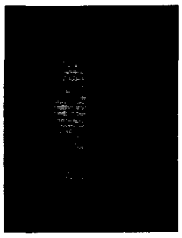
Ⅳ. 결 론

본 논문에서는 새로이 국내 표준으로 제정된 ARIA 암호 알고리듬의 하드웨어 구조를 처음으로 제안하고 있다. ARIA 알고리듬은 국가보안기술연구소에서 개발, 발표한 128-비트 블록 암호 알고리듬으로써, 안전성, 효율성 면에서 AES보다 나은 암호 알고리듬이다. ARIA의 Involution SPN 구조로써 하드웨어 설계할 때 AES 암호 알고리듬과 비교하여 장점을 가지고 있다. 본 논문에서는 위 장점을 최대한 활용하여 S-box를 공유할 수 있는 구조와 적은 면적에 빠른 키 생성을 위한 구조를 제안하였다. 설계된 ARIA 블록은 1,491 slices와 16 BRAM을 사용하여 496 Mbps의 성능을 보인다. ARIA 회로의 실제 환경에서의 검증을 위해 영상 보안 시스템을 개발하였고, 적용 결과 ARIA 블록은 실시간 영상 암호화(복호화)에 충분한 처리율을 내었다. 또한 이 구조는 IC 카드뿐만 아니라 데이터 저장이나 인터넷 보안 규격(IPSec, TLS)과 같은 고속 데이터 처리가 필요한 응용에 적합하다.

참 고 문 헌

[1] 국가보안기술연구소, ARIA 알고리듬 명세서, http://www.nsri.re.kr/ARIA, 2004.
[2] 국가보안기술연구소, ARIA 테스트 벡터, http://www.nsri.re.kr/ARIA, 2004.
[3] 국가보안기술연구소, Security and Performance Analysis of ARIA, http://www.nsri.re.kr/ARIA, 2003.
[4] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E-J. Yoon, S. Lee, J. Lee, S. Chee, D. Han and J. Hong, "New Block Cipher: ARIA," Proc. *ICISC2003,* pp. 432-445, Nov. 2003.
[5] H. W Kim and S. G. Lee, "Design and implementation of a private and public key crypto processor and its application to a security system," *IEEE Trans. on Consumer Electronics,* vol. 50, no. 1, pp. 214-224, Feb. 2004.
[6] F.-X. Standaert, G. Rouvroy, J.-J. Quisquater and J.-D. Legat, "A methodology to implement block ciphers in reconfigurable hardware and its application to fast and compact AES Rijndael," *Proc. FPGA'03,* pp. 216-224, Feb. 2003.
[7] S.-S. Wang and W.-S. Ni, "An efficient FPGA implementation of advanced encryption standard algorith," *Proc. ISCAS'04,* vol.2, pp. 597-600, May 2004.
[8] Xinmiao Zhang and Parhi, K.K., "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems,* vol. 12, pp. 957-967, Sept. 2004.
[9] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater and J.-D. Legat, "Compact and efficient encryption /decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications," *Proc. ITCC 2004.* vol. 2, pp. 583-587, April 2004.
[10] 홍진, 염용진, 권대성, "블록암호 ARIA 0.8의 제한적 8라운드 공격," *제 16회 정보보호와 암호에 대한 학술대회 논문집,* 122-129쪽, 천안, 대한민국, 2004년 9월
[11] Dynalith Systems Co., http://www.dynalith .com

저 자 소 개



박 진 섭(학생회원)
2004년 충북대학교 전기전자 공학부 학사 졸업.
2005년 현재 충북대학교 정보통신 공학과 석사과정 재학 중
<주관심분야 : 디지털 회로 설계, 암호시스템>

김 용 대(정회원)
1990년 충북대학교 정보통신 공학과 학사 졸업.
1993년 충북대학교 컴퓨터공학과 석사 졸업
1989년~1998년 신흥기술연구소 팀장
2000년~현재 충북대학교 정보통신공학과 박사과정
<주관심분야 : Computer arithmetic, ASIC 설계, 암호시스템>

장 태 주(정회원)
1982년 울산대학교 전기공학과 학사 졸업
1990년 한국과학기술원 전기 및 전자 공학과 석사 졸업
1998년 한국과학기술원 전기 및 전자공학과 공학박사 졸업
1982년~2000년 국방과학연구소 선임연구원
2000년~현재 한국전자통신연구원 부설 국가보안 기술연구소 책임연구원
<주관심분야: 암호프로세서 설계, 정보보호, 통계학적 신호처리>

윤 연 상(학생회원)
2004년 충북대학교 전기전자 공학부 학사 졸업.
2004년 현재 충북대학교 정보통신 공학과 석사과정 재학중
<주관심분야 : 디지털 회로 설계, 암호시스템>

양 상 운(정회원)
1992년 충북대학교 정보통신 공학과 학사 졸업.
1998년 충북대학교 정보통신 공학과 석사 졸업
1992년~2000년 국방과학연구소 연구원
2000년~현재 한국전자통신연구소 부설 국가보안기술연구소 선임연구원
<주관심분야 : 암호프로세서 설계, Computer Arithmetic, 정보보호, 반도체>

유 영 갑(정회원)
1975년 서강대학교 전자공학과 학사 졸업
1975년~1979년 국방과학연구소 연구원
1981년 Univ.of Michigan, Ann Arbor 전기전산학과 석사 졸업
1986년 Univ.of Michigan, Ann Arbor 전기전산학과 공학박사 졸업
1986년~1988년 금성반도체 (주) 책임 연구원
1993년~1994년 아리조나 대학교 객원교수
1998년~2000년 오레곤 주립대학교 교환교수
1988년~현재 충북대학교 정보통신공학과 교수
<주관심분야 : VLSI 설계 및 테스트, 고속 인쇄회로 설계, 암호학>