2005년도 대한전기학회 하계학술대회 논문집 2005. 7. 18 - 20

Ubiquitous Computing 환경에 적합한 ARIA 알고리즘 암호화 프로세서의 설계

노경호 , *고광철
한양대학교 전자통신컴퓨터공학과 , *한양대학교 전기제어생체공학부

Design of an ARIA Crypto-processor for the Ubiquitous Computing Enviroment

Kyung-Ho Roh, Kwang-Chul Ko
Hanyang University Dept. of Electronics and Computer Engineering

Abstract - Ubiquitous Computing 환경은 모든 사물과 공간이 지능화되어 사용자가 컴퓨터나 네트워크를 의식하지 않는 상태에서 장소에 구애받지 않고 자유롭게 네트워크에 접속할 수 있는 환경을 의미한다. 만일 이러한 환경 속에서 개인정보들이 노출되었을 경우는 법적, 사회적, 경제적으로 커다란 손실을 초래하게 된다. 이것을 방지하기 위해서는 안정성과 효율성이 높은 암호 알고리즘이 요구된다. 본 논문에서는 한국 표준으로 제정된 ISPN(Involutional SPN) 구조의 블록 암호화 ARIA 알고리즘을 사용하여 고속의 통신망과 Smart Card, PDA, 이동전화 및 다양한 기기 등의 사용이 보편화될 Ubiquitous Computing 환경에 응용 가능한 ARIA 암호화 프로세서(이하 ARIA 프로세서)를 설계하였다.

1. 서 론

암호 알고리즘은 암호키(key)와 복호키가 동일한가에 따라 대칭형(비밀키) 암호 알고리즘과 비대칭형(공개키) 암호 알고리즘으로 구분된다. 대칭형 암호 알고리즘의 대표적인 예로는 미국의 차세대 암호 표준 AES(Advanced Encryption Standard)인 Rijndael 알고리즘과 일본의 Camellia 알고리즘이 있으며 우리나라에는 SEED 알고리즘과 ARIA 알고리즘이 있다.

ARIA는 우리나라의 학·연·관이 공동으로 개발한 정보보호의 핵심 기술이다. ARIA는 민간 암호화 알고리즘인 SEED와 함께 전자정부의 대국민행정서비스용으로 보급되고 있으며, Smart Card 등의 초경량 환경 및 고성능 서버 환경 등에서 SEED에 비하여 상대적인 장점을 가지고 있다.

본 논문에서는 ARIA 알고리즘과 연산 기법을 간단히 기술하였으며, ARIA 암호 알고리즘의 하드웨어 구현을 위한 효율적인 구조를 제안하고 이를 Verilog-HDL로 모델링 하여 설계한 후, FPGA에 구현하여 동작을 검증하였다.

2. 본 론

2.1 ARIA 알고리즘

ARIA는 SPN(Substitution-Permutation Network)구조의 128bit 블록 암호이며, 128bit 192bit, 256bit의 3종류의 암호키 사용을 제공한다. ARIA의 입·출력 문자의 크기와 사용 가능한 암호키의 크기는 미국에 표준 블록 암호인 AES의 입·출력 크기 및 사용 가능한 암호키의 크기와 동일하다. 블록 암호는 대표적인 공격방법인 차분공격(differential cryptanalysis), 선형공격(linear cryptanalysis), 부정 차분공격(truncated differential crypt analysis), 불능 차분 공격(impossible differential cryptanalysis), 고계 차분 공격(higher order differential cryptanalysis), square 공격 등의 다양한 공격 방법에 대하여 안전하여야 한다. ARIA는 이러한 공격 방법들에 대하여 6라운드 이상에서 2^{-128}보다 큰 확률을 가지는 어떠한 특성도 존재하지 않으며, 이는 ARIA가 여러 공격 방법들에 안전함을 의미한다. ARIA는 미국의 AES와는 달리 128bit 전체 블록을 처리하는 확산 단계(diffusion layer)를 사용한다. AES의 확산 단계인 Mix Column 연산은 32bit 블록(전체 블록의 1/4)을 처리하며, Camellia의 확산 단계는 64bit 블록(전체 블록의 1/2)을 처리한다. ARIA의 이러한 특성은 collision 공격 partial sum 공격, 부정 차분 공격에 대하여 AES와 Camellia 보다 강한 안정성을 제공한다.

2.2 알고리즘의 구조

ARIA 알고리즘의 구조는 암호화와 복호화 과정이 같은 ISPN(Involutional SPN)구조이며 다음과 같은 특징을 갖는다.

구분	입출력 크기	입력키 블록크기	라운드 수
ARIA-128	128bit	128bit	12
ARIA-192	128bit	192bit	14
ARIA-256	128bit	256bit	16

표 1. ARIA 알로리즘의 특징

ARIA의 라운드 함수는 다음과 같은 세 부분으로 구성되어 있다.

1. 라운드 키 덧셈(AddRoundKey): 128bit 라운드키를 라운드 입력 128bit와 bit별 XOR한다.
2. 치환 계층(SubstLayer): 두 유형의 치환 계층이 있으며 각각은 2종의 8bit 입출력 S-box 와 그들의 역변환으로 구성된다.
3. 확산 계층(DiffLayer): 간단한 16x16 involution 이진 행렬을 사용한 byte간의 확산 함수로 구성되어있다.

암호화 과정과 복호화 과정은 [그림 1]와 같이 주어지며 암호화 과정과 복호화 과정은 라운드 키를 제외하고는 일치한다.

2.2.1 치환계층

ARIA의 치환계층은 8bit 입출력 S-Box들로 구성된다. S-box들은 다음의 성질을 만족하도록 선택되었다.

- 최대 차분/선형 확률이 2^{-6}
- 대수적 차수 7
- 고정점과 반정점이 없을 것

이러한 성질을 만족하는 것들로는 유한체 $GF(2^8)$상의 함수 x^{-1}에 행렬 곱과 벡터 합이 순차적으로 구성된 아핀 변환을 취한 형태가 널리 사용되고 있다. x^{-1} 와 특

성이 같은 것으로 x^{-2^n},n=0,...,7이 있는데 ARIA에서는 x^{-1}와 x^{247}에 아핀 변환을 취하여 S-box를 생성하였다. S_1, S_2는 다음의 꼴을 가진다.

- 두 유형 모두 S-box S_1, S_2, S_1^{-1}, S_2^{-1} 로 구성
- 두 유형의 치환계층은 서로 역의 관계
- 32bit 단위로 4종의 S-box 사용[2]

두 유형의 치환계층은 교대로 사용되어 involution 구조인 확산 계층과 함께 전체 구조가 involution 구조가 되도록 한다.

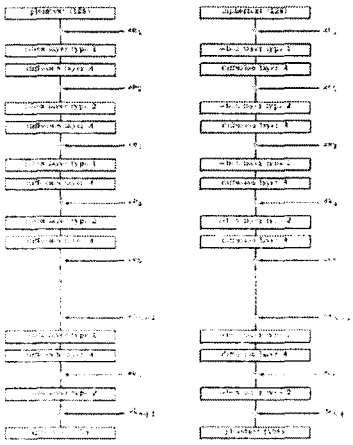

그림 1. 암호화 복호화 과정

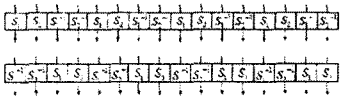

그림 2. 두 유형의 치환계층

2.2.2 확산계층

확산 계층은 ARIA와 다른 블록 암호를 구별 짓는 주요 부분으로 16x16 involution 이진 행렬을 사용한다. 확산 함수는 입력 16byte에 대하여 byte 단위의 행렬 곱을 수행한 결과의 16byte를 출력으로 한다.

ARIA의 확산함수 A : $GF(2^8)^{16} \rightarrow GF(2^8)^{16}$는 입력을 $(x_0, x_1, \cdots, x_{15})$라 하고 출력을 $(y_0, y_1, \cdots, y_{15})$라 하면, [그림 3]의 행렬로 표현된다.

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&0&1&0&0&0&0&1&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix}$$

그림 3. 확산함수의 행렬표현

2.2.3 키확장

ARIA의 키 확장은 초기화 과정과 라운드키 생성 과정의 두 부분으로 나뉜다. 초기화 과정에서는 암호화/복호화 한 라운드를 F 함수로 하는 256bit 입·출력 3라운드 Feistel 암호를 이용하여 암호키 MK로부터 네 개의 128bit 값 W_0, W_1, W_2, W_3를 생성한다. 암호키 MK의 길이는 128, 192 또는 256이므로 위 Feistel 암호의 입력에 필요한 256bit(KL,KR)을 다음과 같이 구성한다.

첫째, 128bit KL은 MK의 상위 128bit를 취한다.

둘째, MK의 남은 bit를 이용하여 KR의 상위 bit를 채우고 나머지는 0으로 채운다.(KL||KR ← MK|| 0-0.)

셋째, F_o와 F_e를 각각 홀수, 짝수 라운드 함수라고 할 때 다음과 같이 W_0, W_1, W_2, W_3를 생성한다.

$$W_0 = KL \qquad W_1 = F_o(W_0, CK_1) \oplus KR$$

$$W_2 = F_e(W_1, CK_2) \oplus W_0 \qquad W_3 = F_o(W_2, CK_3) \oplus W_1$$

Feistel 암호의 128bit 라운드키 CK_i는 π^{-1}의 유리수 부분의 128bit 상수를 이용하여 암호키의 길이에 따라 다음과 같이 결정된다.

암호키 길이	CK_1	CK_2	CK_3
128bit	C1	C2	C3
192bit	C2	C3	C1
256bit	C3	C1	C2

C1 = 0x517cc1b727220a94fe13abe8fa9a6ee0
C2 = 0xf6db14acc9e21c820ff28b1d5ef5de2b0
C3 = 0xdb92371d2126e9700324977504e8c90e

라운드키 생성 과정에서는 네 개의 128bit W_0, W_1, W_2, W_3 를 조합하여 암호화 라운드키 ek_i와 복호화 라운드키 dk_i 를 생성한다. 라운드 수는 암호키의 크기가 128, 192, 256bit인 경우 각각 12, 14, 16라운드이고 마지막 라운드에는 키 덧셈 계층이 두 번 있으므로 각각 13,15,17개의 라운드키를 생성해야 한다. 암호화 라운드키는 다음과 같다.

$ek_1 = (W_0) \oplus (W_1^{\gg 19})$ $ek_2 = (W_1) \oplus (W_2^{\gg 19})$ $ek_3 = (W_2) \oplus (W_3^{\gg 19})$
$ek_4 = (W_0^{\gg 19}) \oplus (W_3)$ $ek_5 = (W_0) \oplus (W_1^{\gg 31})$ $ek_6 = (W_1) \oplus (W_2^{\gg 31})$
$ek_7 = (W_2) \oplus (W_3^{\gg 31})$ $ek_8 = (W_0^{\gg 31}) \oplus (W_3)$ $ek_9 = (W_0) \oplus (W_1^{\ll 61})$
$ek_{10} = (W_1) \oplus (W_2^{\ll 61})$ $ek_{11} = (W_2) \oplus (W_3^{\ll 61})$ $ek_{12} = (W_0^{\ll 61}) \oplus (W_3)$
$ek_{13} = (W_0) \oplus (W_1^{\ll 31})$ $ek_{14} = (W_1) \oplus (W_2^{\ll 31})$ $ek_{15} = (W_2) \oplus (W_3^{\ll 31})$
$ek_{16} = (W_3^{\ll 31}) \oplus (W_3)$ $ek_{17} = (W_0) \oplus (W_1^{\ll 19})$

복호화 라운드키는 암호화 라운드키와 다르며 암호화 라운드키로부터 유도된다. 먼저 키의 순서가 바뀌고 처음과 마지막 라운드키를 제외하고 암호키를 입력으로 하는 확산 함수 A의 출력이 복호화 라운드키가 된다. 라운드 수가 n 일 때, 복호화 라운드키는 다음과 같다.

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

2.3 회로 설계

본 논문에서는 128bit의 입출력 크기와 128, 192, 256bit의 암호키의 사용이 가능한 ARIA 암호화 알고리즘의 특징 중에서 128bit의 입·출력 크기와 128bit 키를 사용할 수 있는 ARIA 암호 프로세서를 설계하였다.

2.3.1 아키텍처의 개요

ARIA 프로세서는 암호화 라운드키를 생성하는 Keygen 모듈, 복호화 라운드키를 생성하는 Dekeygen 모듈, 암호화 및 복호화를 담당하는 Ariacore 모듈, 각 모듈의 동작을 중앙에서 컨트롤하는 Ariactrl 모듈을 가

지고 있다.

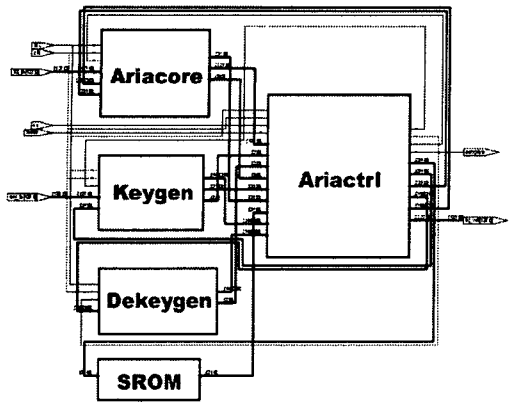

그림 4. ARIA 프로세서

2.3.2 암호화 과정

암호화 과정을 보면 암호 키(MK)와 평문(Plaintext)을 ARIA 프로세서에 넣고 시작신호를 인가하면 암호키가 Keygen 모듈을 통과하면서 W_0, W_1, W_2, W_3 로 변환되고 암호화 라운드키 $ek_1, ek_2, \ldots, ek_n$ 이 출력으로 나온다. Ariactrl 모듈은 생성된 라운드 암호화 키 ek들과 평문을 함께 Ariacore 모듈에 입력하며 Ariacore 모듈을 통과한 신호는 복문(CIPHERTEXT)으로 변환되어 나온다.

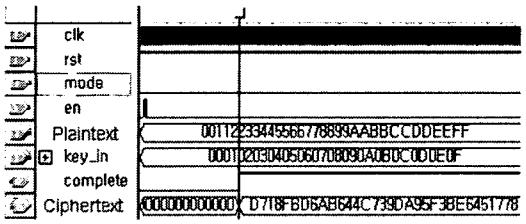

그림 5. 암호화 결과

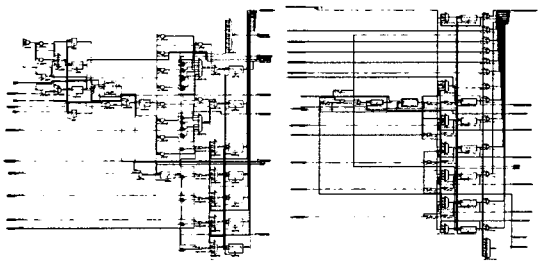

그림 6. Keygen 모듈

그림 7. Dekeygen 모듈

2.3.3 복호화 과정

복호화 과정은 암호화 과정에서 사용한 것과 동일한 암호키(MK)와 복문(Ciphertext)을 ARIA 프로세서에 넣고 시작신호를 인가하면 암호키가 Keygen 모듈을 통과하면서 W_0, W_1, W_2, W_3 로 변환되고 암호화 라운드 키 $ek_1, ek_2, \ldots, ek_n$ 이 출력으로 나온다. Ariactrl 모듈은 생성된 라운드 암호화 키 ek들을 Dekeygen 모듈에 입력시키고 Dekeygen 모듈을 통과한 ek들은 복호화 라운드 키 dk가 된다. Ariactrl 모듈은 새롭게 생성된 dk신호들과 복문을 함께 Ariacore 모듈에 입력하며 Ariacore 모듈을 통과한 신호는 평문(Plaintext)으로 변환되어 나온다.

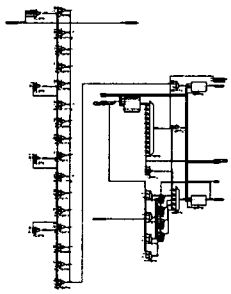

그림8. Ariacore 모듈

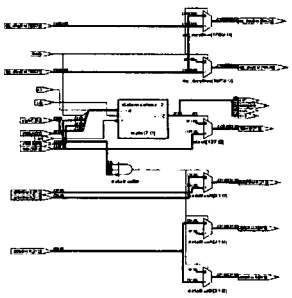

그림 9. Ariactrl 모듈

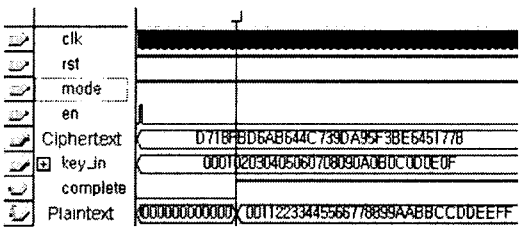

그림 10. 복호화 결과

3. 결 론

본 논문에서 설계한 ARIA 알고리즘 프로세서는 먼저 Verilog-HDL 로 설계하여 각각의 암호 알고리즘을 C언어로 모델링 된 결과의 test vector와 일치한지를 검증하였다. 그리고 설계한 회로를 Synplify를 사용하여 vertex2pro xc2vp30을 타깃 디바이스로 합성하였다. 합성결과는 최대 동작 주파수는 100Mhz이었으며, 사용된 총 LUT는 12793(43%)이었다.

하드웨어로 구현된 한국의 암호화 기술인 ARIA는 Ubiquitous Computing 환경으로 접어 들어가면서 날로 중요해 지고 있는 암호화 분야에서 빠른 성능과 높은 보안성을 만족하여 네트워크, 모바일 통신, 전자상거래 시스템과 다양한 종류의 Ubiquitous Computing 환경을 이루는 단말기에 사용하기에 적합할 것으로 판단한다. 앞으로 192bit, 256bit 까지 암호화가 가능하도록 설계를 추가하고 임베디드 프로세서에 ARIA 프로세서를 연결시킨 후 Coprocessor로 동작하도록 추가할 계획이다.

[참 고 문 헌]

[1] 강주성 외, "현대암호학", 2002년
[2] ARIA개발팀, "블록암호 알고리즘 ARIA",WISC,2003년
[3] 한국정보보호학회,"현대 암호학 및 응용", 2002년
[4] http://www.nsri.re.kr
[5] 한국전자통신연구원,"암호학의 기초", 1999년