論文
3-9

변형된 Feistel 구조를 이용한 Circle-g의 설계와 분석 (A Design and Analysis of the Block Cipher Circle-g Using the Modified Feistel Structure)

임웅택(Lim, Ung-Taeg)[1] 전문석(Moon-Seog Moon)[2]

요 약

본 논문에서는 18-라운드 변형된 Feistel 구조를 갖는 128-bit 블록 암호알고리즘 Circle-g를 설계하고, 차분공격(differential cryptanalysis)과 선형공격(linear cryptanalysis)을 통해 안정성을 분석하였다. Circle-g의 f-함수는 2-라운드만에 완전 확산(diffusion)이 일어나도록 설계되었다. 이러한 우수한 확산효과로 인해 9-라운드 차분특성이 구성될 확률은 2^{-144}로, 12-라운드 선형특성이 구성될 확률은 2^{-141}로 분석되었다. 결과적으로 Circle-g는 128-bit 비밀키를 적용하였을 경우 12-라운드 이상이면 차분공격이나 선형공격은 전수(exhaustive)조사 방법보다 효율성이 떨어지는 것으로 분석되었다.

Abstract

In this paper, we designed a 128-bits block cipher, Circle-g, which has 18-rounds modified Feistel structure and analyzed its secureness by the differential cryptanalysis and linear cryptanalysis. We could have full diffusion effect from the two rounds of the Circle-g. Because of the strong diffusion effect of the F-function of the algorithm, we could get a 9-rounds DC characteristic with probability 2^{-144} and a 12-rounds LC characteristic with probability 2^{-141}. For the Circle-g with 128-bit key, there is no shortcut attack, which is more efficient than the exhaustive key search, for more than 12 rounds of the algorithm.

키워드 : DC(differential cryptanalysis), LC(linear cryptanalysis), NIST, AES, DES, Rijndael

1) 정회원 : 부천대학 전산정보처리과 교수
2) 정회원 : 숭실대학교 컴퓨터학과 교수

논문접수 : 2004. 3. 20.
심사완료 : 2004. 3. 27.

1. 서 론

NIST(National Institute of Standards & Technology)는 DES(data encryption standard)를 대체할 차세대 암호 알고리즘 표준안인 AES(Advanced Encryption Standard)를 공모하여, 2000년 10월 2일 Rijndael[6]을 최종 선정하였다.

NIST는 AES 공모 시 차세대 암호 알고리즘은 다음과 같은 요구사항[1]을 만족하도록 요구하였다.

- 대칭키 블록 암호 알고리즘
- 3중 DES보다 안전할 것
- 키 길이는 128, 192, 256 비트까지 가능할 것
- 블록 크기는 128 비트
- 하드웨어와 소프트웨어로 구현 가능할 것

본 논문에서는 AES 요구사항을 만족하는 독자적인 암호 알고리즘을 설계하고, 차분공격(differential cryptanalysis)[2]과 선형공격(linear cryptanalysis)[3] 분석을 통해 안정성을 입증한다. 이 논문에서 제시한 암호 알고리즘은 Circle-g라 명명하였다.

DES와 같이 Feistel 구조로 설계된 암호 알고리즘이 보통 3-라운드를 거쳐야만 완전 확산(diffusion)이 이루어지는 것과 달리 Circle-g는 단지 2-라운드 만에 완전 확산이 일어난다. 이러한 큰 확산 효과로 인하여 Circle-g는 128-bit 비밀키를 적용하였을 경우 9-라운드 이상이면 차분이나 선형 공격 방법이 전수조사 방법보다 효율성이 떨어지는 것으로 분석되었다.

본 논문에서는 256-bit 비밀키를 적용하였을 경우를 고려하여 총 18-라운드 구조를 갖는 Circle-g를 제안하였다.

본 논문의 구성은 다음과 같이 이루어졌다. 2장에서는 제안된 암호알고리즘의 암호화와 복호화 구조를 설명하였고, 3장에서는 각 구성 부분에 대한 설계특성을 설명하였다. 4장에서는 기본 라운드가 갖는 차분특성과 선형특성 분석을 통해 축소된 3 라운드에 대한 최대 공격 복잡도를 계산하였다. 마지막으로 5장에서는 간단하게 결론을 내린다.

2. Circle-g 설계

2.1 설계 목표

설계한 암호 알고리즘은 DES와 같은 전통적인 Feistel 구조를 갖는다.

일반적으로 Feistel 구조는 비선형 변환을 하는 substitution 부분과 확산 효과를 극대화하는 선형 변환 부분, 라운드 키를 적용하는 부분으로 구성된다.

Circle-g 설계는 AES에서 요구하는 기준을 따랐으며, 구체적으로 다음 두 가지에 중점을 두고 설계하였다.

- 차분공격과 선형공격에 강할 것
- 확산 특성이 좋을 것

2.2 Circle-g 구조

설계된 Circle-g의 전체적인 구조는 [그림 1]과 같다.

Circle-g는 DES와 유사한 18-라운드 Feistel 구조로 설계되었으며 첫 라운드 시작 전과 마지막 라운드 후에 이미 효과가 입증된 XOR whitening 기능을 적용하였다. whitening은 공격자에게 첫 라운드와 마지막 라운드의 f-함수 입력 값을 예측하지 못하게 하는데 효과가 있는 것으로 [4]에서 이미 입증된 방식이다.

Circle-g의 입력 블록 크기는 32-bit 4개 워드로 이루어진 128-bit로서 왼쪽 2개 워드는 각 라운드에서 f-함수로 입력되고 오른 쪽 2개 워드는 f-함수의 출력과 XOR하여 좌측과 우측 두 워드를 상호 교환하는 구조를 갖는다.

각 라운드에 입력되는 4개 워드를 각각 $L_{r,0}$, $L_{r,1}$, $R_{r,0}$, $R_{r,1}$라 하고, f-함수의 출력을 (F_0, F_1)라고 했을 때 r 라운드의 출력 워드와 (r+1) 라운드의 입력 워드와의 관계식은 다음과 같다.

$$L_{r+1,0} = F_{r,0} \oplus R_{r,0}$$
$$L_{r+1,1} = F_{r,1} \oplus R_{r,1}$$
$$R_{r-1,0} = L_{r,0}$$
$$R_{r-1,1} = L_{r,1}$$

2.3 f-함수

f-함수는 substitution 효과를 위한 s-box 적용 부분과 32-bit 라운드 키를 s-box의 출력에 더해주는 키 적용 부분, 그리고 확산 효과 극대화를 위한 g-함수 부분으로 구성된다. 여기에서 g-함수는 substitution과 라운드 키 적용된 두 워드 간에 상호 간섭 효과를 주도록 하는 부분으로서 두 워드 간의 XOR 후 최대 확산 효과를 갖는 MixCol(maximum diffusion transformation)를 적용하는 구조를 갖는다.

f-함수 구조는 다음과 같은 식으로 표현할 수 있다.

$$T_0 = (S(L_0) + K_{2r}) \bmod 2^{32}$$
$$T_1 = (S(L_1) + K_{2r-1}) \bmod 2^{32}$$
$$F_0 = \mathrm{MixCol}(T_0 \oplus \mathrm{ROR}(g(T_1), 8))$$
$$F_1 = \mathrm{MixCol}(T_1 \oplus \mathrm{ROR}(T_0, 8))$$

2.4 s-box

Circle-g의 s-box는 256개의 8-bit 요소들로 구성되며, 8-bit 입력 index 값으로 출력 8-bit 값을 결정한다.

2개 워드를 입력으로 하는 f-함수는 총 8번 s-box를 적용하게 되는데 Circle-g에서는 1개의 s-box 만으로 운영한다. 이는 g-함수가 우수한 확산 특성을 가지고 있기 때문에 가능하고 알고리즘을 하드웨어로 구성 시 메모리 사용을 최소화하기 위한 방안이다.

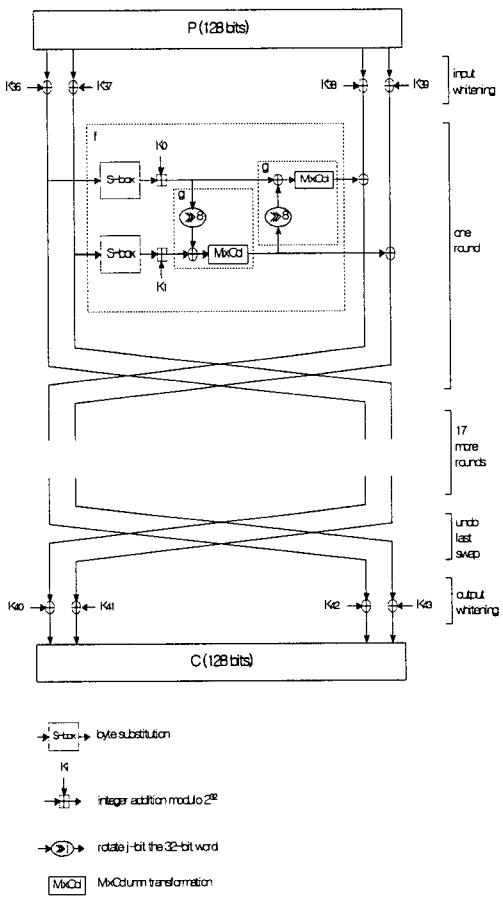
설계된 s-box는 부록에 명시하였다.

2.5 g-함수

g-함수는 확산 효과를 극대화하기 위한 부분으로서 각 라운드에서 입력 워드의 각 bit 간에 간섭 효과를 최대화하기 위해 MixCol를 적용하였다. 한 워드에서의 확산 효과를 다른 워드에도 확산시키기 위해서 MixCol에 입력되는 워드는 인접 워드와 XOR하여 적용한다. 인접 워드와의 XOR는 인접 워드를 rotate right 1-byte한 후에 적용하는데 이것은 특정 위치의 byte가 다른 위치의 byte에도 영향을 미치게 하기 위해서이다.

f-함수에 포함된 두 g-함수를 각각 g_0, g_1라고 했을 때 g-함수를 식으로 표현하면 다음과 같다.

$$g_0 = \mathrm{MixCol}(T_0 \oplus \mathrm{ROR}(g_1, 8))$$
$$g_1 = \mathrm{MixCol}(T_1 \oplus \mathrm{ROR}(T_0, 8))$$



[그림 1] Circle-g 라운드 구조

2.6 MixCol

MixCol 연산은 한 워드를 구성하고 있는 각 바이트 간에 확산을 극대화시키기 위한 연산으로서 Rijndael[6]에서 적용하고 있는 MixColumn 연산 방법과 동일하게 적용하고 있다. 단, MixCol 연산에서 사용되는 MDS(maximum distance separable) 행렬은 하드웨어 구현이 용이하도록 재설계 하였다.

HiCUBE에서 적용하고 있는 MDS 행렬과 MixCol 연산 방법은 식 (1)과 같다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 01 & 01 & 46 & C8 \\ 46 & 01 & C8 & 01 \\ C8 & 46 & 01 & 01 \\ 01 & C8 & 01 & 46 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$y_i = \bigoplus_{j=0}^{3} (m_{ij} \cdot x_j) \quad (1)$$

식 (1)에서 m_{ij}는 MDS 행렬의 i행 j열의 값이고, $m_{ij} \cdot x_j$ 연산은 GF(2^8) 상의 원시다항식 $p(x)=x^8+x^4+x^3+x^2+1$을 이용한 다항식 곱 연산이다.

여기에서 MDS 테이블의 요소는 16진법으로 표기한 수치이다.

2.7 WMT

WMT(word mix transposition)는 각 라운드를 거치면서 바이트 간 확산 효과를 극대화시키기 위해 워드 단위로 행과 열을 바꾸어주는 과정이다.

WMT 연산은 아래 그림과 같다.

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

WMT ⇒

a_{00}	a_{10}	a_{20}	a_{30}
a_{01}	a_{11}	a_{21}	a_{31}
a_{02}	a_{12}	a_{22}	a_{32}
a_{03}	a_{13}	a_{23}	a_{33}

WMT를 두 번 적용하면 원래 위치로 돌아오므로 WMT에 대한 역행렬이 필요가 없다.

2.8 키 스케줄

키 스케줄은 18-라운드에서 사용할 라운드 키 36개 워드와 whitening용 8개를 포함하여 총 44개 워드를 생성한다. 키 스케줄 구조는 크게 초기 키 확장 부분과 라운드 키 생성 부분으로 구성된다.

키 확장 부분은 입력되는 가변길이 비밀키를 입력받아 정해진 가변 키 길이(128, 192, 256 bit)보다 작을 경우 해당 길이만큼 '0'으로 채우고, 확장된 비밀키의 bit간에 간섭 효과를 확산시켜주는 역할을 한다.

라운드 키 생성 부분은 확장된 키를 입력으로 하여 whitening 키와 라운드 키를 생성하는 부분이다.

2.8.1 키 확장

키 확장은 128-bit 또는 192-bit, 256-bit로 입력되는 비밀키를 라운드 키 생성에서 요구되는 256-bit 길이로 키 길이를 확장하는 부분이다. 만약, 입력되는 비밀키의 길이가 요구되는 키 길이(128-bit 또는 192-bit, 256-bit)보다 짧을 경우 부족한 길이만큼 '0'으로 채워서 키 확장의 입력으로 사용한다.

비밀키는 키 길이에 따라 다음과 같이 워드 길이로 표현된다.

$K_{s128} = (K_{s0}, K_{s1}, K_{s2}, K_{s3})$

$K_{s192} = (K_{s0}, K_{s1}, K_{s2}, K_{s3}, K_{s4}, K_{s5})$

$K_{s256} = (K_{s0}, K_{s1}, K_{s2}, K_{s3}, K_{s4}, K_{s5}, K_{s6}, K_{s7})$

키 확장은 입력 bit간에 확산 효과를 최대화될 수 있도록 MixCol를 적용하였다. MixCol의

적용은 입력되는 비밀키의 잉여부분이 '0'으로 채워짐으로서 서브키 생성에 부정적인 영향을 미칠 수 있는 요소를 제거해 주는 역할을 하게 된다.

키 확장에 적용되는 4-워드 MixCol는 4x4 MixCol 테이블과 입력되는 4x4-워드 비밀키에 대한 다항식 곱 연산으로 이루어진다.

4-워드 MixCol는 다음과 같은 연산을 수행한다.

$$\begin{Bmatrix} K_{m0} \\ K_{m1} \\ K_{m2} \\ K_{m3} \end{Bmatrix} = \begin{Bmatrix} 01 & 01 & 46 & C8 \\ 46 & 01 & C8 & 01 \\ C8 & 46 & 01 & 01 \\ 01 & C8 & 01 & 46 \end{Bmatrix} \cdot \begin{Bmatrix} K_{s0} \\ K_{s1} \\ K_{s2} \\ K_{s3} \end{Bmatrix}$$

키 확장 부분으로 입력되는 비밀키는 키 길이에 따라 다음과 같은 연산으로 확장된다.

$$\begin{Bmatrix} K_{m0} \\ K_{m1} \\ K_{m2} \\ K_{m3} \end{Bmatrix} = \{MDT\} \cdot \begin{Bmatrix} K_{s0} \\ K_{s1} \\ K_{s2} \\ K_{s3} \end{Bmatrix} \quad \text{if kw=4} \quad \begin{Bmatrix} K_{m4} \\ K_{m5} \\ K_{m6} \\ K_{m7} \end{Bmatrix} = \{MDT\} \cdot \begin{Bmatrix} K_{m0} \\ K_{m1} \\ K_{m2} \\ K_{m3} \end{Bmatrix}$$

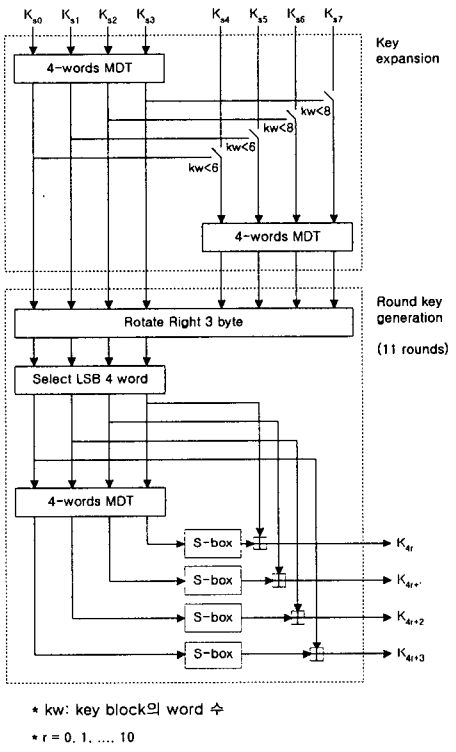
$$\text{if kw=6} \quad \begin{Bmatrix} K_{m4} \\ K_{m5} \\ K_{m6} \\ K_{m7} \end{Bmatrix} = \{MDT\} \cdot \begin{Bmatrix} K_{s4} \\ K_{s5} \\ K_{m2} \\ K_{m3} \end{Bmatrix} \quad \text{if kw=8} \quad \begin{Bmatrix} K_{m4} \\ K_{m5} \\ K_{m6} \\ K_{m7} \end{Bmatrix} = \{MDT\} \cdot \begin{Bmatrix} K_{s4} \\ K_{s5} \\ K_{s6} \\ K_{s7} \end{Bmatrix}$$

여기에서, kw는 입력되는 비밀키의 워드 수이다.

2.8.2 서브키 생성

Circle-g에서 필요로 하는 총 44개의 서브키는 한 라운드에 4개의 서브키를 생성하는 서브키 생성 라운드를 11회 반복 수행시킴으로서 얻어진다.

서브키 생성 라운드 구조는 먼저, 확장된 비밀키를 매 라운드마다 rotate right 3 byte를 하여 확장된 비밀키를 골고루 사용토록 하며, rotate right 3 byte된 결과 중에 최하위 4개 워드를 선택하여 4-워드 MixCol를 적용한 후 각각 s-box의 입력으로 하고, 각 s-box의 출력에 4-워드 MixCol가 적용되기 전의 각 워드 값을 modulo 합 결과를 서브키로 생성한다.



〔그림 2〕 키 스케줄 구조

3. 설계 특성

3.1 s-box 특성

s-box에 의한 치환(substitution) 수행은 비선형 변환과정으로서 암호 알고리즘의 차분공격[2]과 선형공격[3] 특성을 결정하는 부분이다.

Circle-g의 s-box는 Rijndael[6]에서의 설계 방식[12]을 적용하여 다음과 같은 절차로 설계하였다.

s-box 설계는 $GF(2^8)$ 상의 원시다항식 $p(x)=x^8+x^4+x^3+x^2+1$를 적용하여 다음과 같은 방법으로 설계하였다.

① $a(x) \cdot b(x) \ mod \ p(x)=1$ 관계가 성립하는 a의 *inverse* 값 b를 계산

② a의 *inverse* 값 b에 대하여 아래 연산을 적용

$$\begin{bmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 1&1&1&1&1&0&0&0 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&0&0&0&1&1&1&1 \\ 0&0&0&1&1&1&1&1 \\ 0&0&1&1&1&1&1&0 \\ 0&1&1&1&1&1&0&0 \end{bmatrix} \cdot \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

이와 같은 방법으로 설계된 s-box는 부록에 명시하였다.

설계된 s-box에 대한 차분특성과 선형특성은 다음과 같다.

- 최대 차분 특성 확률: 2^{-6}
- 최대 선형근사 *bias*: 2^{-4}

3.2 s-box 적용 후 라운드 키 연산 효과

s-box 적용 후 라운드 키를 modulo 합 연산을 하는 것은 고정된 s-box 테이블의 각 요소에 라운드 키를 modulo 합 연산하여 s-box 테이블을 재구성한 후에 재구성된 s-box를 적용한 것과 동일한 결과를 갖는다. 이것은 다시 말해서 가능한 라운드 키 256개만큼의 가변 s-box를 운영하는 것과 동일한 효과를 갖게 된다.

결국, 라운드 키에 따라 각기 다른 XOR 특성을 가지므로 공격자가 높은 확률로 차분특성이나 선형특성을 구성하기가 곤란하게 된다.

따라서 공격자는 차분공격이나 선형공격 시 s-box의 출력 XOR 값과 라운드 키를 적용한 출력 XOR값을 같게 하는 특성을 이용하는 것이 효과적일 수밖에 없게 된다.

3.3 MixCol 특성

어떤 고정된 4X4 byte 테이블 값과 입력 4X1 byte의 다항식 곱 연산 결과는 대단히 높은 확산효과를 갖는 점이 [6]와 [7]에서 입증되었다.

Circle-g에서 사용하는 MixCol 테이블의 확산 효과에 대한 두 특성은 다음과 같다.

특성-1: MixCol 입력 워드 a의 4-byte 중에 3-byte를 모두 '0'으로 하고, 나머지 1-byte를 0x0이 아닌 값으로 했을 때 입력 값과 출력 값에 대한 최소 hamming distance 값은 6이다.

특성-2: Branch Number는 5이다.

Branch Number는 선형 변환에 대하여 확산 효과를 측정하는 한 방법으로서 선형변환의 입력과 출력에 나타나는 non-zero byte의 최소의 수로 측정된다. W(a)를 MixCol에 입력되는 워드 a의 non-zero byte의 수라고 정의했을 때 branch number는 식 (2)로 표현할 수 있다.

$$\mathrm{Min}_{a \neq 0}(W(a) + W(MixCol(a))) \quad (2)$$

MixCol 변환에서 사용되는 MDS(maximum distance separable) 행렬은 입력 워드의 각 바이트 간에 확산 효과를 극대화시키는 4x4 행렬 테이블이다.

MDS 행렬의 설계는 [8]에서 적용한 설계 방식에 따라 $GF(2^8)$ 상의 원시다항식 $p(x)=x^8+x^4+x^3+x^2+1$을 이용하여 설계하였다.

설계된 MDS 행렬은 다음과 같다.

$$MDS = \begin{bmatrix} 01 & 01 & 46 & C8 \\ 46 & 01 & C8 & 01 \\ C8 & 46 & 01 & 01 \\ 01 & C8 & 01 & 46 \end{bmatrix}$$

MDS 행렬의 요소 값들은 16진법으로 표현하였다.

MDS 행렬 요소 중에 46과 C8은 다항식 곱 연산을 하드웨어로 구성할 때 테이블 참조 방식이 아닌 쉬프트만으로 연산이 가능하도록 각각 아래와 같이 원시다항식 $p(x)$로부터 계산된 값이다.

$$0x46 = p(x)/4 + 1$$
$$0xC8 = p(x)/4 + p(x)/2 + 1$$

여기에서 $p(x)/2$는 $p(x)$를 LSB(lowest significant bit) 쪽으로 1 비트 쉬프트하고 $p(x)/4$는 2 비트 쉬프트 하는 하드웨어적 연산이 가능하다.

만약, 8비트 길이를 갖는 입력 $a(x)$를 다음과 같이 다항식으로 표현하였을 때

$$a(x)=a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x^1+a_0$$

$46 \cdot a(x)$와 $C8 \cdot a(x)$ 연산은 다음과 같이 $a(x)$의 최하위 두 비트 a_0 또는 a_1의 값에 따른 쉬프트 연산과 XOR만으로 연산이 가능해진다.

$$46 \cdot a(x)=a(x)>>2 \oplus a(x) \oplus (a_0==1 \ ? \ p(x)>>2 : 0) \oplus (a_1==1? \ p(x)>>1 : 0)$$
$$C8 \cdot a(x)=a(x)>>2 \oplus a(x)>>1 \oplus a(x) \oplus (a_0==1? \ p(x)>>2 : 0) \oplus (a_0==1? \ p(x)>>1 : 0) \oplus (a_1==1? \ p(x)>>1 : 0)$$

여기에서, a_0와 a_1는 $a(x)$의 최하위 첫 번째와 두 번째 비트를 의미하며, $(a_0==1 \ ? \ p(x)>>2 : 0)$ 식은 $a_0=1$이면 $p(x)$를 우측으로 두 비트 쉬프트 한 값을 반환하고, $a_0=0$이면 0을 반환하는 비교 연산이다.

따라서 MixCol 연산을 쉬프트 연산과 XOR 연산만으로 운영이 가능하여 하드웨어 구성이 용이하다.

MDS 행렬의 구성요소 중에 0x01은 MixCol 연산을 반으로 줄이기 위해 선택한 값이다. 즉, 어떤 입력 요소에 1을 곱하면 자신의 값이 되므로 연산을 생략할 수 있다.

3.4 Key addition 차분특성

f-함수에서의 두 라운드 키 적용은 s-box 출력 32-bit와 라운드 키 32-bit를 더하여 mod 2^{32}를 취함으로서 이루어진다.

이와 같은 워드에 대한 modulo 합 연산은 하위 byte에서 발생되는 carry 값의 유무에 따라 인접 byte 값에 영향을 주게 되므로 결과적으로 라운드 키는 s-box의 출력 값을 숨기는 역할뿐만 아니라 입력의 각 bit가 여러 출력 bit에 확산효과를 주는 결과를 갖게 된다.

또한, 라운드 키 적용을 XOR 연산으로 수행할 경우는 입력 쌍의 XOR 값은 출력 쌍의 XOR 값과 동일한 반면 라운드 키의 modulo 합 연산은 항상 동일한 XOR 값을 갖지 않는다. 라운드 키의 modulo 합 연산 시 입력 쌍의 XOR 값이 출력 쌍의 XOR 값과 동일해질 확률은 입력 XOR 값의 hamming weight에 따라 다르게 나타난다.

다음은 s-box의 두 출력 쌍에 대한 출력 XOR 값 ΔSo에 대한 hamming weight을 W(ΔSo)로 표현했을 때 출력 XOR 값이 동일해질 확률은 다음과 같은 식으로 표현할 수 있다.

$$DP^a=2^{-W(\Delta So)}, \qquad 0 \le DP^a \le 1 \quad (3)$$

이와 같은 라운드 키 합에 대한 XOR 특성은 차분공격이나 선형공격 시 의도하는 라운드 특성 구성이 불가능할 수 있음을 알 수 있다.

3.5 Key Addition 선형특성

[9]에서는 $X \boxplus Y \rightarrow Z$연산에 대한 선형근사 *bias*를 식 (4)와 같이 정의하고 있다.

$$bias \quad \ge \quad 2^{-1-\lfloor h/2 \rfloor} \quad (4)$$

식 (4)에서 h는 $(M_1 \cdot X) \oplus (M_2 \cdot Y) \oplus (M_3 \cdot Z)$의 선형근사를 위한 masking 값 M_1또는 M_2, M_3에 대한 hamming-weight 값이다.

식 (4)에 의하면 $x \boxplus y$ 연산에 대한 선형근사 *bias*는 최소한 2^{-2}이고, *modulo* 덧셈 연산은 선형공격을 어렵게 함을 알 수 있다.

3.7 라운드 수

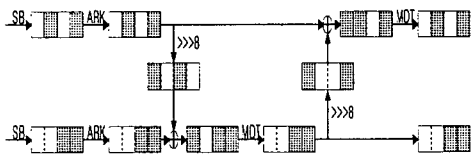
일반적으로 암호 알고리즘의 총 라운드 수는 전수조사보다 효율적인 shortcut 공격에 안전한 라운드 수에 보다 나은 안전성을 위해 잉

여 라운드를 추가하여 결정한다.

Circle-g의 9-라운드 차분특성은 2^{-144}확률로 12-라운드 선형특성은 2^{-141} 확률로 구성되므로 Circle-g에 대한 차분공격은 128-bit의 비밀키를 적용했을 경우 12-라운드 이상이면 전수조사 방법보다 효과가 없음을 알 수 있다. 그러나 Circle-g는 동일한 암호 알고리즘으로 128에서 256 bit까지의 가변 비밀키를 적용해야 함으로 Circle-g의 총 라운드 수는 256-bit 비밀키를 적용하였을 경우 shortcut 공격이 전수조사보다 효율적이지 못한 라운드 수로 결정된다.

따라서 다음과 같은 결론을 얻을 수 있다.

• DES는 완전 확산이 3-라운드 후에나 가능하지만 Circle-g는 2-라운드만에 이루어지므로 Circle-g의 총 라운드 수는 16라운드 이상이면 된다.



〔그림 3〕 f-함수 차분특성

• 18-라운드의 차분특성 구성 확률은 2^{-288}이므로 256-bit 비밀키를 고려할 때 Circle-g의 총 라운드 수는 18-라운드이면 적당하다.

4. 공격 분석

4.1 차분공격

한 라운드에 대한 차분특성이 구성될 확률은 비선형 특성을 갖는 s-box 치환 연산과 덧셈 연산의 차분 특성 확률에 의해 결정된다.

한 라운드에 대한 차분 특성이 구성될 최대 확률 P_{max}는 아래와 같이 정의할 수 있다.

$$P_{max} = (DP^{s}_{max})^{\#As} \quad (5)$$

DP^{s}_{max} : s-box의 최대 차분특성 확률

$\#A_s$: s-box의 입력 active byte 수

식 (5)에서 key addition에 대한 차분특성 확률이 고려되지 않은 것은 식 (3)에서와 같이 DP^{σ}_{max}=1이기 때문이다.

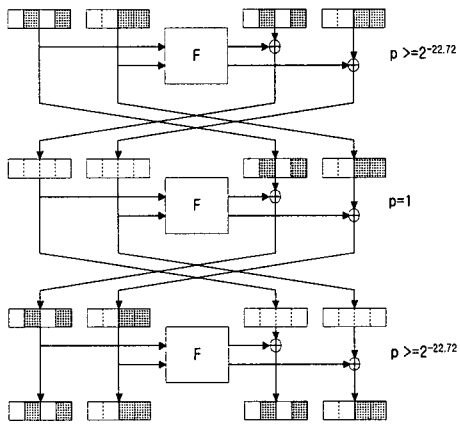
암호 알고리즘에 대한 차분공격은 확률 높은 1-라운드 특성 구성으로부터 시작된다.

[그림 3]은 [그림 4]의 첫 라운드 f-함수에 대한 세부적인 active byte에 분석 그림이다.

[그림 4]에서 ▩는 차분 값이 0이 아닌 active byte를 의미하고, 우측에 나타낸 블록은 각 라운드의 출력에 대한 active byte를 나타낸 것이다.

[그림 4]에서 s-box 입력에 대한 active byte의 수는 모두 8개이므로 3-라운드 차분특성이 구성될 최대 확률은 2^{-48}이 된다.

따라서 9-라운드 특성은 3-라운드 특성을 3번 적용하면 되므로 2^{-144} 확률로 구성된다. 결국 Circle-g에 대한 차분공격은 128-bit의 비밀키를 적용했을 경우 9-라운드 이상이면 전수조사 방법보다 효과가 없음을 알 수 있다.



〔그림 4〕 3-라운드 특성

4.2 선형공격

전체 라운드에 대한 선형공격 특성은 각 연산의 선형근사 *bias* 값으로부터 한 라운드 또는 전체 라운드의 *bias* 값을 계산하는 공식 (6)에 의해서 구한다. 이 식은 [5]의 Piling-up Lemma 식을 이용한 *bias* 계산식이다.

$$b = 2^{n-1}\prod_{i=0}^{n} b_i \quad (6)$$

식 (6)에서 b_i는 각 연산에 대한 *bias* 값으로서 식 (4)에 의해서 구해지고, b는 n개로 구성된 연산에 대한 *bias* 값이다.

따라서 f-함수에 대한 선형근사 *bias*는 다음과 같은 공식으로 계산할 수 있다.

$$b_i = 2 \cdot (b_s)^{\#A_s} \cdot (2^{-1-\lfloor \#A_a/2 \rfloor}) \quad (7)$$

식 (6)에서 $\#A_s$는 active s-box의 수이고, $\#A_a$는 modulo 덧셈이나 뺄셈 연산에 대한 선형공격을 위해 masking하는 값에 대한 active bit 수이다. b_s는 s-box의 최대 선형근사 *bias* 값으로서 Circle-g의 s-box는 2^{-4}이다. 식 (7)에서 $\#A_s$=Min($\#A_a$) 관계가 성립한다.

[그림 4]에서 $\#A_s$는 8이므로 3-라운드 선형특성이 구성될 확률은 2^{-36}으로 계산된다. 9-라운드 특성은 2^{-106} 확률로 12-라운드 특성은 2^{-141} 확률로 구성된다.

5. 결론

Circle-g에서 MixCol 연산은 branch number가 5로서 높은 확산 효과 특성을 갖는다.

MixCol연산의 높은 확산 효과는 차분이나 선형특성 구성 시 f-함수의 출력에 non-zero byte 수를 증가시키므로 결과적으로 특성 구성 확률을 낮추는 효과가 있다.

g-함수는 이웃한 워드와 XOR 연산 처리를 한 후에 MixCol 연산을 수행하게 되므로 g-함수에 입력되는 한 byte의 값은 MixCol 연산을 통해 나머지 다른 모든 byte들에 영향을 미치게 되므로 Circle-g는 단지 2-라운드만에 좌, 우 입력 워드들 간에 모두 영향을 미치는 높은 확산 능력을 갖는다. 이에 반해, DES의 경우는 적어도 3-라운드를 거쳐야만 입력 한 byte가 다른 byte에 영향을 모두 미친다.

Circle-g는 이러한 우수한 확산효과로 인해 3-라운드 차분특성이 구성될 확률이 2^{-48}이고, 9-라운드 특성은 2^{-144} 확률로 구성된다. 또한, 3-라운드 선형특성이 구성될 확률이 2^{-36}이고, 9-라운드 특성은 2^{-106}, 12-라운드 특성은 2^{-141} 확률로 구성된다.

결국, Circle-g는 128-bit 비밀키를 적용하였을 경우 12-라운드 이상이면 차분이나 선형공격은 전수조사 방법보다 효율성이 떨어진다.

참 고 문 헌

[1] National Institute of Standards & Technology, "Announcing Development of a Federal Information Standard for Advanced Encryption Standard," Federal Register, v.62, 1997

[2] E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer-Verlag, 1993.

[3] M. Matsui, "Linear Cryptanalysis of DES Cipher (I)", Symposium on Cryptography and Information Security '93, 1993

[4] J.Kilian and P.Robshaw, "How to Protect DES Against Exhaustive Key Search," Advances in Cryptology - EUCROCRYPT'96, Springer-Verlag, 1996

[5] A.F.Webster and S.E.Tavares, "On the Design of S-boxes," Proc. of CRYPTO'85, Springer-Verlag, 1985.

[6] J.Daemen and V.Rijmen, "The Rijndael Block Cipher," AES Proposal, 1999

[7] B.Schncier, J.Kelsey, D.Whiting, D.Wagner, C.Hall, N.Ferguson, "Twofish: A 128-Bit Block Cipher," AES Proposal, 1998

[8] Alexis Warner Machado. "Differential

Probability of Modular Addition with a Constant Operand", April 2001.

[9] C. Burwick and D. Coppersmith, "MARS-a candidate cipher for AES," IBM Corperation, 1999.

부 록

```
unsigned int s-box[256] = {
 50,  12,  88, 126,   7,  80,  20, 132, 221, 249,
118,  63,  33, 103, 105,  74,
 176, 229, 162, 175, 101,  99, 180, 254, 206,  52,
152,  45, 234, 111,  14, 120,
 115,  82, 217,  96, 122, 218, 252, 160, 153,  65,
154,  48, 113, 158,  84,  15,
  76, 226,  49, 128,  18, 224, 189,  43,  94, 137,
233,  37,  89,  59,  23,  91,
 146,  21, 119, 135, 178, 200, 110, 202,  22, 220,
70, 164,  85, 192, 123, 140,
 231, 183, 139,  36,  19,  92,  51,  68, 147, 156,
17,  56, 116,  71, 172, 131,
  13, 228,  47, 237, 198,  90, 107, 138,  34,  26,
46,  75, 245, 108, 190, 106,
   4, 127, 239, 142, 170,  53, 204,  86, 242,  35,
182, 186, 213, 149, 243, 143,
  98, 166, 212, 230, 144, 171, 157,   9, 114,  40,
79, 194,  28, 181,  78,  64,
  32,  10,  69, 246, 125,   0, 121,  87, 129,  72,
62,  54, 227, 253,  24, 150,
 216,  81, 133,  38, 238, 222,  57, 244, 215, 165,
5,  31, 199, 195, 124, 179,
 151, 163,  16, 201, 214, 250,  66, 161, 100, 130,
136,  42,   8,  67, 159, 141,
 173, 145,  44, 208, 188, 248, 168, 187,  61,  60,
6, 211, 235, 174,  27, 155,
  58, 167,  83,   2,  73, 255, 251, 197, 209, 219,
29, 104,   1, 191,  30, 240,
  41,  93, 225, 134, 169,  55,  25, 184,  11,  97,
196, 241,  77, 203, 117, 247,
  39, 223, 207, 232, 112, 102,   3, 109, 193, 185,
148,  95, 210, 205, 236, 177,
};
```

임 웅 택

1985년 : 금오공과대학 전자공학과 졸업
1992년 : 국방대학원 전자계산과 석사
1997년~현재 : 부천대학 전산정보처리과 부교수
1998년~현재 : 숭실대학교 컴퓨터학과 박사과정 수료
<관심분야> 정보보호, 블록 암호 설계 및 분석

전 문 석

1981년 : 숭실대학교 전자계산학과 졸업
1986년 : University of Maryland. Computer Science(석사)
1989년 : University of Maryland. Computer Science(박사)
1989년 : Morgan State Univ. 부설 Physical Science Lab. 책임연구원
1989년~1991년 : New Mexico State University 부설 Physical Science Lab. 책임연구원
1991년~현재 : 숭실대학교 컴퓨터학과 정교수
<관심분야> PKI, 침입탐지시스템, 인증시스템, 전자상거래보안, 병렬처리시스템