한국군사과학기술학회지 제14권 제2호, pp. 238~245, 2011년 4월

학술논문 정보기술 부문

DES 키 확장을 이용한 S Box 재설계에 관한 연구

A Study on a S Box Redesign using DES Key Expansion

이 준*
Jun Lee

Abstract

We suggest a DES key expansion algorithm which is strong enough to overcome Differential Cryptanalysis(DC) and Linear Cryptanalysis(LC). Checking the weak points of DES, we found that the opened S box provide all information on the various kinds of attack. Using the key expansion we redesigned the S box which is not open to anybody who has no key. DC and LC can not be applied to the suggested algorithm without the redesigned S box information. With the computer experiments we show that the efficiency of this algorithm is almost the same as that of DES with respect to the crypto speed.

Keywords : DES, 키 확장, S Box, Differential Cryptanalysis, Linear Cryptanalysis

1. 서 론

현대 암호 알고리즘은 Kerckhoffs 원칙[1]으로 요약된다. 키를 제외한 모든 알고리즘을 공개하여 그 안전성을 과학적으로 검증하기 위함이다. 검증 과정에서 사용되는 평문과 암호문의 조건은 기지 평문 공격 혹은 선택 평문 공격이 가능하다고 가정한다. DES는 암호화 알고리즘이 공개되어 표준화로 보급된 최초의 블록암호이다. 또한 현재까지 가장 많이 보급되었으며 세계에서 가장 많이 사용하는 블록 암호이다. 하지만 DES는 키 길이가 56비트로 설정되어 키 공간이 원천적으로 작아 컴퓨터 성능 발달과 더불어 전수 공격(Exhaustive Attack)의 취약함이 있다. 또한 S box 연구 결과 차분 공격(Differential Cryptanalysis)과 선형 공격(Linear Cryptanalysis)은 전수 공격보다 훨씬 적은 경우의 수 공격으로 키 예측을 가능하게 한다. 선형 공격 실험 결과 DES-12는 2^{43}개의 기지 평문으로 50시간에 암호화 키 56비트를 구하는데 성공하였다[2]. 금융과 같이 정보 보호가 요구되는 분야는 triple DES(3DES)와 같이 키 길이를 확장하는 알고리즘을 제시하여 암호의 안전도를 높이기 위한 노력을 하였으나 DES 알고리즘을 세 번 반복하는 임시적인 방법이다. 본 연구는 기본적인 DES 알고리즘에 확장된 키를 이용하여 S box를 재설계함으로 선형 공격과 차분 공격에 강한 블록 암호 알고리즘을 제안한다.

† 2011년 1월 21일 접수~2011년 3월 25일 게재승인
* 공군사관학교 (Korea Air Force Academy)
책임저자 : 이 준(jlee@afa.ac.kr)

2. 용어 정의

- **Key 비트**는 $k_0k_1k_2\cdots k_{n-1}$로 표시한다. **Key의 부분키**

는 Key의 일부분으로 구성된 연속적인 비트열 $k_n k_{n+1} k_{n+2} \cdots k_{n+j-1} k_{n+j}$이며 $\prod_{i=n}^{n+j} k_i$로 표시

- Sa는 8개로 구성된 DES의 S box에서 a번째 상자이며 Sa(n)는 Sa의 n번째 행
- DCa(α, β)는 Sa의 입력 차분 α에 대한 출력 차분이 β가 되는 경우의 수
- NSa(α, β)는 Sa의 입력 64개 중에서 α로 마스크된 입력 비트와 β로 마스크 된 출력 비트의 XOR 값이 일치되는 경우의 수

3. 관련연구

가. 차분 공격

평문 쌍의 입력차분과 그에 대응하는 암호문 쌍의 출력차분 사이의 관계를 통계적으로 분석하여 키를 추측하는 선택 평문 공격이다. 입력 차분을 갖는 평문 쌍을 선택하여 대응되는 암호문 쌍을 획득하고, 획득한 암호문 쌍을 이용하여 키를 추측한다. S box 분석과 라운드 특성을 이용하여 전수 공격보다 훨씬 더 적은 경우의 수로 키를 추측할 수 있게 한다. Biham과 Shamir 연구[3]에서 DES의 S box는 신중하게 설계되었으나, 차분 공격에 취약한 요인은 S box를 포함한 암호 구조에 있다고 하였다. 차분 공격에 견딜 수 있는 다양한 S box 재설계에 대한 연구결과가 제시[4,5]되었지만 취약점[6]이 지적되었다. 결론적으로 차분 공격에 대한 DES의 안전은 S box 재설계로는 부족하며 암호 함수 자체를 수정하는 것이다.

나. 선형 공격

Matsui에 의해 연구[7]된 결과로 주어진 암호 알고리즘에 대하여 높은 확률로 만족하는 평문과 그에 대응하는 암호문 사이의 선형 관계식을 이용하여 키를 복구하는 기지 평문 공격이다. 선형 관계식을 조사한 후 임의의 평문, 암호문 쌍에 대하여 관계식을 적용하여 키를 추측한다. 선형 공격은 8개의 S box에 대한 입력과 출력의 선형관계로부터 시작된다. 계산된 입출력 비트 사이에 상관관계가 많이 있다고 생각되는 모든 S box 중에서 편차가 가장 큰 것을 이용하여 암호함수의 최량 표현을 구한 후 키 비트를 찾는다. 공격 대책은 S box에서 입력과 출력사이 선형관계를 높은 확률로 추측가능하게 하는 편차 정보를 차단하는 것이다.

4. DES 취약점

컴퓨터 계산 능력의 향상, 차분 공격, 선형 공격에 대한 연구 결과는 다음과 같은 DES의 취약점을 제시한다.

가. 키 축소

DES 키는 64비트가 입력되나 그 중 8비트를 사용하지 않고 56비트로 각 라운드에서 사용될 부분키를 생산한다. 키 비트 하나의 축소는 전수 공격의 수를 반으로 줄이므로 8비트 축소는 키 공간을 1/256으로 축소한다. DES 알고리즘은 각 라운드에 사용되는 부분 키의 각 비트에 대해 키 위치가 공개된다. 다양한 공격 방법은 전수 공격보다 적은 수의 경우로 키 추측을 가능하게 한다. 키 공간 축소를 가능하게 하는 암호 해독 기술과 컴퓨터 계산 능력의 발달은 DES의 안전에 결정적인 취약 요소가 된다.

나. S box 공개

차분 공격과 선형 공격은 S box 분석에서 시작된다. 이에 따라 다양한 방법의 S box 재설계를 연구하였지만 타당한 결과를 얻지 못하였다. S box가 공개되는 알고리즘은 차분 공격과 선형 공격 대상이 된다. DES의 취약점은 S box가 공개됨에 있다. S box가 공개될 경우 공격에 필요한 분석이 가능하다. 키로부터 S box가 생성될 수 있다면 키를 가진 송신측과 수신측만 재설계 된 S box를 공유할 수 있다. 키가 없는 공격자는 S box를 확보할 수 없어 차분 공격과 선형 공격에 필요한 자료를 얻을 수 없다. 따라서 S box를 재설계하여 공개하는 것은 근본적인 대책이 될 수 없고, 키로부터 S box를 재설계하여 자격자만이 소유하는 방법이 공격에 대한 근본적인 대책이 될 것이다.

5. 제안 알고리즘

확장키로부터 재설계 된 S box를 S box 재구성이라 정의하고, 관련 알고리즘을 다음과 같이 제안한다. S box 재구성에 필요한 확장키를 S_{key}라 하자

가. S_{key} 구성

S box 재구성을 위한 키 S_{key}는 128비트로 구성되며 각 4비트 단위로 구분하여 32개의 부분키를 만든

다. S box 재구성에 사용되는 32개 부분키는 4개를 한 묶음으로 8개로 구성되는 2차원 배열이며 다음과 같이 형성한다.

$$S_{key} = s_0 s_1 s_2 \cdots s_{126} s_{127}$$
$$S_{key}[i][j] = s_k s_{k+1} s_{k+2} s_{k+3}$$
$$k = 16*(i-1) + 4*j$$

여기서 i는 S box의 i번째 Si를 의미하며, 1부터 8까지 정수다. j는 0부터 3까지 정수로 Si의 행을 의미한다. 행 번호는 가장 윗줄이 0번이며 가장 아랫줄이 3번이다.

나. S box 재구성

재구성된 a번째 S box를 $\acute{S}a$이라하자. DES의 S box와 S_{key}를 이용하여 다음과 같은 방법으로 재구성한다. $S_a(n,c)$와 $S_a^{'}(n,c)$는 S_a와 $\acute{S}a$의 n번째 행 c열을 의미한다.

$$\acute{S}_a(n,c) = \{S_a(n,c) + S_{key}[a][n]\} \bmod 16$$

S_{key}의 모든 비트가 0으로 구성된 경우 재구성된 S box는 DES의 S box와 동일하다.

예를 들면 다음과 같이 S_{key}로부터 $S_{key}[1][n]$가 다음과 같이 구성된 경우

$$S_{key}[1][0] = 1010,\ S_{key}[1][1] = 0011$$
$$S_{key}[1][2] = 1011,\ S_{key}[1][3] = 0111$$

재구성된 $\acute{S}1$은 다음과 같다.

Table 1. 재구성된 $\acute{S}1$ 예(16진수)

8	E	7	B	C	9	5	2	D	4	0	6	F	3	A	1
3	2	A	7	1	5	0	4	D	9	F	E	C	8	6	B
F	C	9	3	8	1	D	6	A	7	4	2	E	5	0	B
6	3	F	9	B	0	8	E	C	2	A	5	1	7	D	4

재구성된 S box는 다음과 같은 특징이 있다.

① S_a의 각 행은 중복 없이 0부터 15가 고루 분포된다.

② 위와 같이 128비트 S_{key}에 따라 재구성된 S box는 2^{128}개 종류를 갖는다. 키 S_{key}에 따라 S box가 유일하게 구성된다.

다. 알고리즘

키에 따라 새롭게 재구성된 S box의 table을 사용할 경우 알고리즘은 $\acute{S}$ box table로 2^{11}비트 메모리가 필요하다. 본 연구에서 제안된 S box 재구성 알고리즘은 완성된 table을 별도로 미리 작성하지 않고 암호화 및 복호화 과정 비선형 시점에서 값을 구한 후 처리하므로 추가 메모리가 필요 없다.

암호화 및 복호화 과정 각 라운드에서 부분키와 결합한 48비트는 다음과 같이 비트열 f로 표시된다.

$$f = f_0 f_1 f_2 \cdots f_{47}$$

S box에서 값을 찾을 때 Sa box의 입력은 6비트로 f 중에서 f_{6a-6}부터 f_{6a-1}이다. 입력에서 f_{6a-6}와 f_{6a-1}는 Sa box의 행 n을 결정하고, $f_{6a-5}f_{6a-4}f_{6a-3}f_{6a-2}$ 가운데 4비트는 열 c를 결정한다. Sa box의 행에 해당하는 재구성 부분키 값 $S_{key}[a][n]$을 $Sa(n,c)$에 더한 결과에서 하위 4비트만 취하면 $\acute{S}a$ box에서 값을 찾은 것과 동일한 결과를 얻는다. Table 2는 E table 확장 후 라운드의 부분키와 결합한 48비트와 S_{key}를 입력하여 재구성 된 $S^{'}$ box에서 32비트 비선형 처리된 결과를 얻는 알고리즘이다.

DES의 E에 의해 48비트로 확장된 R은 라운드 부분키와 결합하여 f가 된 후 함수 S에 입력된다. 입력 s는 3차원 배열로 입력되는 S box이며 상자 번호, 행, 열로 구분된다. S_key는 S_{key} 128비트를 S box 번호와 행에 따라 입력되는 2차원 배열이다. 출력은 행별 덧셈으로 재구성된 $S^{'}$ box 출력 8개가 모여 32비트 q를 구성한다. 특정 라운드에서 부분키와 비트별 XOR로 결합한 입력 48비트 f에 대한 출력 32비트 q는 다음과 같다

$$q = \prod_{i=1}^{8} \{S(i, r_i, c_i) + s_{key}(i, r_i)\} \bmod 16$$
$$= \prod_{i=1}^{8} S_i^{'}(r_i, c_i) = \prod_{i=0}^{31} q_i$$
$$r_i = f_{6i-6} f_{6i-1}$$
$$c_i = f_{6i-5} f_{6i-4} f_{6i-3} f_{6i-2}$$

Table 2. S box 재구성 알고리즘

```
void S( byte (*s)[4][16], byte *f, byte *q,
                                    byte (*S_key )[4])

// f : E 확장된 6바이트 입력
// s : DES의 변형되지 않은 S box table 입력
// S_key : S_key를 S box 번호와 행으로 나눈 입력
// q : 재구성된 S box의 4바이트 출력
{
int i, row, col, idx ;
int shft_Val ; // 4비트 Sa에서 shift 된 출력
short sp_box[8] =
      { f에서 6비트씩 8개 S box 행렬 입력 초기화 } ;

for ( i = 0 ; i < 8 ; i ++ )
 {
   row = sp_box[i]&0x1 + (sp_box[i]&0x20)/0x10 ;
   col = (sp_box[i] &0x1e) / 2 ;

   shft_Val =
     (s[i][row][col] + S_key[i][row]) % 0x10 ;

   idx = i / 2 ;
   if ( i % 2 == 0 )
      *(q + idx) = shft_Val ;
   else
      *(q + idx) += (shft_Val << 4) ;
 }
}
```

q를 구하는 수식은 기존 S box에서 비선형으로 처리된 값을 구하는 절차 후 해당 행의 부분키 값을 더하여 하위 4비트만 취하므로 S box 재구성을 위한 추가 메모리는 필요 없다.

라. 하드웨어 구성

제안된 알고리즘의 하드웨어 구현은 S box의 각 단 출력(Sa 출력)과 그에 해당하는 S_{key}의 S_key 4비트를 더하여 하위 4비트를 얻음으로 재구성된 S' box의 값을 찾는다. Fig. 1은 S box의 a번째 상자 Sa와 Sa의 행에 대한 key를 4 X 1 MUX로 선별하여 4비트 전가산기(Full Adder)로 더한 후 하위 4비트를 취함으로 재구성된 Sa 값을 얻는다. E table에 의해 32비트에서 48비트로 확장된 후 라운드 부분키 48비트와 결합한 f는 6비트씩 나누어 8개 S box의 Sa 입력이 된다. S box의 각 단 Sa 입력 6비트 중 처음 비트와 마지막 비트는 Sa의 행 r_i을 결정하고, 가운데 4비트는 열 c_i을 결정한다. Fig. 1에서 Sa 위의 입력 4비트는 열 c_i을 결정하며, S_0와 S_1 두 비트는 Sa의 행 r_i과 S_key를 동시에 결정한다. 4X1 MUX에서 관련 행을 찾아 4비트 Adder 입력으로 연결한다. 4비트 Adder의 연산 결과에서 carry를 무시하면 sum 4비트는 modular 16 연산과 같은 결과를 얻는다. 위와 같은 구조를 8단을 설치 할 경우 병렬처리로 S box 재구성 하드웨어를 얻을 수 있다.

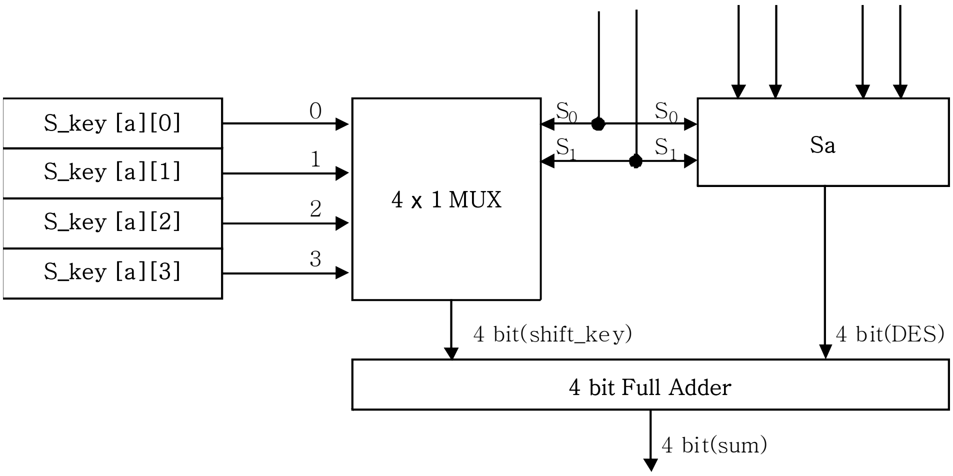

Fig. 1. Sa Box 재구성 하드웨어(s box 한 단)

6. 안전성 검토

안전성 검토는 제안된 알고리즘에 대해 차분 공격과 선형 공격을 검토한다. S box 재구성에 사용되는 임의의 128 비트 키 S_{key} 두 개를 선택하여 다음처럼 명명한다.

key 1 : 0xcc, 0x2a, 0xef, 0x1e, 0xd1, 0xed, 0xf4, 0x7f, 0xc4, 0x2c, 0xea, 0x9e, 0xd9, 0x3d, 0xfb, 0x8f

key 2 : 0xfc, 0xb5, 0x33, 0xfd, 0x57, 0x30, 0xab, 0xfd, 0xb5, 0x33, 0xfd, 0x57, 0x30, 0xab, 0xfb, 0x8f

가. 차분 공격

차분 공격은 S box의 XOR 분포 테이블의 DCa(α,β)을 이용하여 키를 분석한다. 제안된 알고리즘에 따라 재구성된 S box는 키 S_{key}가 없는 경우 XOR 분포 테이블을 알 수 없다.

예를 들면 DES의 S1 box 두 개의 입력 Se = 1x과 S'e = 35x에 대한 차분 Si = 34x에 대해서 출력이 S'o = Dx 임을 알고 있을 때 가능한 키는 다음 Table 3과 같이 추측할 수 있다.

Table 3. S box 키 적용과 키 추측(16진수)

구분	가능한 S box 입력	가능한 키
no key 적용	06, 32 10, 24 16, 22 1C, 28	07, 33 11, 25 17, 23 1D, 29
key 1 적용	**04, 30**	**05, 31**
key 2 적용	**18, 2C**	**19, 2D**

하지만 key 1과 key 2를 S box 재구성에 적용한 경우 가능한 입력과 키는 Table 3과 같이 각각 변화하지만 키 S_{key}가 공개되지 않으므로 재구성된 S box를 얻을 수 없으며 가능한 키도 추측할 수 없다. DES의 차분 공격분석은 2라운드 특성을 이용한다. 다음과 같은 예에서 S box 재구성에 키를 사용하지 않고 S box를 공개할 경우 2라운드 특성이 구성될 확률은 Table 4에서처럼 대략 1/234가 된다.

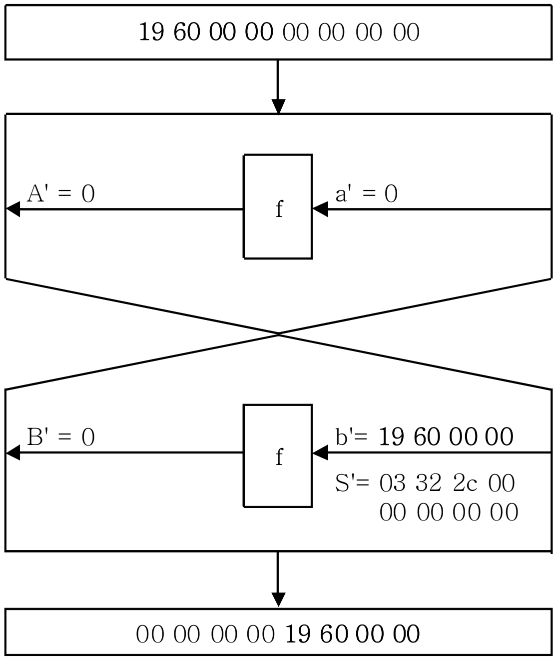

Fig. 2. 2라운드 특성

키에 따른 2라운드 특성이 구성될 확률은 다음 Table 4와 같이 계산된다.

Table 4. 키에 따른 2라운드 특성 확률

	No key	key 1	key 2
$S1_{ib'}$ = 03x → $S1_{ob'}$ = 0	14/64	**4/64**	**8/64**
$S2_{ib'}$ = 32x → $S2_{ob'}$ = 0	8/64	**8/64**	**8/64**
$S3_{ib'}$ = 2Cx → $S3_{ob'}$ = 0	10/64	**8/64**	**10/64**
특성 구성 확률	1/234	**1/1024**	**1/410**

16라운드 DES 공격은 15라운드 반복 특성을 이용해서 공격한다. 15라운드 반복 특성은 2라운드 특성을 7번사용 하고 확률 1을 갖는 1라운드 특성을 추가하여 구성한다. 따라서 15라운드 반복 특성 확률은 다음 Table 5와 같다. key 1, key 2가 없는 경우 Table 4에서 2라운드 구성 확률을 얻을 수 없어 Table 5에서 15라운드 반복 특성 확률을 알 수 없다. 따라서 선택 평문 공격에서 필요한 평문 쌍의 수를 추론할 수 없다.

Table 5. 키에 따른 15라운드 반복 특성 확률

	No key 확률	**key 1 확률**	**key 2 확률**
15라운드 반복 특성 확률	$(1/234)^7 \simeq 2^{-56}$	$\mathbf{(1/1024)^7 \simeq 2^{-70}}$	$\mathbf{(1/410)^7 \simeq 2^{-61}}$

나. 선형 공격

DES의 선형 근사는 S box의 Sa 입력 6비트가 α로 마스크 된 입력비트 위치와 β로 마스크 된 출력비트 위치의 XOR 값이 일치되는 경우의 수 NSa(α,β)를 사용한다. 예를 들면 $NS_5(16,15)$ = 12로 그 의미는 S_5의 입력 4번째 비트는 확률 12/64 = 0.19로 모든 출력비트의 XOR와 같아진다는 것을 의미한다. 이 식은 키 K를 임의로 고정하여 F 함수에 무작위 입력 X가 들어가면 확률 0.19로 다음 식이 성립함을 의미한다.

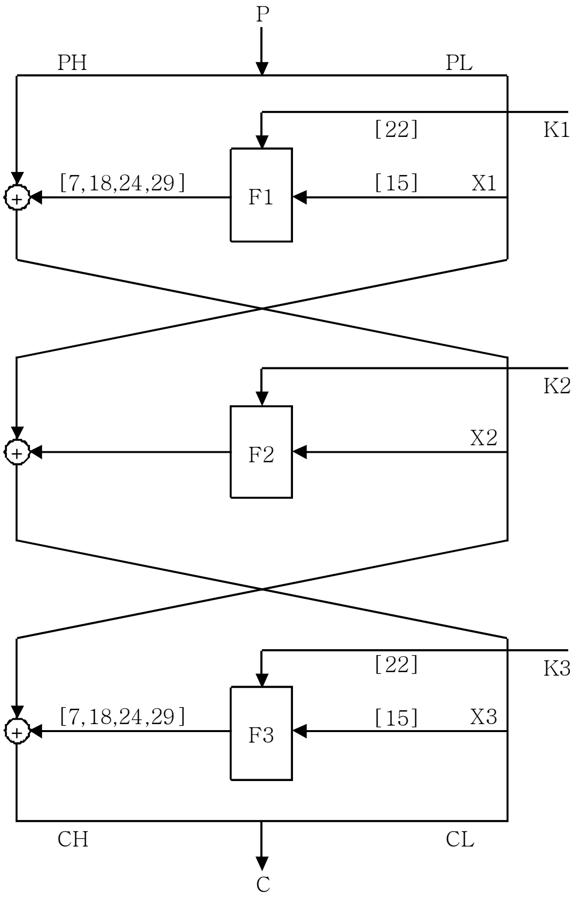

Fig. 3. 3단 DES

$$X[15] \oplus F(X,K)[7,18,24,29] = K[22]$$

위 식과 다음 식들에서 사용한 기호는 Matsui 논문에서 사용한 기호를 따른다.

Fig. 3의 3단 DES에서 재구성된 S box를 제1단의 F 함수에 위 식을 적용할 경우 다음 식이 성립할 확률은 Table 6 NS5(16,15)에 따른다.

$$X_2[7,18,24,29] \oplus P_H[7,18,24,29] \oplus P_L[15] = K_1[22]$$

키를 적용하지 않은 경우 위 식은 12/64확률로 성립하며, key 1을 적용한 경우 22/64, key 2를 적용한 경우 20/64확률로 성립한다. 같은 방식으로 제3단의 F 함수에 같은 식을 적용할 경우 위 식과 동일한 확률로 다음 식이 성립한다.

$$X_2[7,18,24,29] \oplus C_H[7,18,24,29] \oplus C_L[15] = K_3[22]$$

위의 두 식에서 X_2를 삭제하면 다음과 같은 식을 얻을 수 있다.

$$P_H[7,18,24,29] \oplus C_H[7,18,24,29] \oplus P_L[15] \oplus C_L[15] = K_1[22] \oplus K_3[22]$$

무작위로 주어진 평문과 암호문 쌍에 대하여 위 식이 성립하는 확률은 제1단의 F 함수에 적용된 식과 제3단의 F 함수에 적용된 식이 동시에 성립할 확률과 동시에 성립하지 않을 확률의 합이므로 키가 적용되지 않은 경우 0.7, key 1이 적용된 경우 0.55, key 2가 적용된 경우 0.57이 된다. 하지만 key 1과 key 2는 공개되지 않으므로, S box 재구성에 키를 적용한 경우 NSa (16,15) 값은 비공개이며 |NSa(16,15)-32| 값이 가장 큰 최량인지 알 수 없다. 그 결과 Matsui가 제안한 알고리즘의 적용은 최량확률을 알 수 없으므로 공격에 필요한 기지 평문 쌍과 추정 성공확률을 얻을 수 없다.

Table 6. 키에 따른 NS5(16,15)와 3단 DES

구분	NS5(16,15)	K1[22]⊕K3[22] 식 성립 확률
No key	12/64	$(12/64)^2+(1-12/64)^2=0.69$
key 1	22/64	$(22/64)^2+(1-22/64)^2=0.55$
key 2	20/64	$(20/64)^2+(1-20/64)^2=0.57$

7. 실 험

공개키로 64비트 초기 메시지 M을 전달하는 암호화 및 복호화 과정과 시간을 측정하였다.

가. 실험장비

제안된 알고리즘은 C++로 작성하였다. 실험은 업무용 PC에서 실행하였다. PC 성능은 Intel Atom CPU 230, 1.6GHz, 0.99GB RAM이다.

나. 실험 내용

DES에서 암호화에 요구되는 최소 블록단위는 64비트 M에 대한 암호화 및 복호화 시간을 측정하였다.

M ={ 65, 66, 67, 68, 69, 'F', 'G', 'H' }

DES의 암호화 및 복호화 시간은 대단히 짧아 1회 측정으로 정확한 값을 얻을 수 없다. 실험 순서는 구현된 알고리즘의 암호화 및 복호화 과정의 정확성 확인과 암호화/복호화 효율성 측정이다. 알고리즘 DES, 3DES 및 제안 알고리즘 3개를 구현하여 실험하였다. 각 알고리즘에 대해 메시지 M의 암호화 및 복호화 과정 시간측정으로 효율성을 비교하였다.

다. 실험 결과

64비트 M에 대한 암호화/복호화 과정을 10^8번 반복하여 각 알고리즘에 대한 암호화/복호화 실행 평균시간을 얻었다.

Table 7. 64비트 암호화/복호화 시간(sec)

구 분	암호화 시간	복호화 시간
DES	$25.81*10^{-6}$	$25.84*10^{-6}$
3DES	$79.17*10^{-6}$	$79.09*10^{-6}$
제안된 DES	$25.78*10^{-6}$	$26.23*10^{-6}$

DES와 3DES 실행시간은 라운드 수에 비례하고 있으며, 이론과 유사한 결과를 보인다. 제안된 알고리즘의 S box 재설계에 요구되는 추가시간은 DES 암호화/복호화 평균 실행 시간의 0.7% 정도로 부담이 없다. 3DES와 비교할 때 제안된 알고리즘은 실행속도 측면에서 DES와 비슷하며 암호화 강도는 전수공격 측면에서 3DES보다 강함을 알 수 있다.

8. 결 론

본 연구는 DES의 키 취약성에 대한 대책을 연구하였다. DES 취약성은 키 공간 축소와 S box의 공개로 차분 공격과 선형 공격에 대한 특수한 구조를 제공함에 있다. 이에 대한 대책으로 DES 확장키를 이용한 S box 재구성 방안을 제시하였다. 제안된 알고리즘에서 확장키가 없는 경우 재구성된 S box를 알 수 없어 선형 공격이나 차분 공격에 필요한 자료를 구성할 수 없다. 본 연구는 별도의 추가 메모리 없이 128비트 확장키로부터 S box를 재구성 할 수 있음을 보였다. 제안된 알고리즘은 컴퓨터 실험으로 암호화 및 복호화 과정 확인과 효율성을 시간으로 측정하여 DES 및 3DES와 비교하였다. 제안된 알고리즘의 암호화 및 복호화에 소요되는 시간은 DES와 비슷하지만, 확장키가 없는 경우 재구성 된 S box를 공개하지 않아 선형 공격과 차분 공격에 충분히 견딜 수 있음도 예제를 통해 보였다.

제안된 알고리즘의 하드웨어 구현은 4비트 MUX와 전가산기(Full Adder)를 S box 하단에 각 1개씩 추가 요구된다. 또한 제안된 알고리즘 복호화는 별도의 S box inverse table을 요구하지 않으며 확장키로부터 재구성된 S box를 사용한다. 제안된 암호화 알고리즘은 Kerckhoffs 원칙을 만족하며, 키를 제외한 암호화 과정 전부를 공개하는 블록암호이다.

후 기

본 연구 논문은 10년도 공군사관학교 국고연구비(KAFA 10-26) 예산지원으로 수행된 결과입니다.

Reference

[1] 안태남 외 2 역, “정보보안 이론과 실제”, 한빛 미디어, 2006.

[2] 김광조, “DES 선형 해독법에 관한 해설(1)”, 통신정보보호학회지 제3권 제3호, 1993.

[3] E. Biham and A. Shamir, “Differential Cryptanalysis

of DES-like Crypto Systems", Jour. of CRYPTOLOGY, Vol. 4, No. 1, 1991.

[4] E. Biham and A. Shamir, "Differential Cryptanalysis of FEAL and N-hash", Technical Report, Weizmann Institute of Science, Israel, 1991.

[5] M. H. Dawson and S. E. Tavares, "An Expanded Set of S Box Design Criteria based on Information Theory and it Relation to Differential-like Attacks", Proc. of EUROCRYPT91, Springer-Verlag, 1991.

[6] L. Brown, M. Kwan, J. Pieprzyk and J. Seberry, "Improving Registance to Differential Cryptoanalysis and the Redesigh of LOKI", Abstract of ASIACRYPT'91, 1991

[7] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", Abstracts of EUROCRYPT'93, pp. W112～W123, May 1993.