情報保護學會論文誌
第12券 第6號, 2002. 12

블록 암호에 대한 효율적인 선형 공격 방법

최 준*, 홍득조*, 홍석희*, 이상진*, 임종인*

Multiple Linear Cryptanalysis-Revisited

Jun Choi*, Deukjo Hong*, Seokhie Hong*, Sangjin Lee*, Jongin Lim*

요 약

1993년도에 선형 공격이 Matsui에 의해 제안된 이후에 이를 개량한 여러 선형 공격들이 등장하였다. 그 중에 한 가지는 B. Kaliski와 M. Robshaw에 의한 방법인데, 이 방법은 여러 개의 독립적인 선형 근사식을 동시에 이용하여 블록 암호를 공격하는 새로운 방법이였다. 이 방법은 선형 공격 보다 더 적은 기지 평문수를 요구한다는 장점은 있었지만 실제로 그들의 방법을 블록 암호에 적용하는 데에는 문제점이 있었다. 본 고에서는 그러한 문제점을 해결하면서 동시에 여러 개의 독립적인 선형 근사식을 이용할 수 있는 방법을 제시한다. 본 고에서 제시된 방법을 이용했을 때 선형 공격에 비해 8,16 라운드 DES에 대해 5배,1.25배 더 적은 기지 평문을 가지고 각각 95%, 86% 확률로 공격에 성공할 수 있었으며, 또한 선택 평문을 이용한 L. R. Knudsen과 J. E. Mathiassen의 방법을 본 고에서 제시한 방법에 접목하면, 약 $2^{40.6}$개 이하의 기지 평문들을 이용하여 86% 성공 확률로 키 15 비트를 찾을 수 있다. 이 결과는 현재까지 DES에 대한 공격 중 가장 우수한 결과이다.

ABSTRACT

Many Linear attacks have introduced after M. Matsui suggested Linear Cryptanalysis in 1993. The one of them is the method suggested by B. Kaliski and M. Robshaw. It was a new method using multiple linear approximations to attack for block ciphers. It requires less known plaintexts than that of Linear Cryptanalysis(LC) by Matsui, but it has a problem. In this paper, we will introduce the new method using multiple linear approximation that can solve the problem. Using the new method, the requirements of the known plaintexts is 5(1.25) times as small as the requirements in LC on 8(16) round DES with a success rate of 95%(86%) respectively. We can also adopt A Chosen Plaintext Linear Attack suggested by L. R. Knudsen and J. E. Mathiassen and then our attack requires about $2^{40.6}$ chosen plaintexts to recover 15 key bits with 86% success rate. We believe that the results in this paper contain the fastest attack on the DES full round reported so far in the open literature.

Keyword : *블록 암호, 선형 공격, 다중 선형 공격*

Ⅰ. 서 론

블록 암호에 대한 공격법 중 하나인 차분 공격(DC)[1]이 E. Biham, A. Shamir에 의해 처음 소개됐을 때 비교적 높은 확률을 갖는 하나의 차분 특성이 요구되었는데, 이후 L. R. Knudsen은 비록 높은 확률을 갖지 못하는 차분 특성들이라도 찾아내려는 키 비트들이 동일한 여러 차분 특성들을 규합하여 이용하면 효율적인 공격이 될 수 있는 부정 차분 공격[3]을 제안하였다. 이 공격은 하나의 차분 특성이 갖고 있는 확률보다 여러 차분 특성들을 합했을 때의 확률이 하나의 차분 특성에 대한 확률보다 커진다는 이점을

* 고려 대학교 정보 보호 기술 연구 센터 CIST({jun,hongdj,hsh,sangjin,jilim}@cist.korea.ac.kr)

이용한다.

차분 공격과 더불어 블록 암호의 강력한 공격 중에 하나인 선형 공격은 차분 공격과 유사한 흐름을 갖고 발전하였다. M. Matsui는 DES에 대해 처음으로 선형 공격(LC)[5]을 적용하였으며, 확률이 높은 하나의 선형 근사식만을 이용하였다. M. Matsui이후 B. Kaliski와 M. Robshaw는 선형 공격의 원리는 유지하면서 여러 개의 선형 근사식을 동시에 이용하여 공격하는 방법을 제안하였다.[2] 본 고에서는 이 공격법을 다중 선형 공격법(Multiple Linear Cryptanalysis)이라 부르고 편의상 MLC라 표기한다.

MLC의 다중 선형 근사식에서 사용되는 여러 선형 근사식들은 선형 근사식 각각이 모두 같은 키 비트들을 포함 해야한다. 즉, [2]의 표현을 따른다면 각 선형 근사식에서 오른쪽 부분이 모두 같은 선형 근사식들을 의미한다. 그러나 그들의 방법을 실제 공격에 적용하기 위해서는 강력한 제한 조건이 요구되었다. 즉, 각 선형 근사식이 포함하는 키 비트들의 XOR 된 값 1 비트가 모두 같아야만 되므로, 각 선형 근사식에 포함되는 키 비트들의 값이 $K[k_1]$, $K[k_2]$, $\cdots$, $K[k_c]$($k_1, k_2, \cdots, k_c$은 각 선형 근사식마다 포함하는 키 비트들의 위치를 의미하고 이 위치는 각각의 선형 근사식마다 다르다)일 때, 각 선형 근사식에 대한 $K[k_1]\oplus K[k_2]\oplus\cdots\oplus K[k_c]$을 모두 동일하게 하기 위한 추가적인 계산량이 요구되었다. 이러한 계산량으로 인해서 MLC를 현실적으로 블록 암호 공격에 이용하는 데에는 많은 어려움이 있다.
본 논문에서는 선형 근사식의 개수가 증가함에 따라 계산 시간이 지수승으로 증가하는 다중 선형 공격법의 문제점을 극복하는 방법을 제안한다. 구체적으로 말하면 하나의 선형 근사식을 적용하는데 소요되는 계산량을 T라하고 선형 근사식의 개수가 k일 때 전체적인 계산 시간은 단지 kT이다. 따라서 필요한 기지 평문을 줄이면서, 효과적으로 공격할 수 있는 공격이 된다.

본 논문의 방법의 요점은 하나의 선형 근사식에 대한 확률보다 여러 선형 근사식들을 동시에 이용하였을 때 각각의 선형 근사식들의 확률의 합이 더 크다는 원리를 이용한다. 편의상 본 논문에서의 방법을 이 이후에 ELC(Efficient LC)라 부르겠다.

이 논문은 모두 6개의 절로 구성되어 있다. 2절은 M. Matsui에 의해 소개된 선형 공격을 요약하고, 3절은 B. Kaliski와 M. Robshaw에 의해 제시된 다중 선형 공격법(MLC), 4절은 본 논문에서 제시한 효율적인 선형 공격법(ELC)의 소개와 분석, 그리고 5절에서는 ELC를 8,16 라운드 DES에 대해 적용한 실험 결과를 보일 것이다. 끝으로 6절에서는 이 논문의 결론을 언급할 것이다.

II. 표기와 선형 공격(LC)

이 논문의 대부분의 표현은 새롭게 정의내린 몇 가지를 제외하고는 [2,5]의 표현을 따른다.

- $P(=P_H\|P_L)$: 평문 64 비트(||은 연접을 의미함)
- $C(=C_H\|C_L)$: 암호문 64 비트
- P_H, P_L : 평문의 상위 32 비트, 하위 32 비트
- C_H, C_L : 암호문의 상위 32 비트, 하위 32 비트
- X_i : i번째 라운드의 F함수에서 32 비트 입력
- K_i : i번째 라운드의 48 비트 라운드 키
- $F_i(X_i, K_i)$: i번째 라운드의 F 함수
- $A[i]$: A의 i번째 비트
- $A[i,j,\cdots,k] : A[i]\oplus A[j]\oplus\cdots\oplus A[k]$
- χ_P : 선형 근사식이 포함하는 평문 비트들
- χ_C : 선형 근사식이 포함하는 암호문 비트들
- χ_K : 선형 근사식이 포함하는 키 비트들
- χ_{F_r} : 하나의 선형 근사식이 포함하는 F함수의 출력 비트들
- $P[\chi_P]$: χ_P가 평문의 $i_1,\ldots,i_a$비트들 일 때 $P_{i_1}\oplus\ldots\oplus P_{i_a}$
- $C[\chi_C]$: χ_C가 암호문의 $j_1,\ldots,j_b$비트들 일 때 $C_{j_1}\oplus\ldots\oplus C_{j_b}$
- $K[\chi_K]$: χ_K가 키의 $k_1,\ldots,k_c$비트들 일 때 $K_{k_1}\oplus\ldots\oplus K_{k_c}$
- $F_r(C_L,K_r)[\chi_{F_r}]$: χ_{F_r}가 F함수의 출력 비트들 $l_1,\ldots,l_d$일 때 $F_r(C_L,K_r)_{l_1}\oplus\ldots\oplus F_r(C_L,K_r)_{l_d}$

선형 공격은 임의의 주어진 평문 P와 키 K 및 이에 대응하는 암호문 C에 대하여 확률 $p\neq 1/2$로 다음과 같은 선형 근사식이 존재할 때 적용 가능하다.

$$P[\chi_P]\oplus C[\chi_C]=K[\chi_K] \quad (1)$$

식 (1)이 성립할 확률이 $p\neq 1/2$이면, 키의 1비트 정보 $K[\chi_K]$를 다음의 알고리즘에 의해 유추할 수 있다.

◉ 알고리즘 1

(Step 1)

주어진 모든 기지 평문과 이에 대응되는 암호문의 쌍으로부터 식 (1)의 좌변을 계산한다.

(Step 2)

이 값이 0이 되는 평문수 T_i가 전체 기지 평문 수의 1/2이상이 되면 $K[\chi_K]=0$ ($p>1/2$일 때), 또는 1 ($p<1/2$일 때)로 추정하며, 이 값이 0이 되는 평문수가 전체 평문의 1/2이하가 되면 $K[\chi_K]=1$ ($p>1/2$ 일 때), 또는 0 ($p<1/2$일 때)로 추정한다.

위의 알고리즘이 성공할 확률은 전체 기지 평문수 N과 식 (1)의 성립 확률 p로 결정되며 N 또는 $|p-1/2|$이 클수록 공격에 성공할 가능성이 커진다. 식 (1)의 형태를 갖는 선형 근사식 중에 $|p-1/2|$이 최대가 되는 것을 최량 표현이라 부르며, 그 성공 확률을 최량 확률이라 부른다.

위의 방법에서 알 수 있듯이, 알고리즘 1은 (1)식을 이용하여 한 비트 키 정보만을 얻어내고 있다. [5]의 방법을 따라간다면, 첫 번째 라운드 또는 마지막 라운드의 키를 먼저 추측하여 보다 많은 비트를 알아내는 방법이 있다. 이 논문에서는 이것을 알고리즘 2라고 부를 것이다. 알고리즘 2를 서술하기 전에 r 라운드 암호에 알고리즘 2를 적용하기 위한 기본적인 성질과 등식을 알아보겠다.

r 라운드의 암호를 공격하기 위해서는 $r-1$ 라운드의 최량 표현을 사용한다. 즉, 최종 라운드는 r번째 F함수에 입력되는 라운드키 K_r을 이용하여 복호화된다는 점을 이용하여 근사식 중에 F함수를 넣어서 계산한다. 그 결과는 다음의 등식을 이끌며, 이 식을 $r-1$ 라운드의 최량 확률로 성립시키도록 한다.

$$P[\chi_P] \oplus C[\chi_C] \oplus F_r(C_L, K_r)[\chi_{F_r}] = K[\chi_K] \quad (2)$$

그런데 만일 식 (2)에 틀린 라운드 키 K_r이 대입되면 식 (2)는 거의 성립하지 않을 것이다. 즉, 식 (2)의 성립 확률이 거의 1/2이 될 것이다. 따라서 다음의 알고리즘에서와 같이 maximum likeli hood 법을 이용하여 K_r 및 $K[\chi_K]$를 의미있게 추정 할 수 있다.

◉ 알고리즘 2

(Step 1)

K_r의 각 후보값에 대한 카운터를 설정하여 0으로 초기화한다.

(Step 2)

주어진 각 기지 평문과 이에 대응되는 암호문쌍에 대하여, K_r의 각 후보값에 대한 식 (2)의 좌변을 계산하고, 그 결과가 0으로 된 키에 해당하는 카운터 값에 1을 증가시킨다.

(Step 3)

모든 카운터의 값에서 최대치 T_{max}와 최소치 T_{min}을 비교하여 기지 평문수 N에 대하여

- $|T_{max}-N/2|>|T_{min}-N/2|$이면 T_{max}에 대응하는 K_r을 선택하고, 식 (2)의 우변은 0 ($p>1/2$ 일 때) 또는 1 ($p<1/2$ 일때)로 추정한다.
- $|T_{max}-N/2|<|T_{min}-N/2|$이면 T_{min}에 대응하는 K_r을 선택하고, 식 (2)의 우변은 1 ($p>1/2$ 일 때) 또는 0 ($p<1/2$ 일때)로 추정한다.

선형 공격을 알고리즘 2에 적용하기 위해서는 K_r, C_L의 비트들 중 $F_r(C_L, K_r)[\chi_{F_r}]$에 영향을 미치는 비트들을 일치시켜야 한다. 이러한 부분키 비트를 effective 키 비트라 하고 마찬가지로 이러한 암호문 비트를 effective 암호문 비트라 한다. 만약 어떤 키 비트의 추측값이 옳았다면 $F_r(C_L, K_r)[\chi_{F_r}]$의 값은 올바른 값이고 이를 통해 선형공격을 할 수 있을 것이다. 반대로 키 추측이 틀렸다면 $F_r(C_L, K_r)[\chi_{F_r}]$의 값은 0과 1이 거의 같은 비율로 존재하여 선형공격이 이루어지지 않는다. 따라서 많은 양의 데이터에 대해 위 식을 이용하여 0과 1의 개수의 차이를 살펴보면 올바른 추측일 때의 값은 다른 추측값에 비해 구별되게 나타날 것이다.

위의 방법을 확장시켜 첫 번째 라운드 키와 마지막 라운드 키를 동시에 추측하는 방법[8]이 있다. 이 방법은 두 개의 라운드 키를 동시에 고려하므로 추측해야 하는 양이 많고 effective 키 비트들의 양이 늘어나지만 근사식의 확률을 높일 수 있으므로 성공확률은 높아진다. [8]에 제시된 방법은 M. Matsui가 1993년도에 처음으로 선형 공격을 소개한 이후 이듬해인 1994년도에 그것을 약간 개선하여 DES를 공격할 때 좀 더 효율적인 방법이다. 실제로 선형 공격을 할 때 본 고에서 DES에 대한 공격 복잡도를 비교하는 척도는 [8]에 나와있는 공격 복잡도를 기준으로 하고 있다.

Ⅲ. 다중 선형 공격(MLC)

χ_K는 같고 χ_P, χ_C들은 다른 선형 근사식 n개가 존재한다고 가정하자. 이 n개의 근사식 모두를 동시에 사용하는 다중 선형 공격법[2]은 다음과 같다. 우선 n개의 근사식 중 i번째의 근사식을 다음과 같이 정의한다.

$$P[\chi^i_P] \oplus C[\chi^i_C] = K[\chi_K] \quad (3)$$

위 선형 근사식의 성립 확률이 p일 때, 선형 근사식에 대한 bias는 $|p-1/2|$이 된다는 것을 유념하자.

분석을 쉽게 하기 위해서 각각의 bias 값 ε_i를 0보다 크다고 가정한다(만약 원래의 bias 값 ε_i이 0보다 작다면 선형 근사식의 왼쪽이나 오른쪽 중 하나에 1을 XOR함으로써 bias 값이 양수가 되도록 맞추어 줄 수 있다). 위의 방법을 알고리즘 1에 적용하여 개선하면 다음의 알고리즘 1M을 얻을 수 있다.

◉ 알고리즘 1M

(Step 1)

N=평문과 암호문 쌍의 개수

T_i=식 (3)의 왼쪽의 값을 계산하여 0 이 되는 쌍의 수 $(1 \le i \le n)$

(Step 2)

각각의 T_i에 가중치 a_i를 적용하여 $U = \sum_{i=1}^{n} a_i T_i$를 계산 (a_i들은 $\sum_{i=1}^{n} a_i = 1$을 만족한다.)

(Step 3)

만약 $U > N/2$ 이면 $K[\chi_K]=0$ 이라고 추측한다.

만약 $U < N/2$ 이면 $K[\chi_K]=1$ 이라고 추측한다.

알고리즘 1M에 사용된 선형 근사식들은 모두 χ_K가 같아야 한다는 가정이 있었다. 그래서 [2]에서 그러한 가정을 없애기 위한 방법을 다음과 같이 제안했다. 이 논문에서는 이 방법을 편의상 알고리즘 1M*라 부른다.

우선 n개의 선형 근사식이 있고, 그것의 i번째 선형 근사식이 (4)의 형태를 갖는다고 가정하자.

$$P[\chi'^i_P] \oplus C[\chi'^i_C] = K[\chi'^i_K] \quad (4)$$

각각의 선형 근사식을 결합하기 위하여 각각의 $i, j (2 \le i, j \le n, i \ne j)$에 대해 $K[\chi'^i_K]$와 $K[\chi'^j_K]$가 같은지 틀린지를 먼저 추측한다. n개의 선형 근사식 중 하나의 $K[\chi_K]$ (0 혹은 1)를 고정시킨다면 추측 값의 경우의 수는 2^{n-1}이고, 다음과 같이 n개의 같은 키 비트에 대한 선형 근사식을 얻을 수 있다.

$$P[\chi'^i_P] \oplus C[\chi'^i_C] \oplus \Delta^i = K[\chi'^1_K] (2 \le i \le n) \quad (5)$$

위의 식에서의 Δ^1은 항상 0이고, $i \ge 2$의 경우 Δ^i는 추측에 의존한다고 하자. 위의 식을 이용하여 각각의 추측에 대해 다중 선형 공격을 적용하여 $K[\chi'^1_K]$의 값을 결정한다.

M. Matsui가 알고리즘 1에서 알고리즘 2라는 응용을 생각했듯이, B. Kaliski와 M. Robshaw 또한 알고리즘 1M으로부터 알고리즘 2M을 비슷한 방법으로 구성해내었다. 우선 식 (2)와 유사하게 다중 선형 근사식의 경우 n개의 선형 근사식 중에 i번째 선형 근사식 (6)을 얻을 수 있다. (2)와 다른 점은 [8]에서 제시되었던 방법으로서 고려하는 라운드 함수가 두 라운드로 확장되었다는 것뿐이다.

$$P[\chi'^i_P] \oplus C[\chi'^i_C] \oplus F_1(P_L, K_1)[\chi'^i_{F_1}] \oplus F_r(C_L, K_r)[\chi'^i_{F_r}] = K[\chi_K] \quad (6)$$

위의 식 (6)의 bias 값을 양수 ε_i라 하면 다음의 알고리즘을 얻을 수 있다.

◉ 알고리즘 2M

(Step 1)

N=평문과 암호문 쌍의 개수

$K_1^{(g)} (g = 1, 2, \ldots)$는 K_1의 후보가 되는 값들

$K_r^{(h)} (h = 1, 2, \ldots)$는 K_r의 후보가 되는 값들

$T^i_{(g,h)}$는 각 후보 쌍 $(K_1^{(g)}, K_r^{(h)})$에 대하여 식 (6)의 왼쪽 값이 0 이 되는 쌍의 개수

(Step 2)

$a_i = \varepsilon_i \sum_{i=1}^{n} \varepsilon_i$로 놓고 각각의 g, h에 대해

$U = \sum_{i=1}^{n} a_i T^i_{g,h}$를 계산한다.

(Step 3)

$U_{\max}$는 $U_{(g,h)}$ 값들 중에서 가장 큰 값 $U_{\min}$는

$U_{(g,h)}$ 값들 중에서 가장 작은 값

만약 $|U_{\max}-N/2|>|U_{\min}-N/2|$이면 키의 후보를 $U_{\max}$의 값들로 보고 $K[\chi_K]=0$ 이라고 추측한다.

만약 $|U_{\max}-N/2|<|U_{\min}-N/2|$이면 키의 후보를 $U_{\min}$의 값들로 보고 $K[\chi_K]=1$ 이라고 추측한다.

알고리즘 2M에 사용된 선형 근사식들도 알고리즘 1M에 사용된 선형 근사식과 마찬가지로 모두 χ_K가 같아야 한다는 가정이 있었다. 그래서 [7]에서 그러한 가정을 없애기 위한 방법을 다음과 같이 제안했다. 이 논문에서는 이 방법을 편의상 알고리즘 2M*라 부른다.

우선 n개의 선형 근사식들이 있고, 그것의 i번째 선형 근사식이 (7)의 형태를 갖는다고 가정하자(식 (7)은 이후에 새로운 알고리즘 E를 소개하면서 언급할 것이다. 식 (6)과의 차이는 n개의 선형 근사식에 대한 우변의 χ_K들이 모두 다르다는 것이다).

◉ 알고리즘 2M*

(Step 1)

N = 평문과 암호문 쌍의 개수

$K_1^{(g)}(g=1,2,...)$는 K_1의 후보가 되는 값들

$K_r^{(h)}(h=1,2,...)$는 K_r의 후보가 되는 값들

$T^i_{(g,h)}$는 각 후보 쌍 $(K_1^{(g)},K_r^{(h)})$에 대하여 식 (7)의 왼쪽 값이 0이 되는 쌍의 개수

(Step 2)

$a_i=\varepsilon_i\sum_{i=1}^{n}\varepsilon_i$로 놓고 각각의 g,h에 대해 식 (7)의 우변에 대한 각각의 추측값 $C=(c_1,\cdots,c_n)$을 대입하여 다음을 계산한다.

$$U_{g,h}[C]=\sum_{i=1,c_i=0}^{n}a_iT^i_{g,h}+\sum_{i=1,c_i=1}^{n}a_i(N-T^i_{g,h})$$

(Step 3)

$U_{\max}$는 $U_{(g,h)}$ 값들 중에서 가장 큰 값이라면 키의 후보를 $U_{\max}$의 값들로 보고 $K[\chi_K^i]=c_i$로 추측한다.

알고리즘 2M*는 $C=(c_1,\cdots,c_n)$의 각 항 한 비트를 결정하기 위해 n 비트에 대한 추측을 하는 과정이 요구된다는 것을 유념하자.

Ⅳ. 효율적인 선형 공격(ELC)

이번 절에서는 새로운 알고리즘 ELC를 제안하고 LC, MLC와 ELC에 대한 비교 결과를 나타낼 것이다.

4.1 MLC의 문제점

이제 우리는 MLC[2]의 문제점을 지적하고 본 논문에서 새로운 방법을 제안한 이유에 대해서 논의할 것이다.

알고리즘 1M, 2M을 여러 선형 근사식에 적용하기 위해서는 모든 선형 근사식들이 같은 χ_K, 같은 effective 키 비트들을 가져야 한다는 가정이 필요했다. 그러나 이들 조건은 알고리즘 1M, 2M의 사용을 제한한다. 최근에 제안되고 설계된 블록 암호들은 비교적 높은 확률을 가지면서 그러한 조건들도 만족하는 선형 근사식을 많이 갖지 않을 것이다.

더욱이 다음과 같은 의문점을 생각해 볼 수 있다: [2]와 [7]에서 각각 제시된 알고리즘 1M*, 2M*는 서로 다른 χ_K를 갖는 여러 선형 근사식들에 대해 좋은 대책이 될 수 있을까? 알고리즘 1M*, 2M*는 이용되는 선형 근사식에 비례하여 너무 많은 추가적 계산 과정을 요구하기 때문에 대답은 부정적이다. 예를 들어 만일 우리가 서로 다른 χ_K를 갖는 n개의 선형 근사식을 사용한다고 할 때, 선형 공격에 소요되는 계산량이 T이면 2^{n-1} 정도의 추측 과정(키 한 비트 정보를 미리 추측)을 거쳐야만 하기 때문에 $2^{n-1}T$의 계산시간이 필요하다. 따라서 n이 작지 않다면 계산량은 매우 클 것이다.

그러나 다음의 두 가지 관점을 한번 생각해보자. 첫 번째, 다중 선형 근사식을 이용하기 위해 알고리즘 1M*, 2M*와 같이 추가적인 계산량을 인정하고 하나의 선형 근사식에 대해 오른쪽 키 한 비트 정보와 한(두)라운드 하나의 S-box 에 대한 키 6(12) 비트(DES의 경우)들을 찾을 것인가? 아니면 둘째, 여러 선형 근사식을 이용하지만 알고리즘 2M을 약간 변형시켜 선형 근사식들의 오른쪽 키 한 비트 정보는 못찾더라도 추가적인 계산량 없이 알고리즘 2M*와 마찬가지로 한(두) 라운드 S-box의 키 비트들을 찾을 것인가? 두 가지 물음중에 두 번째가 암호 알고리즘을 공격하는데에 있어서는 더 효율적일 것이다. 따라서 본 고에서 제시하게 될 ELC는 여러 선형 근사식을 이용하지만 알고리즘 1M에서

키 한 비트 정보를 예측하는 관점대신 알고리즘 2M에 초점을 맞추어 한 라운드에 대한 effective 키 비트를 알고리즘 2M*이 갖고 있는 추가적 계산량의 문제를 해결하면서 추측하려한다.

이러한 문제점은 알고리즘 2M 안에 $U_{g,h}(=\sum_{i=1}^{n} a_i T^i_{g,h})$ 부분이 $|T^i_{g,h}-N/2|$의 weighted sum으로 대체된다면 해결될 수 있다. 본 논문에서 제안하는 알고리즘은 각 선형 근사식의 $K[\chi_k]$에 대해서는 추측을 하지 않고 단지 처음과 마지막 라운드의 effective 키 비트에 대한 추측에 초점을 맞춘다. 실제 공격에서 키 비트를 찾는 알고리즘의 효율성을 위해 각 선형 근사식은 같은 effective 키 비트를 가져야만 한다는 것을 유념하자.

4.2 알고리즘 E의 소개와 분석

ELC는 다음의 알고리즘을 따른다. 알고리즘 분석의 편이성을 위해 각각의 선형 근사식에 대한 bias를 나타내는 ε_i 들은 모두 양수라고 가정하며, n개의 선형 근사식들에 대해, 그것의 i번째 선형 근사식이 (7)의 형태를 갖는다고 가정하자.

$$P[\chi^i_P]\oplus C[\chi^i_C]\oplus F_1(P_L,K_1)[\chi^i_{F_1}] \oplus F_r(C_L,K_r)[\chi^i_{F_r}]=K[\chi^i_K] \quad (7)$$

◉ 알고리즘 E

(Step 1)

N은 공격에 사용되는 기지 평문의 수이고, r은 블록 암호의 전체 라운드 수를 나타낸다고 하자. $K_1^{(g)}(g=1,2,...)$와 $K_r^{(h)}(h=1,2,...)$는 각각 K_1과 K_r에 대한 후보가 되는 값들이다. 그러면 각각의 키 순서쌍 $(K_1^{(g)},K_r^{(h)})$과 i번째 선형 근사식에 대해 $T^i_{g,h}$는 K_1이 $K_1^{(g)}$, K_r이 $K_r^{(h)}$에 의해 대체 되었을 때 식 (7)의 왼쪽이 0이 되는 평문들의 수라고 하자.

(Step 2)

$a_i=\varepsilon_i / \sum_{i=1}^{n}\varepsilon_i$ $(\sum_{i=1}^{n} a_i=1)$이라고 하자. g, h 각각에 대해 $U_{g,h}=\sum_{i=1}^{n} a_i|T^i_{g,h}-N/2|$을 계산한다.

(Step 3)

$U_{\max}$를 모든 $U_{g,h}$들 중에서 가장 큰 값이라고 하자. $U_{\max}$에 대응되는 g, h을 K_1, K_r의 후보로 본다.

충분히 큰 N에 대해, 만일 키 후보쌍 $(K_1^{(g)},K_r^{(h)})$이 올바르다면, $T^i_{g,h}(i=1,2,\cdots,n)$과 $N/2$의 차이는 다른 키 순서쌍과 $N/2$의 차이 보다 클 것이고 따라서, 알고리즘 E에서 $U_{g,h}$이 매우 높은 확률로 가장 큰 값이 될 것이다. 만일 $(K_1^{(g)},K_r^{(h)})$ 이 틀린 키 후보쌍 이라면, 틀린 키 후보들에 대한 $U_{g,h}$은 거의 비슷한 값으로 나올 것이며 분명히 올바른 키 일 때의 값과 차이가 날 것이다. N이 커질수록 올바른 키와 틀린 키에 대한 통계치 사이의 간격은 더 커질 것이다. 이점이 바로 알고리즘 E가 올바른 키와 틀린 키를 구별해 낼 수 있는 이유이다.

어떤 경우에 대해서는 알고리즘의 첫 라운드나 마지막 라운드 중에 한 라운드에 대해서만 effective 키 비트들을 추측하기 위해서 식 (7)을 약간 변화시켜 알고리즘 E를 적용하겠다. 즉, 그 식은 다음의 식 (8)의 형태를 갖는다(아래의 식에서 n의 의미는 보통 r 라운드 암호의 1 혹은 마지막 라운드를 의미하는 r의 값이 된다).

$$P[\chi^i_P]\oplus C[\chi^i_C]\oplus F_n(P_L,K_n)[\chi^i_{F_n}]=K[\chi^i_K] \quad (8)$$

4.3 ELC에서 요구하는 데이터 복잡도

$|T^i_{g,h}-N/2|$의 weighted sum을 이용하는 알고리즘 E에서는, 오직 선형 근사식들의 effective 키 비트들만이 고려된다. $K[\chi_K]$이 0 혹은 1이라는 것에 상관없이 알고리즘 E는 단지 $K[\chi_K]$값이 고정되어 있다는 사실만을 이용한다. 본 논문에서는 알고리즘의 키를 찾기 위해 요구되는 기지 평문의 수를 줄이기 위해 가능한 많은 선형 근사식들을 모으는 것에 중점을 둔다. 이렇게 여러 선형 근사식들을 동시에 알고리즘 E에 이용했을 때, 고정된 성공 확률에 대해 요구되는 데이터 복잡도는 어떻게 결정이 될까? 이 질문에 대한 해답은 실험 결과로서 알 수 있었다. 데이터 복잡도를 결정하는 원리는 부정 차분 공격[3]에서 데이터 복잡도를 결정하는 것과 유사하다. 데이터 복잡도가 결정되는 방법은 다음과 같다.

M. Matsui는 [6]에서 차분 특성의 확률과 선형 근사식의 확률을 정의했고, 이를 통해 차분 공격과 선형 공격에 대한 몇 가지 성질들을 알 수 있다. p_D를 하나의 차분 특성에 대한 확률이라 하고, $p_L=1/2+\varepsilon$을 하나의 선형 근사식에 대한 확률이라 하자. 차분 공격과 선형 공격에서 $1/p_D$의 상수 배와, $1/\varepsilon^2$의 상

수 배는 각각의 공격에 대해 필요한 평문수를 결정한다.

부정 차분 공격에서는 사용되는 모든 차분 특성들의 확률의 합 (p_D^{all})이 공격에 필요한 데이터 양을 추정하기 위해 요구되며, 따라서 n개의 선형 근사식 각각에 대한 bias가 $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$일 때 $\sum_{i=1}^{n} \varepsilon_i^2$이 ELC에서 필요한 데이터 양을 결정하기 위해 요구된다고 추측한다. 즉, $1/p_D^{all}$이 부정 차분 공격에서 필요한 평문수를 결정하고, 이런 사실에 기반하여 $1/\sum_{i=1}^{n} \varepsilon_i^2$이 ELC에서 필요한 평문수를 결정한다고 추측할 수 있다. 편의상 $\sum_{i=1}^{n} \varepsilon_i^2$을 E^2에 의해 표기한다. 다음 절에서 여러 실험 결과들을 통해 E^2이 선형 공격[5]에서 하나의 선형 근사식에 대한 bias의 제곱처럼 행동한다는 것을 볼 수 있다.

예를 들어, 알고리즘 2에서 $2\varepsilon^{-2}$의 기지 평문을 이용하여 48.6% 성공 확률을 가질 수 있었다. 유사하게 알고리즘 E에서 $2E^{-2}$의 기지 평문을 이용하여 약 45-50% 성공 확률을 가질 수 있다.

4.4 알고리즘 2M과 알고리즘 E의 관계

알고리즘 2M의 Step 3에서 볼 수 있듯이, 알고리즘 2M은 만약 $|U_{\max} - N/2| > |U_{\min} - N/2|$ 이면 키의 후보를 $U_{\max}$의 값들로 보고 $K[\chi_K]=0$ 이라고 추측하고, 만일 $|U_{\max} - N/2| < |U_{\min} - N/2|$ 이면 키의 후보를 $U_{\min}$의 값들로 보고 $K[\chi_K]=1$ 이라고 추측한다. 즉, 알고리즘 2M은 $U_{(g,h)}$의 값들 중에서 $N/2$과 절대값 차이가 가장 큰 순서쌍 (g,h)를 키의 후보로 본다. 하지만 알고리즘 2M에서는 여러 선형 근사식의 오른쪽 부분 $K[\chi_K]$가 모두 동일하다는 가정하에서 이루어지고, 이에 따라 effective 키 비트들과 $K[\chi_K]$에 대한 추측을 통해 키의 한 비트 정보도 얻게된다.

이에 반해 알고리즘 E는 Step 2에서 $U_{g,h} = \sum_{i=1}^{n} a_i |T_{g,h}^i - N/2|$를 고려하고, $U_{g,h}$들 중에서 가장 큰 값 $U_{\max}$에 대한 (g,h)를 키의 후보로 본다. 즉 알고리즘 E도 effective 키 비트의 후보를 결정하는 면에서 $U_{(g,h)}$의 값들 중에서 $N/2$과 절대값 차이가 가장 큰 순서쌍 (g,h)를 키의 후보로 선택하게 된다. 알고리즘 E에서는 여러 선형 근사식에 대한 $K[\chi_K]$의 값이 0 혹은 1이라는 것은 중요하게 여기지 않는다. 그 이유는 다음과 같다. 만일 알고리즘 E에 이용되는 i번째 선형 근사식에 대한 $K[\chi_K]=1$ 이라고 가정해보자(실제 공격에서 $K[\chi_K]$의 값은 알 수 없다). 그러면 올바른 키 순서쌍 (g,h)에 대해 알고리즘 E의 Step 2에서 $T_{g,h}^i$값은 다른 키 후보 순서쌍의 $T_{g',h'}^i$값과 비교했을 때 가장 작은 값이 될 것이다. 이런 경우 $|T_{g,h}^i - N/2|$은 분명히 $|T_{g',h'}^i - N/2|$보다 크다. 따라서 올바른 effective 키 비트 후보에 대한 $|T_{g,h}^i - N/2|$은 $U_{(g,h)}$를 가장 큰 값이 되게 하며 이런 접근은 알고리즘 2M과 동일하다. 따라서 올바른 키 추측에 대해 알고리즘 2M과 알고리즘 E는 동일한 기지 평문에 대해 같은 성공 확률을 갖게 될 것이다.

본 논문에서는 [표 2]의 시뮬레이션 결과를 통해 이를 확인할 수 있었다. 물론 알고리즘 2M은 각 선형 근사식에 대한 키의 한 비트 정보 $K[\chi_K]$은 알 수 없을 것이다. 하지만 키의 한 비트 정보까지 얻기 위해 매우 크게 요구되는 계산 과정을 고려하기 보다는, 훨씬 적은 기지 평문과 여러 선형 근사식을 동시에 이용하여 effective 키 비트를 효율적으로 추측할 수 있다면 이것이 더 좋은 방법일 것이다.

4.5 LC, MLC, ELC의 비교

본 절에서는 시뮬레이션 결과를 통하여 LC, MLC, ELC를 서로 비교할 것이다. 다음의 식들은 DES를 대상으로 한 시뮬레이션에서 사용된 선형 근사식들이다.

$$P_H[7,18,29] \oplus P_L[15] \oplus C_H[15] \oplus C_L[7,18,29] \oplus F_4(C_L, K_4)[15] = k_1[22] \oplus k_3[22] \quad (\varepsilon = 0.03) \tag{9}$$

$$P_H[3,21] \oplus P_L[9] \oplus C_H[9] \oplus C_L[3,21] \oplus F_4(C_L, K_4)[9] = k_1[14] \oplus k_3[14] \quad (\varepsilon = 0.03) \tag{10}$$

$$P_H[7,18,24,29] \oplus F_1(P_L, K_1)[7,18,24,29] \oplus C_H[7,18,24] \oplus F_7(C_L, K_7)[7,18,24] = k_3[22] \oplus k_4[44] \oplus k_5[22] \quad (\varepsilon = 25 \times 2^{-12}) \tag{11}$$

$$C_H[7,18,24,29] \oplus F_1(P_L, K_1)[7,18,24] \oplus F_7(C_L, K_7)[7,18,24,29] = k_3[22] \oplus k_4[44] \oplus k_5[22] \quad (\varepsilon = 25 \times 2^{-12}) \tag{12}$$

[표 1]은 4 라운드 DES에 대해 (9)과 (10)을 각각 알고리즘 2에 적용했을 때와 (9), (10)을 동시에 알고리즘 E에 적용했을 때의 결과를 나타내고 있다. 여기서 E^2은 각 선형 근사식의 bias 제곱보다 두 배 큰 값이다. 분명히 우리는 같은 성공 확률에 대해 알고리즘 E가 알고리즘 2보다 더 적은 평문수를 요구한다는 것을 볼 수 있다.

[표 1] 4 라운드 DES에 대한 LC, ELC 비교

	실험 결과			
평문수	2222	4444	8888	17776
(9)알고리즘 2	52.7%	64%	75.5%	89.7%
(10)알고리즘 2	49.5%	75.1%	93.2%	97.6%
(9)&(10) 알고리즘 E	75.5%	92.3%	97.4%	99.9%

[표 2]는 7 라운드 DES에 대해 (11)와 (12)를 각각 알고리즘 2에 적용했을 때와 (11), (12)를 동시에 알고리즘 2M, 알고리즘 E에 적용했을 때의 결과를 나타내고 있다. 이 결과에서 보듯이 알고리즘 2M과 알고리즘 E는 서로 다른 알고리즘이지만 같은 결과를 나타내고 있다. 물론 [표 2]에서 사용된 두 식 (11), (12)는 χ_K가 서로 같은 두 개의 선형 근사식이므로, MLC를 적용하는 데에 추가적인 계산 과정이 필요 없었지만, 우리가 이제 살펴볼 8라운드 DES를 공격하는데 쓰이는 139개의 선형 근사식들은 모두 χ_K가 서로 다른 것들이다. 따라서 139개의 선형 근사식에 대해서는 MLC에 대해 추가적인 계산 과정이 요구되므로 알고리즘 2M을 적용하기 어렵지만 알고리즘 E를 이용한다면 139개의 선형 근사식 모두를 추가적인 계산 과정 없이 이용할 수 있다는 것을 보일 것이다.

[표 2] 7 라운드 DES에 대한 LC, MLC, ELC 비교

	실험 결과			
평문수	13422	26844	53688	107376
(10)알고리즘 2	3%	2%	1%	51%
(11)알고리즘 2	1%	1%	15%	45%
(11)&(12) 알고리즘 2M	4%	13%	46%	94%
(11)&(12) 알고리즘 E	3.8%	13.7%	45.3%	95%

V. DES 공격에 대한 새로운 결과

5.1 8 라운드 DES에 대한 공격

우리는 8 라운드 DES에 대한 마지막 라운드 S1-box의 effective 키 비트들을 추측하기 위해 $|\varepsilon_i|>0.00003$인 139개의 7 라운드 선형 근사식들을 찾아냈다. 우리가 오직 S1-box만 고려한 이유는 DES의 8개 S-box들 중에서 S1-box에 대한 $\sum_{i=1}^{139}\varepsilon_i^2\approx 0.00001838$이 다른 7개의 S-box들의 $\sum\varepsilon_i^2$보다 훨씬 크기 때문이다. 이 값(0.00001838)은 [5]에서 최량 확률을 갖는 7 라운드 선형 근사식의 ε^2보다 약 5.06배 더 큰 값이다.

[표 3]은 139개의 7 라운드 선형 근사식에 대한 알고리즘 E의 이론적인 기대치와 실제 실험값 사이의 비교를 말해주고 있다. [표 3]의 결과에서 보듯이 이론적인 기대치는 실험 결과와 거의 유사하고, 139개의 선형 근사식을 사용하는 알고리즘 E에서 요구하는 기지 평문의 수는 같은 성공 확률에 대해 알고리즘 2 [5]에서 필요로 했던 기지 평문의 수보다 약 5배 정도 감소된 값이다.

[표 3] 8 라운드 DES에 대한 ELC의 데이터 복잡도

평문수	이론치	실험치
108800	48.6%	47%
217600	78.5%	77%
435200	96.7%	97.3%
870400	99.9%	99.4%

5.2 16 라운드 DES에 대한 공격

B. Kaliski와 M. Robshaw는 각 라운드에서 하나의 active S-box를 포함하고 $|\varepsilon_i|>10^{-8}$인 10,006개의 14 라운드 선형 근사식을 찾았으며 이 경우의 $\sum_{i=1}^{10,006}\varepsilon_i^2\approx 1.23\times 10^{-11}$[2]이다. 그들은 이들 선형 근사식들을 동시에 적용할 수 있는 효율적인 방법이 존재하는지를 의문으로 남겨놨었다.

우리 또한 각 라운드에서 하나의 active S-box를 포함하고 $|\varepsilon_i|>10^{-8}$인 10,098개의 14 라운드 선형 근사식을 찾았으며 이 경우의 $\sum_{i=1}^{10,098}\varepsilon_i^2\approx 2.21\times 10^{-11}$이다. 비록 10,006개의 선형 근사식보다 더욱 많은 선형 근사식을 찾았지만, 16라운드 DES에 대해 S1-

box의 키 6 비트들과 S5-box의 키 6 비트들을 찾기 위해 실제 공격에 이용할 수 있는 선형 근사식은 10,098중에 몇 개 안된다는 것을 알 수 있었다. 왜냐하면 DES의 16 라운드에 대해 첫 라운드와 마지막 라운드에서 각각 하나의 active S-box들만 고려되게 선형 근사식을 구성하다보면, 비교적 높은 확률을 가지는 선형 근사식은 기껏해야 3개의 선형 근사식이 실제 공격에서 사용가능 했으며, 이 3개에 대한 $\sum_{i=1}^{3}\varepsilon_i^2 \approx 4.04\times10^{-13}$이므로, ELC의 알고리즘 E에 이들 선형 근사식을 동시에 적용하면, 같은 성공 확률에 대해 최량 확률 $1/2-1.19\times2^{-21}$을 갖는 하나의 선형 근사식에 대한 LC의 알고리즘 2에서 요구했던 기지 평문의 수 2^{43}보다 약 1.25배 만큼 기지 평문수를 줄일 수 있다고 추정할 수 있었다.

하지만 16 라운드 DES의 안전성을 더욱 줄일 수 있는 다른 방법이 존재한다. 우리는 기지 평문 공격 대신 선택 평문 공격을 채택할 수 있다. L. R. Knudsen과 J. E. Mathiassen은 선택 평문을 사용하는 방법[4]을 소개했다. 그들은 16 라운드 공격에 13 라운드 선형 근사식을 사용했고, 키 15 비트들을 찾기 위해 2^{42}개의 선택 평문을 필요로 하였다. 그들이 사용한 선형 근사식은 식 (13)을 따른다. 우선 식에서 각 라운드에 대한 확률값은 각각의 라운드 수를 i로 표현할 때 다음과 같다.

$$\begin{aligned} p_i &= 1, && i\in\{3,7,11,15\}\\ p_i &= 42/64, && i\in\{4,10,12\}\\ p_i &= 30/64, && i\in\{5,9,13\}\\ p_i &= 12/64, && i\in\{6,8,14\} \end{aligned}$$

식에서 $A=[7,18,24]$, $B=[7,18,24,29]$, $A\oplus B=[29]$, $D=[15]$은 M. Matsui[5]의 표현 방법으로서 마스크 되는 비트 위치를 나타내며, 3,…,15는 라운드 수를 의미한다.

$$\begin{array}{ll} 3: & ---\\ 4: & A\leftarrow D\\ 5: & D\leftarrow A\oplus B\\ 6: & B\leftarrow D\\ 7: & ---\\ 8: & B\leftarrow D\\ 9: & D\leftarrow A\oplus B\\ 10: & A\leftarrow D\\ 11: & ---\\ 12: & A\leftarrow D\\ 13: & D\leftarrow A\oplus B\\ 14: & B\leftarrow D\\ 15: & --- \end{array} \qquad (13)$$

이제 우리는 ELC와 L. R. Knudsen, J. E. Mathiassen의 방법을 접목시킨다. 그러기 위해 우리 또한 식 (13)과 비슷한 구조를 가지는 13 라운드 선형 근사식을 5개 추가로 찾을 수 있었다. 5개를 찾을 수 있었던 이유는 만일 (13)과 같은 구조의 선형 근사식을 하나 찾아낸다면 DES의 대칭성(Symmetry)에 의해 그것과 동일한 active S-box들과 키 비트들을 포함하고 있는 선형 근사식을 하나 더 쉽게 찾을 수 있다. 여기서 대칭성을 이용한다는 것의 의미는 하나의 선형 근사식이 찾아지면 그 선형 근사식의 평문과 암호문을 서로 바꿔서 대입해도 그 선형 근사식이 성립한다는 것이다. 하나의 예로서 (13)에 대해 대칭성 성질을 적용하면 식 (14)의 형태가 된다. (14)에서의 기호들은 (13)에서 사용된 기호들과 동일한 의미를 나타낸다.

$$\begin{array}{ll} 3: & ---\\ 4: & B\leftarrow D\\ 5: & D\leftarrow A\oplus B\\ 6: & A\leftarrow D\\ 7: & ---\\ 8: & A\leftarrow D\\ 9: & D\leftarrow A\oplus B\\ 10: & B\leftarrow D\\ 11: & ---\\ 12: & B\leftarrow D\\ 13: & D\leftarrow A\oplus B\\ 14: & A\leftarrow D\\ 15: & --- \end{array} \qquad (14)$$

물론 나머지 4개의 선형 근사식들은 (13),(14)와는 bias 값이 틀린 선형 근사식들이 된다. 본 논문에서는 16 라운드 DES를 공격하기 위해 예로서 제시한 (13)와 (14)에 추가하여 4 개의 선형 근사식을 동시에 ELC의 알고리즘 E에 동시에 적용한다. 참고로 각 선형 근사식 6개를 $L_1, L_1^*, L_2, L_2^*, L_3, L_3^*$로 표현하고 여기서 *의 의미는 하나의 선형 근사식에 대해 대칭성 원리에 의해 얻어진 다른 하나의 선형 근사식을 의미한다. 예를 들어 (13)에 대해 (13)*는 (14)를 의미한다. 또한 '*'이 붙은 식은 원래 식의 bias와 동일한 bias와 동일한 active S-box를 갖는다는 것을 유념하자. 그러면 L_1, L_1^* 두 선형 근사식은 각각 같은 bias $2^{-19.85}$, L_2, L_2^* 두 선형 근사식은 각각 같은 bias $2^{-25.95}$, 끝으로 L_3, L_3^* 두 선형 근사식은 각각 같은 bias $2^{-33.7}$를 갖는다. 따라서 이 값들을 모두 고려하여 공격에 필요한 평문수를 계산하면, 86% 성공 확률에 대해 알고리즘 E

는 $2^{40.6}$개 이하의 평문량을 요구한다. 이 결과는 DES 16 라운드에 대한 선형 공격 중에 가장 우수한 결과이다.

Ⅳ. 결 론

본 논문에서는 블록 암호를 공격하기 위해 다중 선형 근사식을 이용한 새로운 방법(ELC)을 제안했다. 이 공격은 선형 공격[5] 보다 더 적은 기지 평문수를 요구하고, 다중 선형 공격[2]에 비해 더 많은 선형 근사식을 가지고 실제 공격에 활용할 수 있었다.

본 논문에서는 ELC의 알고리즘 E에 대한 데이터 복잡도를 추정하기 위해서 E^2을 정의하고, E^2이 선형 공격(LC)에서 하나의 선형 근사식에 대한 bias의 제곱처럼 행동한다는 것을 추정할 수 있었으며 본 논문의 실험 결과를 통해 그것을 확인하였다.

본 논문의 추정에 기반하여 8, 16 라운드 DES에 대해 알고리즘 E를 적용했을 때의 요구되는 데이터 복잡도를 결정할 수 있다. 8 라운드 DES에 대해 95% 성공 확률을 얻기 위해 약 $2^{18.73}$개의 기지 평문이 필요했으며, 16 라운드 DES에 대해서는 86% 성공 확률을 얻기 위해 약 $2^{42.68}$개의 기지 평문이 필요했다.

만일 새로운 대안으로서 기지 평문들 대신 알려진 선택 평문들을 고려한다면, 알고리즘 E를 이용하여 16 라운드 DES를 공격할 때 요구되는 데이터 복잡도를 낮출 수 있었다. 이에 따라 본 논문의 방법과 L. R. Knudsen과 J. E. Mathiassen에 의해 소개된 방법[4]을 접목시킨다면 약 $2^{40.6}$개 이하의 선택 평문만 가지고도 86% 이상의 성공 확률을 가질 수 있다는 것을 보였다. 이 결과는 본 논문에서의 방법이 지금까지 DES에 대한 여러 공격들 중 가장 복잡도가 낮은 방법이라는 것을 말한다.

본 논문에서의 실험들은 ELC에서 알고리즘 E에 적용되는 여러 선형 근사식에 대한 E^2이 LC[5]에서 알고리즘 2에 적용되는 하나의 선형 근사식의 ε^2처럼 행동한다는 것을 보여준다. 공격을 하기 위해서는 먼저 성공 확률과 그 확률에 도달하기 위해 요구되는 평문수를 결정하는데 이러한 관점에서 본 논문에서 정의 내린 E^2이 중요한 역할을 한다.

참 고 문 헌

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

[2] B. R. Kaliski Jr and M. J. B. Robshaw, "Linear Cyptanalysis Using Multiple Approximations", *In Advances in Cryptology, CRYPTO'94*, LNCS 839, Springer-Verlag, pp. 26~39, 1994.

[3] L. R. Knudsen, "Truncated and higher order differential", *In Fast Software Encryption, 2nd International Workshop, FSE'95*, LNCS 1008, Springer-Verlag, pp. 196~211, 1995.

[4] L. R. Knudsen and J. E. Mathiassen, "A Chosen-Plaintext Linear Attack on DES", *In Fast Software Encryption, 7th International Workshop, FSE'2000*, LNCS 1978, Springer-Verlag, pp. 262~272, 2000.

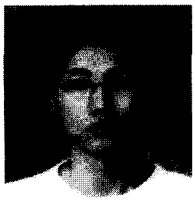
[5] M. Matsui, "Linear cryptanalysis method for DES cipher", *In Advances in Cryptology - Eurocrypt'93*, LNCS 765, Springer-Verlag, pp. 386~397, 1994.

[6] M. Matsui, "New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis", *In Fast Software Encryption, 3rd International Workshop, FSE'96* LNCS 1039, Springer-Verlag, pp. 205~218, 1996.

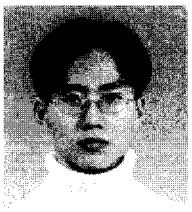
[7] B. R. Kaliski Jr and M. J. B. Robshaw, "Linear Cyptanalysis Using Multiple Approximations and FEAL", *In Fast Software Encryption, FSE 95*, LNCS 1008, Springer-Verlag, pp. 249~264, 1995.

[8] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", *In Advances in Cryptology - Crypto'94*, LNCS 839, Springer Verlag, pp. 1~11, 1994.

〈著者紹介〉

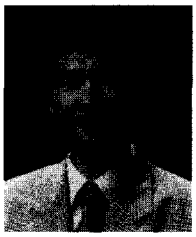


최 준 (Choi Jun) 정회원
2001년 2월 : 경희대학교 이학부 졸업
2003년 2월 : 고려대학교 정보보호학과 석사 수료 예정
2001년 2월~현재 : 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 암호학, 시스템파라미터분석, 정보보호



홍 득 조 (Deukjo Hong)
1999년 8월 : 고려대학교 수학과 학사
2001년 8월 : 고려대학교 수학과 석사
2001년 9월~현재 : 고려대학교 정보보호대학원 박사과정
〈관심분야〉 블록 암호 및 스트림 암호 분석 및 설계, 블록 암호 운영모드 분석

홍 석 희 (Seok Hie Hong) 정회원
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
〈관심분야〉 정보보호, 암호 알고리즘, 비밀키 암호 설계 및 분석, 패스워드 기반 프로토콜



이 상 진 (Sang-Jin Lee) 정회원
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 8월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월 : 한국전자통신연구소 선임 연구원
1999년 3월~현재 : 고려대학교 자연과학대학 부교수, 고려대학교 정보보호대학원 겸임교수, 고려대학교 정보보호기술연구센터 연구실장
〈관심분야〉 블록 암호 및 스트림 암호 분석 및 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석

임 종 인 (Jong-In Lim) 정회원
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1999년 2월~현재 : 고려대학교 자연과학대학 정교수, 한국통신정보보호학회 편집위원장, 고려대학교 정보보호대학원 원장, 고려대학교 정보보호기술연구센터 센터장
〈관심분야〉 블록 암호 및 스트림 암호 분석 및 설계, 암호 프로토콜, 공개키 암호 분석