情報保護學會論文誌
第19卷 第1號, 2009. 2

# 부정차분을 이용한 전력분석 공격의 효율 향상*

강 태 선,† 김 희 석, 김 태 현, 김 종 성, 홍 석 희‡

고려대학교 정보경영공학전문대학원

# Performance Improvement of Power Attacks with Truncated Differential Cryptanalysis*

Taesun Kang,† Hee-Seok Kim, Tae-Hyun Kim, Jong-Sung Kim, Seok-Hie Hong‡

Graduate School of Information Management and Security, Korea University

## 요 약

1998년 Kocher 등이 블록암호에 대한 차분전력공격(Differential Power Attack, DPA)을 발표하였는데 이 공격으로 스마트 카드와 같이 위조방지가 되어있는 장비에서도 암호알고리즘 연산에 사용된 암호키를 추출할 수 있다. 2003년 Akkar와 Goubin은 DES와 같은 블록암호의 전 후반 3~4 라운드의 중간값을 마스킹 값으로 랜덤화해서 전력분석을 불가능하게 하는 마스킹 방법을 소개하였다. 그 후, Handschuh 등이 차분분석을 이용해서 Akkar의 마스킹 방법을 공격할 수 있는 방법을 발표하였다. 본 논문에서는 부정차분 분석을 이용해서 공격에 필요한 평문수를 Handschuh 등이 제안한 공격방법보다 효과적으로 감소시켰으며 키를 찾는 마지막 절차를 개선하여 공격에 사용되는 옳은 입력쌍을 선별하기위한 해밍웨이트 측정시 발생할 수 있는 오류에 대해서도 효율적인 공격이 가능함을 증명하였다.

## ABSTRACT

In 1998, Kocher et al. introduced Differential Power Attack on block ciphers. This attack allows to extract secret key used in cryptographic primitives even if these are executed inside tamper-resistant devices such as smart card. At FSE 2003 and 2004, Akkar and Goubin presented several masking methods, randomizing the first few and last few(3~4) rounds of the cipher with independent random masks at each round and thereby disabling power attacks on subsequent inner rounds, to protect iterated block ciphers such as DES against Differential Power Attack. Since then, Handschuh and Preneel have shown how to attack Akkar's masking method using Differential Cryptanalysis. This paper presents how to combine Truncated Differential Cryptanalysis and Power Attack to extract the secret key from intermediate unmasked values and shows how much more efficient our attacks are implemented than the Handschuh-Preneel method in term of reducing the number of required plaintexts, even if some errors of Hamming weights occur when they are measured.

**Keywords** : Masking Method, Side channel attack, Des, Truncated Differential Cryptanalysis

접수일(2008년 9월 17일), 게재확정일(2008년 11월 24일)

* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21사업'의 지원비를 받았음

† 주저자, boj12@naver.com

‡ 교신저자, hsh@cist.korea.ac.kr

## Ⅰ. 서 론

1998년 Kocher 등이 발표한 블록암호에 대한 전력분

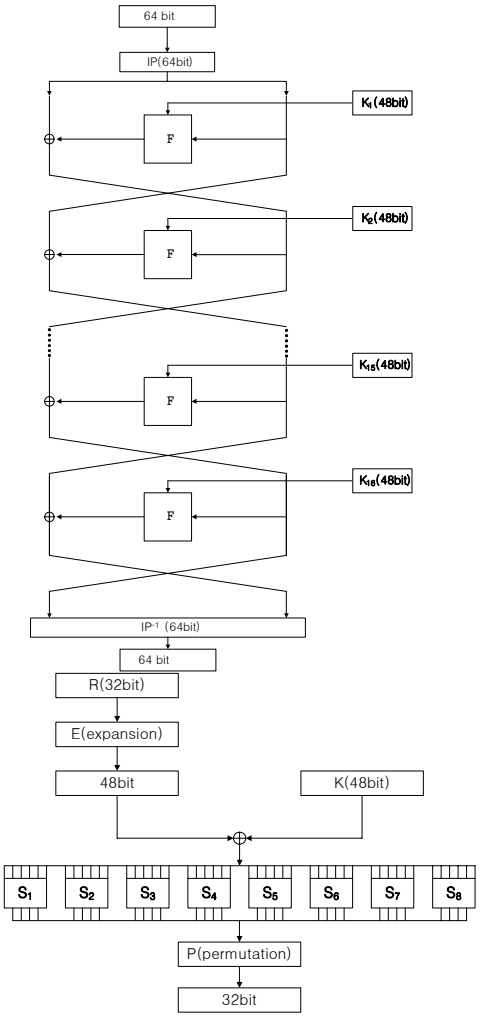
석 공격[1]을 통해 암호설계자가 예측하지 못한 부가적인 정보들을 이용해 암호분석이 가능하다는 것이 알려졌다. 이후 많은 논문에서 알고리즘이 수행되면서 소비되는 전력 또는 방출되는 전자기파 등을 분석하여 비밀키를 복구하는 공격방법이 제안되었으며 이에 대한 대응방안으로 알고리즘 수행순서를 랜덤하게 하는 방법과 알고리즘 수행 도중에 난수로 중간값을 마스킹하는 방법들이 발표되었다[2]. FSE 2003과 2004에서 Akkar와 Goubin 등은 알고리즘이 수행되는 라운드의 전·후반 일부 라운드에 마스킹을 적용하여 전력분석 공격(Differential Power Analysis, DPA)에 대응하는 방안(Unique Masking Method)을 발표하였다[3]. 이에 대해서 SAC 2006에서 Handschuh와 Preneel은 마스킹이 적용되지 않는 내부 라운드에 차분분석(Differential Cryptanalysis, DC)을 적용하여 Unique Masking Method로 마스킹된 DES 알고리즘의 라운드 키를 찾는 공격방법을 제안하였다[4]. CT-RSA 2006에서 Schramm와 Paar는 고차분전력분석(High Order Differential Power Analysis, HODPA)에 안전한 마스킹 대응기법(High Oder Masking)을 발표하였는데[9] 이 기법은 매 라운드마다 별도의 마스킹 테이블을 만들어야 하므로 하드웨어적인 비용이 많이 소요된다. 그래서 하드웨어적인 제한요소가 있는 경우에 전 후반 일부 라운드만을 마스킹하므로서 전력분석공격에 대한 대응기법으로 적용될 수가 있다. 본 논문에서는 부정차분(Truncated Differential Cryptanalysis, TDC)을 적용하여 Unique Masking Method 및 High Order Masking이 축소된 라운드에 적용된 DES 알고리즘에서 Handschuh와 Preneel이 제안한 방법보다 적은 평문쌍으로 DES 알고리즘 라운드 키를 찾을 수 있는 방안을 제시하였으며 해밍웨이트 측정시 발생할 수 있는 오류를 고려해서도 제안 공격방법이 좀더 효율적인 공격방법인 것을 보였다: Handschuh-Preneel 공격은 8×65,000개의 평문에 대해 전력 측정값이 필요하다면, 본 논문의 공격은 16×120개의 평문에 대한 전력 측정값이 필요하다. 본 논문구성은 다음과 같다. 2장에서는 DES에 대한 Unique Masking Method을 설명하고 3장에서는 차분특성을 이용한 기존 전력분석 공격방법을 소개하고 4장에서는 부정차분을 이용한 공격방법을 제안하였고 5장에서는 기존 공격방법과 제안 공격방법의 효율성을 비교하였다.

## Ⅱ. DES에 대한 Extended Unique Masking Method

### 2.1 DES의 구조 및 기호

1977년 미연방표준 암호 알고리즘으로 공표된 DES의 기본 제원은 다음과 같다.

- 입출력 크기: 64bit
- 키 크기: 56 bit
- 비선형 함수로 구성된 S box 사용
- 라운드 수: 16 라운드




[그림 1] DES 기본 알고리즘 및 F 함수

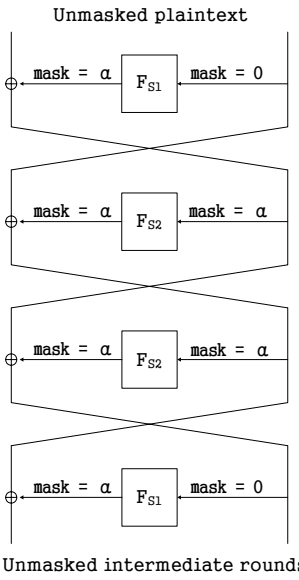
**기호 정리**

- P': 제한된 라운드의 평문 입력값의 차분
- C': 제한된 라운드의 암호문 출력값의 차분
- δ: S box 입력(6 bit) 차분
- MSE: 평균 제곱 오차(Mean Square Error)로 두 값의 차이를 극대화하기 위해 차이 값을 제곱한다.
- a', b', c', d': DES F 함수 입력차분
- a'', b'', c'', d'': DES F 함수 출력차분

### 2.2 DES에 대한 Extended Unique Masking Method

Akkar와 Goubin 등이 제안한 Extended Unique Masking Method는 DES[5]와 같은 Feistel 구조의 암호알고리즘과 AES[6]와 같은 Substitution Permutation Network 구조를 가진 암호알고리즘의 전·후반부 일부 라운드의 마스킹에 적용될 수 있다. [그림 2]의 Feistel 구조의 DES에 적용하기위해서 F 함수내의 S box에 대해 식 (1)과 같은 두 가지의 새로운 S box 함수를 도입시키면 4 라운드까지 중간 값들을 안전하게 마스킹 할 수 있다. $\alpha$ 값은 32 bit의 input과 output의 mask 값이며 암호화 알고리즘 실행시마다 달라진다.

$$\forall x \in \{0,1\}^{48},\ S_1(x) = S(x) \oplus P^{-1}(\alpha) \qquad (1)$$
$$\forall x \in \{0,1\}^{48},\ S_2(x \oplus E(\alpha)) = S(x) \oplus P^{-1}(\alpha)$$




[그림 2] 마스킹이 된 DES 알고리즘의 전반부 4라운드

식 (1)을 통해서 DES 알고리즘의 중간 결과값들은 랜덤값($\alpha$)으로 마스킹이 되기 때문에 전력분석을 통해 구한 데이터 값이 의미 없어지게 된다.

## Ⅲ. 차분특성을 이용한 전력분석 공격 방법

Biham과 Shamir가 제안한 차분분석(Differential Cryptanalysis, DC)[7]은 DES 알고리즘의 S box의 입출력쌍이 어떤 확률을 가지고 정해진 차분을 따라서 발생하는 취약점을 이용한 분석방법이다. Handschuh와 Preneel이 제안했던 차분분석을 이용한 전력분석 공격방법은 전후 일부 라운드에 Unique Masking Method가 적용된 DES 알고리즘에서 마스킹이 적용되지 않은 라운드에서의 차분특성을 만족시키는 평문의 해밍웨이트 값을 이용한다. 그 정보에 부합하는 옳은 입력문쌍(right pair)을 해밍웨이트 값으로 선별하고 그 평문들의 해밍웨이트 정보를 이용해서 해당 라운드 키를 추출하는 공격방법이다. [그림 2]에서 첫 번째 라운드부터 네 번째 라운드까지 마스킹이 되어 있는 경우 네 번째 라운드의 F 함수 출력 값이 세 번째 라운드의 입력 값과 xor 해서 마스킹이 없어지게 된다. 결과적으로 네 번째 라운드 결과 값은 평문의 고유한 차분특성을 가지게 되고 이와 동일하게 네 번째 라운드 F 함수 입력값에도 마스킹 값이 없어지기 때문에 해밍웨이트 측정을 통해 의미있는 결과를 얻을 수 있다. 공격의 전제조건으로 부채널 분석으로 S box 입력값의 해밍웨이트 측정이 가능하다고 전제한다. 평문의 고유한 차분특성과 전력분석에서 얻을 수 있는 정보를 통해 해당 라운드 키를 추출할 수 있다.

### 3.1 옳은 입력문쌍 후보 선별

4 라운드의 차분특성을 $P' = 405C0000\ 04000000$,
$C' = 04000000\ 00540000$라 할 때 5 라운드 F 함수 입력 S box에서 S2 box를 제외한 나머지 S box 들의 차분 값은 '0'이 되므로 이 조건을 만족시키는 4 라운드 S box 입력문쌍을 공격에 사용할 옳은 입력문쌍으로 선별한다. 이 차분에 대해 4라운드 F 함수의 S3 box와 S4 box의 차분특성은 0xA → 0x01와 0x28 → 0x00가 된다.(4 라운드 차분특성의 확률은 $3.8\times10^{-4}$이다). 나머지 S box들의 차분특성은 0x00 → 0x00이다. 우선 S3

box input 차분이 0xA이기 때문에 이를 해밍웨이트 관점에서 보면 다음과 같은 관계식이 성립한다.

$$\text{입력차분: } \delta = 001010 \Rightarrow hwt(\delta) = 2 : hwt(x_i) = hwt(x_i') \pm 2 \text{ 또는 } hwt(x_i) = hwt(x_i'), \quad (x_i, x_i' : sbox \text{ 입력 평문쌍}). \quad (2)$$

공격에 사용될 옳은 입력문쌍은 식 (2)을 만족시킨다.

### 3.2 공격 방법

S box의 입력차분의 해밍웨이트 값이 2 일 때 식 (2)가 성립하고 식 (2)을 만족시키는 평문쌍의 해밍웨이트 종류는 총 17 가지가 된다. 공격 대상인 네 번째 라운드 S3 box의 경우 차분특성의 확률이 10/64이므로 이 차분특성을 만족시키는 입력쌍($x_i$, $x_i'$)은 10개가 존재하고 이 입력쌍의 해밍웨이트 분포는 각 S box의 라운드 키 값에 대해서 다른 S box와 구별되는 분포를 갖게 된다. 즉, S box 차분특성에 대해서 독특한 분포를 갖게 되고 이 독특한 분포를 이용해서 라운드 키를 복구할 수 있다. S4 box의 경우 차분특성 확률은 16/64이고 이 차분특성을 만족시키는 입력쌍($x_i$, $x_i'$)은 16개가 존재한다.

[표 1]를 보면 S box 입력차분(δ)으로 가능한 17가지 해밍웨이트 쌍에 대해서 네 번째 라운드의 S3 box 입력 평문쌍의 해밍웨이트 분포와 S4 box 입력 평문쌍의 해밍웨이트 분포가 서로 다름을 볼 수 있다. 옳은 입력문쌍 후보 선별시 해밍웨이트 값만을 이용하기 때문에 틀린 입력문쌍이 옳은 입력문쌍으로 잘못 선별될 수도 있으나 네 번째 라운드의 공격대상 S box의 입출력 차분특성을 만족시키는 옳은 입력문쌍은 [표 1]에 나와 있는 S box 입력평문쌍의 해밍웨이트 분포와 유사하게 된다. 다수의 선별한 평문쌍 후보가 수집되고 그 평문쌍의 해밍웨이트 값 분포가 구해지면 이 평문쌍 분포와 공격대상인 S box 입력문쌍과 모든 가능한 키 값(0~63)과의 xor 값의 해밍웨이트 분포를 서로 비교해서 최대한 유사한 분포를 보이는 키를 옳은 키 값으로 본다. 비교시 평균제곱오차(Mean Square Error) 방법을 이용하여 최소값을 갖는 키 값을 옳은 키로 추출한다.

$MSE_{k^*}$= Σ{(모든 키 값($k^*$)과 S box 입력 쌍과의 xor 결과 값의 해밍웨이트 분포) - (선별된 옳은 입력문쌍에 대한 해밍웨이트 분포)}$^2$ , $k^*$: 가능한 모든 키 값(0 ~ 63). (3)

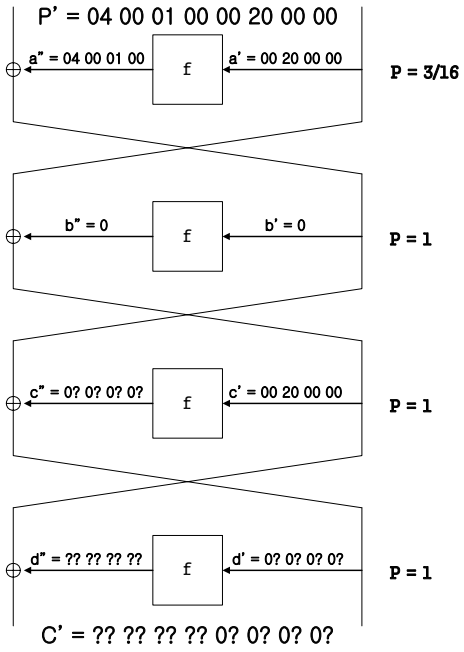
다른 차분특성을 이용하여 다른 S box에 대한 공격을 동일한 방법으로 반복하면 라운드 키를 찾을 수 있다.

[표 1] 입력 차분특성의 해밍웨이트 값에 대한 S3, S4box의 입력 평문쌍의 해밍웨이트값 분포

| | (hwt($x_i$), hwt($x_i'$)) | S3 box | S4 box |
|---|---|---|---|
| 1 | (0, 0) | 0 | 0 |
| 2 | (1, 1) | 0 | 0 |
| 3 | (2, 2) | 2 | 4 |
| 4 | (3, 3) | 0 | 4 |
| 5 | (4, 4) | 2 | 0 |
| 7 | (5, 5) | 2 | 0 |
| 7 | (6, 6) | 0 | 0 |
| 8 | (0, 2) | 0 | 0 |
| 9 | (1, 3) | 0 | 0 |
| 10 | (2, 4) | 1 | 2 |
| 11 | (3, 5) | 1 | 2 |
| 12 | (4, 6) | 0 | 0 |
| 13 | (2, 0) | 0 | 0 |
| 14 | (3, 1) | 0 | 0 |
| 15 | (4, 2) | 1 | 2 |
| 16 | (5, 3) | 1 | 2 |
| 17 | (6, 4) | 0 | 0 |
| 합계 | | 10 | 16 |

## Ⅳ. 부정차분을 이용한 제안 공격 기법

차분특성과 전력분석을 결합한 공격방법은 제한된 라운드에 적용된 어떠한 마스킹 기법에도 적용이 가능하기 때문에 상당히 유효한 공격 방법이기는 하지만 하나의 S box에 대한 옳은 키 값을 추출하기 위해서는 $2^{16}$개의 평문쌍이 필요하게 되고 이 평문쌍에 대한 해밍웨이트 측정을 통해 공격에 필요한 옳은 입력문쌍 후보를 선별해야한다. 이를 실제 실험과 결부시켜 생각해 보면 $2^{16}$쌍의 평문에 대해 해밍웨이트를 오류없이 정확히 측정한다는 것이 쉽지 않은 일임을 알 수 있다. 따라서 필요한 평문쌍의 개수를 최소화할 필요가 있으며 부정차분특성을 이용하면 필요한 평문수를 최소화할 수 있다.




[그림 3] 4라운드 F 함수 S3 box에 부정차분 적용

### 4.1 부정차분 공격

부정차분 공격은 1994년 Lars R. Knudsen에 의해 제시된 공격 방법[8]으로 일반적인 차분 공격은 특정 라운드 후의 출력값 비트 모두를 예측하여 공격하는데 실제 공격상에는 구해진 출력값의 모든 비트 정보들을 사용하지 않고 특별한 경우에는 한 비트의 정보만 알아도 공격에 이용할 수 있음을 알 수 있다. 이러한 이유로 출력값 모두를 예측하지 않고 공격에 필요한 비트 정보들만을 예측하여 공격하는 개념이 부정 차분 공격이다. 이렇게 필요한 특정 비트의 정보만을 예측하기 때문에 필요한 비트 외의 정보는 랜덤한 값을 가지는 차분 특성 모두가 이 공격에 사용 가능하고 이런 관점에서 부정 차분 특성은 차분 특성 중 필요한 비트들의 값이 같은 여러 차분 특성들을 합쳐 놓은 것으로 볼 수 있다. 따라서 차분 공격에 사용되는 차분 특성보다 훨씬 높은 확률의 특성을 찾을 수 있다. 결국 확률이 높아짐에 따라 공격에 요구되어지는 선택 평문의 수가 현저히 줄어드는 장점이 있다. [그림 3]에서 3라운드 S3 box 출력값으로 '0011' 등 11가지의 모든 출력값을 고려하기 때문에 '0?0?0?0?'[1]의 출력차분이 생기며 차분확률은 1이 된다.

### 4.2 DES에 대한 부정차분 특성을 이용한 전력분석 공격

#### 4.2.1 부정차분을 이용한 평문 선별

DES에 대한 부정차분 특성을 고려할 때 공격하려는 S box에만 '0'이 아닌 차분이 오게 만들어서 차분 확률을 높인다. 공격대상인 첫 번째 라운드의 S3 box에 대해서는 P = 3/16 확률을 갖는 '04000100 00200000' 차분을 적용하고 3 라운드 F 함수 출력의 부정차분을 고려하면 총 11가지의 부정차분 출력이 존재하게 되고 이 출력값들이 다음 4 라운드 입력값에 각각 영향을 주게 된다([표 2] 참조). 11가지의 출력차분으로 인해 4 라운드 F 함수의 S1 box와 S3 box의 입력 차분이 항상 '0'이 되며 나머지 S box의 차분은 '0' 또는 '1'이 존재한다. 이 조건을 가지고 공격에 사용될 평문쌍을 선별한다. 즉, 4 라운드 F 함수의 각각 S box 입력값의 해밍웨이트를 측정하여 S1 box와 S3 box의 해밍웨이트 값이 동일하고 나머지 S box의 해밍웨이트 값이 동일하거나 ±1의 차이를 보일때 1 라운드 입력 평문쌍을 공격에 사용될 옳은 입력문쌍 후보로 선별한다.

#### 4.2.2 선별된 평문쌍을 이용한 키 복구

평문 입력 차분 '04000100 00200000'이 1라운드 S3

[표 2] 3 라운드 S3 box의 11가지 출력차분에 대한 4 라운드 S box 입력쌍의 해밍웨이트 분포

| | S3 box 출력차분 | 4라운드 S box별 입력쌍의 해밍웨이트 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
| 1 | 0011 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0101 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0110 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0111 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 5 | 1001 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 1010 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 7 | 1011 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8 | 1100 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 9 | 1101 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 10 | 1110 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 11 | 1111 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

1) '?' 표시는 출력차분의 종류에 따라서 '1' 또는 '0' 값을 가짐을 표시

box에 대해 0x04 → 0x9의 차분을 갖게 하는데 이 차분을 만족시키는 1 라운드 S3 box 입·출력 후보쌍들은 총 12개가 된다. 이때 S box 입력쌍들과 위 4.2.1 절에서 부정차분을 통해 선별된 평문과의 XOR 값들 중 적어도 하나는 옳은 키 값이 나오게 된다. 부정차분을 만족시키는 여러 개의 평문을 가지고 이 과정을 반복하면 옳은 키는 매번 나오고 옳지 않은 키는 옳은 키를 제외한 63개 키에 랜덤하게 분포하게 되고 위 공격을 반복한 후 제일 많이 나온 키를 옳은 키로 선택한다.

선별된 옳은 입력문쌍$(x_i, x_i^{'})$가 있고 공격 대상인 첫 번째 라운드 S3 box 입력 차분을 만족시키는 평문쌍$(P_j, P_j^*)$이 있을 때 다음과 같이 정의한다.

**(정의)** $T_j = \{(P_j, P_j^*) \oplus (x_i, x_i^{'}) \mid 0 \le i \le 11\}$

**(정리)** S box 입력 차분을 만족하는 평문쌍$(P_j, P_j^*)$에 대하여 $T_j$에는 항상 옳은 키 값이 존재한다.

옳은 입력문쌍은 주어진 S box 입력차분을 반드시 만족하게 되는데 이때 S box 입력차분이 어떤 것인지 알 수는 없지만 S box 입력 차분을 만족하는 후보 중에 하나가 옳은 입력 차분이 되고 옳은 입력문쌍과 모든 가능한 차분 입력쌍의 xor 결과 중 하나는 항상 옳은 키 값을 가리키게 되므로 $T_j$에는 옳은 키 값이 항상 존재한다.

### 4.3 시뮬레이션

공격방법을 검증하기 위해서 PC를 이용한 시뮬레이션(Matlab 활용)을 수행하였다. DES와 공격 알고리즘(right pair filtering, key guessing)은 모듈별로 작성하였다. 시뮬레이션 전제조건으로 F 함수의 S box 입력값에 대한 해밍웨이트 측정은 가능하다고 하며 입력 평문 생성시 랜덤함수를 사용한다. 각 S box에 대해서 1,000회 반복하고 3% 이내의 오차에서 시뮬레이션[2]하였다.

#### 4.3.1 차분을 이용한 기존 공격

3.1절에서 사용한 차분(P' = 405C0000 04000000, C' = 04000000 00540000)을 이용하면 5 라운드 F 함수에서 S2 box를 제외한 나머지 S box들의 차분 값은 '0'이 되므로 이 조건을 만족시키는 4 라운드 입력 평문쌍을 공격에 사용할 옳은 입력문쌍으로 보고 20 ~ 30개 선별한다. 다음 단계로 4 라운드 S3 box 차분 특성에 따라 S3 box 입력 쌍과 모든 가능한 키 값(0~63)과의 xor 결과값의 해밍웨이트 분포를 구한다. 마지막으로 옳은 쌍으로 선별되는 평문쌍들의 4 라운드 S3 box 입력값의 해밍웨이트 분포를 구하고 MSE 방법을 이용하여 옳은 입력문쌍의 분포와 S box 차분특성을 만족시키는 입력쌍과 가능한 모든 키 값(0~63)의 xor 결과값의 분포를 비교하여 최소가 될 때의 키 값을 옳은 키로 정한다.

#### 4.3.2 부정차분을 이용한 제안 공격

기존 공격방법의 차분 대신에 필요한 평문수를 최소화하기 위해 별도의 차분을 공격대상 S box별로 구한다([표 3] 참조). 공격 대상이 되는 S box에만 입력 차분이 존재하도록 하고 2 라운드의 차분 확률값이 '0'이 될 수 있도록 평문입력의 상위 32bit 차분값을 하위 32bit 차분값의 1라운드 F 함수의 출력값으로 한다([그림 3] 참조). 공격에 사용될 옳은 입력문 선별 단계에서

[표 3] 부정차분을 이용한 공격시 각 S box별 입력 차분 및 필요 평문쌍 개수

| 공격 대상 S box | 입력 차분 / 차분 확률 | 4 라운드 입력 차분이 '0'이 되는 S box | 필요한 평문쌍 개수 |
|---|---|---|---|
| S1 | 00808202 60000000 / 14/64 | S1, S5 | $2^7$ |
| S2 | 40004010 02000000 / 14/64 | S2, S6 | $2^7$ |
| S3 | 04000100 00200000 / 12/64 | S1, S3 | $2^7$ |
| S4 | 80401000 00040000 / 8/64 | S2, S4 | $2^7$ |
| S5 | 00040080 00002000 / 10/64 | S5, S8 | 27 |
| S6 | 00200008 00000400 / 16/64 | S4, S6 | 26 |
| S7 | 00100001 00000060 / 14/64 | S5, S7 | 27 |
| S8 | 00020820 00000002 / 12/64 | S3, S8 | 27 |

2) 기준 평문개수를 가지고 1,000회 반복시 3% 이상 즉, 30회 이상 키를 찾지 못하면 평문개수를 증가시켜서 다시 1,000회 반복한다. 3% 이내에서 오차가 발생하면 그때의 평문개수를 최소 필요 평문개수로 본다.

는 3라운드 공격대상 S box 입력 차분값(6bit)에 대해서 출력 차분값(4bit)이 될 수 있는 모든 값을 구한다. 출력 차분에 대해 4 라운드 입력값의 차분값을 모두 구하고 차분 특성을 이용해서 공격에 사용할 옳은 입력문쌍을 선별한다. 마지막으로 옳은 입력문쌍과 S box 입력차분을 만족시키는 평문쌍들과의 xor 결과값들을 카운트한다. 이 단계를 반복해서 최대한 많이 카운트 받은 값을 옳은 키 값으로 한다.

## Ⅴ. 효율성 비교

### 5.1 S box 라운드 키 복구에 필요한 평문 개수

[표 4] 키 복구에 필요한 평문쌍 및 옳은 입력문쌍 개수

| | 기존 방법 | 제안 방법 |
|---|---|---|
| 평문쌍 개수 | $2^{16}$ | $2^{6}$ ~ $2^{7}$ |
| 옳은 입력문쌍 개수 | 26 | 6 ~ 8 |

필요한 입력평문 개수를 차분 확률을 통해 대략적으로 구해본다. 우선 차분을 이용한 기존 공격방법을 살펴보면 Akkar 등이 사용한 차분 확률이 $3.8\times10^{-4}$으로 차분확률만을 고려할 때 20개 정도의 공격에 필요한 평문쌍을 구하기 위해서는 대략 $2^{16}$개의 평문쌍이 필요하나 시뮬레이션시 해밍웨이트 필터링을 만족시키는 평문쌍 중에 대략 15% 정도가 차분특성을 만족시키지 못하는 잘못된 평문쌍이기 때문에 공격에 필요한 평문쌍을 구하기 위해서는 이론치보다 더 많은 평문쌍을 요구한다.

부정차분을 이용한 제안 공격방법은 공격하려는 S box에 대해 개별적인 차분을 적용하기 때문에 기존 공격방법과의 비교를 위해 S3 box를 예를 들면 차분확률은 $1.9\times10^{-1}$ 이 되고 필요한 입력 평문쌍은 대략 $2^{6}$이 된다. 실제 공격에서는 4 라운드 입력값의 해밍웨이트 값으로 필터링을 하기 때문에 S1 box와 S3 box의 차분이 '0'이 될 확률은 $5.1\times10^{-2}$ 이 된다.

차분특성을 이용한 기존 공격방법은 차분 확률이 $3.8\times10^{-4}$로 작기 때문에 필요한 평문이 기분적으로 많이 필요하게 되지만 부정차분을 이용하는 제안공격 방법은 차분 확률이 $1.9\times10^{-1}$로 기존 공격방법에 비해 매우 높기 때문에 필요 평문이 상대적으로 적다.

### 5.2 전체 키 복구에 필요한 평문 개수

기존 공격방법으로 2개의 S box의 라운드 키를 복구하는데 대략 $2^{16}$개의 평문쌍이 필요하고 8개의 모든 S box의 라운드 키를 복구하는데 $8\times2^{16}$개 정도의 평문이 필요하다. 이때 차분의 특성으로 정확한 키 k와 $k\oplus\delta$ 두 개의 키 후보값이 추출되기 때문에 정확한 k 값을 찾기 위해 $2^{8}$의 전수조사가 추가된다. 그러면 전체 키 64bit 중 48bit를 복구하게 되고 DES 키 스케줄에서 라운드 키 선택치환(PC2)과 키 선택 치환(PC1)의 인버스 값을 구하고 전수조사를 고려하면 필요한 평문수는 $2^{19}$개고 전수조사 횟수는 $2^{16}$번이 된다. 부정차분을 이용한 제안공격방법은 하나의 S box를 공격하기 위해 $2^{8}$개의 평문이 필요하므로 라운드 키 복구를 위해서는 $2^{11}$개의 평문이 필요하고 전체 키 복구를 위해서는 평문개수 $2^{11}$ 개, 전수조사 횟수 $2^{16}$번이 필요하게 된다. 결과적으로 전수조사 횟수는 동일하나 기존 공격대비 상당히 적은 평문개수로 전체 키를 찾을 수 있다.

### 5.3 해밍웨이트에 대한 오류[3] 발생시 필요 평문 개수

공격에 필요한 평문을 선별하기 위해 F 함수의 각 S box에 입력값의 정확한 해밍웨이트 값을 측정해야하는데 noise 등의 이유로 정확한 해밍웨이트 값을 측정하지 못하는 경우가 발생할 수 있고 이는 실험 결과에 영향을 미친다. 오류 발생에 따라 필요한 평문 개수는 증가하게 되고 오류 발생률에 증가함에 따라서 키를 복구하지 못할 수도 있다. 기존 차분특성을 이용한 공격방법의 경우 오류 발생률이 40% 이상에서는 오류를 고려하지 않았을 때의 평문 개수(6,5000개) 보다 2배 많은 평문이 입력되어도 키를 복구하지 못함을 알 수 있지만 제안공격 방법에서는 50% 오류를 고려해서도 키를 복구할 수 있음을 시뮬레이션 결과 알 수 있다. 그 이유는 제안 공격방법이나 기존 공격방법이나 동일하게 잘못된 입력문을 옳은 입력문으로 가정하고 공격에 사용하나 기존 공격방법은 잘못된 평문쌍이 랜덤하게 올바른 입력쌍의 해밍웨이트 분포에 영향을 주어 식 (3)의 MSE

3) 오류는 해밍웨이트를 잘못 측정하므로서 선별조건을 만족시키는 입력문쌍임에도 만족시키지 못하는 경우로 분류되거나 선별조건을 만족시키지 못하는 입력문쌍임에도 옳은 입력문쌍으로 선별되어 공격에 사용되는 경우를 말하며 이는 라운드 키를 추측하는 과정에서 치명적인 오류를 발생시킨다. 예를 들어 40% 오류율이라 함은 선별조건을 만족시키지 못하는 40%의 잘못된 입력문쌍이 옳은 입력문쌍으로 선별되는 것을 의미한다.

방법 적용시 옳은 키 값임에도 최소값이 나오지 않게 된다. 그러나 제안기법은 옳은 입력문은 항상 옳은 키 값을 카운트하고 옳지 않은 입력문은 잘못된 키 값을 램덤하게 카운트하기 때문에 오류률이 증가하는 것에 비례해서 입력 평문 개수가 증가한다면 옳은 키 값을 찾을 수 있기 때문이다.

[표 5]에서 보면 오류율이 점차 증가함에 따라 필요한 평문수도 같이 증가함을 보이고 있다.

[표 5] 해밍웨이트 측정시 오류 발생에 대해 키 복구에 필요한 평문쌍 개수(S3 box의 예)

| 오류율 | 기존 방법시 필요 평문쌍 개수 | 제안 방법시 필요 평문쌍 개수 |
|---|---|---|
| 10% | 66,000 | 115 |
| 20% | 75,000 | 120 |
| 30% | 85,000 | 135 |
| 40% | - | 172 |
| 50% | - | 180 |

## Ⅵ. 결 론

본 논문에서는 전력분석 공격방법과 부정차분특성을 이용하여 Masking Method가 적용된 Feistel 구조의 암호 알고리즘(DES)의 라운드 키 값을 찾을 수 있음을 보였다. 이러한 공격방법의 성공여부는 공격에 필요한 옳은 입력문쌍을 어떻게 정확하게 선별하는가와 공격에 필요한 평문쌍을 최소화하여 해밍웨이트 측정시 발생할 수 있는 오류를 얼마나 많이 줄일 수 있느냐가 관건인데 이 두가지 조건을 살펴볼 때 옳은 평문쌍을 선별하는 과정은 어느 공격 기법이나 해밍웨이트 측정과정이므로 유사하다고 할 수 있지만 필요 평문수에 있어서는 기존 논문의 공격방법으로는 2개의 S box의 라운드 키를 찾기 위해 $2^{16}$개의 평문쌍이 필요한데 본 논문에서 제안하는 공격방법으로는 $2^{7}$쌍의 평문이 필요하므로 공격이 성공하기 위해 필요한 평문수에서는 상당히 차이가 남을 알 수 있다.

본 논문에서는 실질적인 실험을 고려하여 옳은 입력문 선별을 위해 해밍웨이트 측정시 발생할 수 있는 오류를 고려하였는데 기존 논문의 공격방법의 경우에는 해밍웨이트 측정시 오류가 증가하는 것에 따라 필요한 평문이 증가하다가 어느 정도 이상의 오류율(40% 이상)이 발생하면 키를 못 찾는 반면, 본 논문에서 제안하는 공격방법에서는 50% 오류율에서도 키를 찾는 것을 증명하였다. 이는 키를 찾는 방법이 기존 공격방법에 비해 유연성이 있음을 보여주는 것으로 본 논문에서 제안하는 공격 기법이 좀 더 효율적임을 의미한다.

본 논문에서 제안한 공격방법에 대한 대응방안은 알고리즘 수행과 관련된 중간값을 측정하지 못하게 하는 것인데 이는 모든 중간값에 마스킹 값을 도입하거나 알고리즘 오퍼레이션을 랜덤화하여 옳은 입력문쌍을 선별하기위한 측정 지점을 잡지 못하게 하는 방안들이 있을 수 있으나 이는 하드웨어적이나 소프트웨어적인 비용상승과 효율 저하를 의미하므로 본 논문에서 제안한 공격방법에 대한 적절한 대응방안에 대한 연구가 필요하다고 볼 수 있다.

## 참고문헌

[1] P. Kocher, J. Jaffe, and B. Jun, “Introduction to Differential Power Analysis and Related Attacks,” Technical Report, Cryptography Research Inc., 1998.

[2] T.S. Messerges, “Power analysis Attacks and Countermeasures for Cryptographic Algorithms,” Ph.D. Thesis, University of Illinois, pp. 541-548, Jan. 2000.

[3] M.L. Akkar, R. Bevan, and L. Goubin, “Two Power Analysis Attacks against One-Mask Methods,” Fast Software Encryption Workshop 2004, LNCS 3017, pp. 332-347, 2004.

[4] H. Handschuh and B. Preneel, “Blind Differential Cryptanalysis for Enhanced Power Attacks,” Workshop on Selected Areas in Cryptography 2006, LNCS 4356, pp. 163-173, 2007.

[5] National Institute of Standards and Technology (NIST), “Data Encryption Standard,” FIPS Publication 46-3, pp. 8-21, 1999.

[6] National Institute of Standards and Technology (NIST), “Advanced Encryption Standard,” FIPS Publication 197, pp. 7-25, 1999.

[7] E. Biham and A. Shamir, “Differential Cryptanalysis of DES-like Cryptosystems,” Journal of Cryptology, vol. 4, no. 1, pp.3-72, Jan. 1991.

[8] L.R. Knudsen, “Truncated and Higher Order Differential,” Fast Software Encryption Workshop 1994, LNCS 1008, pp. 229-236, 1995.
[9] K. Schramm and C. Paar, “Higher Order Masking of the AES,” RSA Conference 2006, Cryptographers' Track, LNCS 3860, pp. 208 - 225, 2006.

## <著 者 紹 介>

강 태 선 (Taesun Kang) 학생회원
1996년 2월: 한국항공대학교 항공기계공학과 학사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 부채널 공격, 공개키 암호, 암호칩 설계 기술

김 희 석 (HeeSeok Kim) 학생회원
2006년 2월: 연세대학교 수학과 졸업(학사)
2008년 2월: 고려대학교 정보경영공학전문대학원 공학석사
2008년 3월~현재: 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 부채널 공격, 공개키 암호시스템 안전성 분석 및 고속구현, 타원곡선 알고리즘

김 태 현 (Tae Hyun Kim) 학생회원
2002년 2월: 서울 시립대학교 수학과 이학사
2004년 8월: 고려대학교 정보보호 대학원 공학석사
2005년 2월~현재: 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 부채널 공격, 공개키 암호 알고리즘, 암호칩 설계 기술

김 종 성 (Jongsung Kim) 정회원
2000년 8월: 고려대학교 수학과 학사
2002년 8월: 고려대학교 수학과 석사
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 박사
2007년 2월: 고려대학교 정보보호대학원 박사
2007년 3월~2008년 3월: 고려대학교 정보보호기술연구센터 연구전임강사
2008년 4월~현재: 고려대학교 정보보호기술연구센터 연구교수
<관심분야> 암호 알고리즘 설계 및 분석, 부채널 공격, 유비쿼터스 시스템

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 2월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2004년 4월~2005년 2월: K.U.Leuven 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보경영공학전문대학원 조교수
2008년 9월~현재: 고려대학교 정보경영공학전문대학원 부교수
<관심분야> 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식