情報保護學會論文誌
第19卷 第1號, 2009. 2

CLEFIA와 ARIA 블록 암호에 대한 다중불능차분공격*

최 준 근,[1†] 김 종 성,[1‡] 성 재 철,[2] 홍 석 희[1]

[1]고려대학교 정보경영공학전문대학원, [2]서울시립대학교 수학과

Multiple Impossible Differential Cryptanalysis of Block Cipher CLEFIA and ARIA*

Joongeun Choi,[1†] Jongsung Kim,[1‡] Jaechul Sung,[2] Seokhie Hong[1]

[1]Center for Information Security Technologies(CIST), Korea University, [2]Department of Mathematics, University of Seoul

요 약

CLEFIA는 SONY사에서 제안한 128-비트 블록 암호이다. 그리고 ARIA는 국내 표준으로 선정된 128-비트 블록 암호이다. 본 논문에서는 다중 불능 차분 공격을 소개하고, [7]에서 제시한 9-라운드 불능 차분을 이용하여 다중 불능 차분 공격을 CLEFIA에 적용한다. 또한 [11]에서 제시한 4-라운드 불능 차분을 이용하여 다중 불능 차분 공격을 ARIA에 적용한다. 본 논문의 CLEFIA 및 ARIA에 대한 다중 불능 차분 공격은 지금까지 제안된 불능 차분 공격보다 더 좋은 결과를 보여준다.

ABSTRACT

CLEFIA is a 128-bit block cipher which is proposed by SONY corporation and ARIA is a 128-bit block cipher which is selected as a standard cryptographic primitive. In this paper, we introduce new multiple impossible differential cryptanalysis and apply it to CLEFIA using 9-round impossible differentials proposed in [7], and apply it to ARIA using 4-round impossible differentials proposed in [11]. Our cryptanalytic results on CLEFIA and ARIA are better than previous impossible differential attacks.

Keywords : Block Cipher, CLEFIA, ARIA, Impossible Differential Cryptanalysis

Ⅰ. 서 론

CLEFIA[1,2]는 음악이나 영상 등의 데이터 유통이 가속하는 환경하에서, 저작권 보호 및 사용자 인증을 위해 SONY사에서 개발한 128-비트 블록 암호이다. 최근 CLEFIA 블록 암호에 대한 연구가 진행되면서 여러 가지 분석 결과가 제시되고 있다. SONY사의 자체 평가에서는 선형 공격, 차분 공격, 불능 차분 공격, 연관키 공격 등에 의한 안전성 분석 결과를 제시하였으며[1,3], [4]에서는 차분 오류 공격을 이용하여 18개의 오류 암호문으로 128-비트의 비밀키를 복구할 수 있고, 54개의 오류 암호문으로 192-비트 및 256-비트의 비밀키를 복구할 수 있음을 보였다. [5]에서는 새로운 접근으로 [3]에서 제시한 불능 차분 공격의 결과보다 더 효율적인 결과를 제시하였다. 그리고 [6]에서는 9-라운드 불능 차분을 제시하고, 이를 이용하여 불능 차분 공격을 12-라운드

접수일(2008년 7월 7일), 게재확정일(2008년 10월 31일)
* 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임. (No. R01-2008-000-11879-0)
† 주저자, joongeun@cist.korea.ac.kr
‡ 교신저자, joshep@cist.korea.ac.kr

[표 1] 분석 결과 비교

구 분		라운드 수	키 길이	선택 평문	시간 복잡도
CLEFIA	[1,3]	10	128, 192, 256	$2^{101.7}$	2^{102}
	[1,3]	11	192, 256	$2^{103.5}$	2^{188}
	[5]	11	128, 192, 256	$2^{103.1}$	$2^{98.1}$
	[1,3]	12	256	$2^{103.8}$	2^{252}
	[5]	12	128, 192, 256	$2^{119.3}$	$2^{114.3}$
	[6]	12	128, 192, 256	$2^{118.9}$	2^{119}
	[7]	12	128, 192, 256	$2^{110.93}$	2^{111}
	본 논문	12	128, 192, 256	$\mathbf{2^{109.6}}$	$\mathbf{2^{109.6}}$
	[5]	13	192, 256	2^{120}	2^{181}
	[6]	13	192, 256	$2^{119.8}$	2^{147}
	[7]	13	192, 256	$2^{111.72}$	$\leq 2^{158}$
	본 논문	13	192, 256	$\mathbf{2^{110.3}}$	$\mathbf{\leq 2^{127.9}}$
	[5]	14	256	$2^{120.4}$	$2^{245.4}$
	[6]	14	256	$2^{120.3}$	2^{211}
	[7]	14	256	$2^{112.3}$	$\leq 2^{199}$
	본 논문	14	256	$\mathbf{2^{110.8}}$	$\mathbf{\leq 2^{176.4}}$
	[7]	15	256	2^{113}	$\leq 2^{248}$
	본 논문	15	256	$\mathbf{2^{111.1}}$	$\mathbf{\leq 2^{216.7}}$
ARIA	[10]	6	128, 192, 256	2^{121}	2^{112}
	[11]	6	128, 192, 256	2^{113}	$2^{121.6}$
	본 논문	6	128, 192, 256	$\mathbf{2^{110.4}}$	$\mathbf{\leq 2^{111.5}}$

CLEFIA-128과 14-라운드 CLEFIA-192, CLEFIA-256에 적용하였다.

현재까지 CLEFIA 블록 암호에 대한 가장 효과적인 분석 결과는 9-라운드 불능 차분을 이용한 12-라운드 CLEFIA-128, 13-라운드 CLEFIA-192, 그리고 15-라운드 CLEFIA-256에 대한 불능 차분 공격이다[7]. CLEFIA의 구조와 관련된 9-라운드 불능 차분[3], 행렬의 branch number와 관련된 다른 9-라운드 불능 차분[6]과 다르게 [7]에서는 CLEFIA의 선형 행렬 M_0, M_1의 특징을 분석하여 향상된 9-라운드 불능 차분을 제시하였다. 본 논문에서는 새로운 다중 불능 차분 공격을 소개하고, [7]에서 제시한 9-라운드 불능 차분을 다중 불능 차분 공격에 적용하여 CLEFIA를 분석한다.

한편, 국내 표준으로 선정된 ARIA[8]블록 암호에 대한 연구도 최근 활발하게 진행되고 있다. 자체 평가에서는 차분 공격, 선형 공격, 부정 차분 공격, 고계 차분 공격 등에 의한 안전성 분석 결과를 제시하였으며[8], [9]에서는 부정차분공격 및 선형 공격에 대한 분석 결과를 제시하였다. 이후 Wu 등에 의해서 처음으로 4-라운드 불능 차분이 제시되었고[10], 이를 이용하여 6-라운드 불능 차분 공격법을 적용하였다. 그리고 [11]에서는 [10]에서 제안된 4-라운드 불능 차분을 일반화시킨 후, 2^{113}의 선택 평문과 $2^{121.5}$의 시간 복잡도로 6-라운드 ARIA의 비밀키를 복구하였다. 또한 서정갑 등이 ARIA에 대한 차분전력분석공격[12]을 소개하였다. 본 논문에서는 [11]에서 제시한 4-라운드 불능 차분을 다중 불능 차분 공격에 적용하여 ARIA를 분석한다.

본 공격은 동일한 입력 불능 차분에 대한 여러 개의 출력 불능 차분을 순차적으로 고려하여 올바르지 않은 키를 다중으로 걸러내기 때문에 필요한 평문 쌍과 시간 복잡도를 줄일 수 있다. 본 논문의 분석 결과와 이전의 불능 차분 공격 분석 결과에 대한 비교는 [표 1]과 같다.

본 논문의 구성은 다음과 같다. 2장에서는 CLEFIA와 ARIA 블록 암호 알고리즘을 묘사하고, 3장에서 다중 불능 차분 공격 방법을 소개한다. 4장에서는 다중 불능 차분 공격법을 CLEFIA에 적용하고, 5장에서 다중 불능 차분 공격법을 ARIA에 적용한다. 마지막으로 6장은 본 논문의 결론이다.

Ⅱ. CLEFIA 블록 암호

본 장에서는 본 논문에 사용되는 표기법을 정리하고, CLEFIA와 ARIA 블록 암호를 소개한다.

2.1 표기법

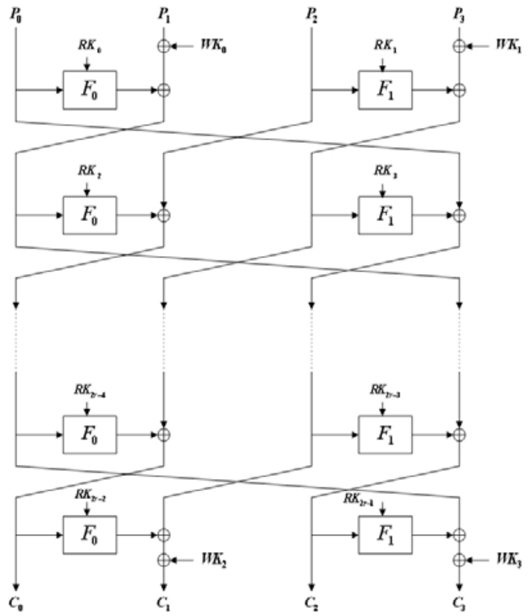
- $a \oplus b$: a와 b의 비트별 배타적 논리합
- $a \| b$: a와 b의 연접
- a^T : 벡터 a의 전치 벡터
- P (C) : 128-비트 평문 (암호문)
- Δa : 차분
- $a_{(n)}$: n-비트 데이터 a
- $[a_i^0, a_i^1, a_i^2, a_i^3]$: i번째 라운드의 출력값 $(a_i^j \in \{0,1\}^{32})$
- a_i^I (a_i^O) : i번째 라운드의 입력값 (출력값)
- a_i^S (a_i^D) : i번째 라운드의 S-box Layer 이후의 값 (Diffusion Layer 이후의 값)

· $a_{i,j}^{*}$: a_i^{*}의 j번째 바이트 ($* \in \{I, O, S, D\}$)
· $RK(WK)$: 라운드 키(화이트닝 키)

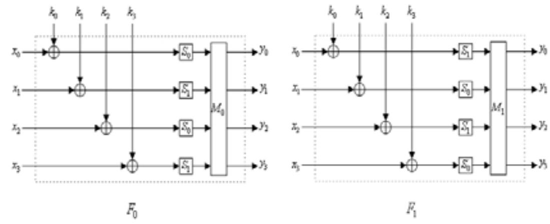
2.2 CLEFIA 블록 암호 소개

CLEFIA는 128-비트 블록 크기를 가지며, 128-비트, 192-비트, 그리고 256-비트의 비밀키 크기를 가진다. 그리고 키 길이에 따라 각각 18-라운드, 22-라운드, 26-라운드를 적용한다. CLEFIA는 128-비트의 평문 $P=(P_0, P_1, P_2, P_3)$과 n-비트 키($n=128,\ 192,\ 256$)를 입력받아 128-비트의 암호문을 출력한다. CLEFIA의 암호화 과정은 다음과 같다 ([그림 1] 참조).

단계 1. $x_0^0 = P_0,\ x_0^1 = P_1 \oplus WK_0,$
$x_0^2 = P_2,\ x_0^3 = P_3 \oplus WK_1,$
단계 2. for $i=1$ to $r-1$,
$x_i^0 = x_{i-1}^1 \oplus F_0(x_{i-1}^0, RK_{2i-2}),\ x_i^1 = x_{i-1}^2,$
$x_i^2 = x_{i-1}^3 \oplus F_1(x_{i-1}^2, RK_{2i-1}),\ x_i^3 = x_{i-1}^0;$
단계 3. $C_0 = x_{r-1}^0,$
$C_1 = F_0(x_{r-1}^0, RK_{2r-2}) \oplus x_{r-1}^1 \oplus WK_2,$
$C_2 = x_{r-1}^2,$
$C_3 = F_1(x_{r-1}^2, RK_{2r-1}) \oplus x_{r-1}^3 \oplus WK_3$



[그림 1] r-라운드 CLEFIA 전체 구조



[그림 2] F 함수 구조

F_0 함수와 F_1 함수는 32-비트 데이터 값과 키를 입력받아 32-비트 값을 출력한다 ([그림 2] 참조). F_0 함수의 데이터 처리 과정은 다음과 같다.

단계 1. $x = x_{0(8)} | x_{1(8)} | x_{2(8)} | x_{3(8)}$
단계 2. $S(x) = S_0(x_{0(8)} \oplus RK_{0(8)}) |$
$S_1(x_{1(8)} \oplus RK_{1(8)}) | S_0(x_{2(8)} \oplus RK_{2(8)}) |$
$S_1(x_{3(8)} \oplus RK_{3(8)}) = z_{0(8)} | z_{1(8)} | z_{2(8)} | z_{3(8)}$
단계 3. $y = M_0(z_{0(8)}, z_{1(8)}, z_{2(8)}, z_{3(8)})^T$

F_1 함수의 데이터 처리 과정은 다음과 같다.

단계 1. $x = x_{0(8)} | x_{1(8)} | x_{2(8)} | x_{3(8)}$
단계 2. $S(x) = S_1(x_{0(8)} \oplus RK_{0(8)}) |$
$S_0(x_{1(8)} \oplus RK_{1(8)}) | S_1(x_{2(8)} \oplus RK_{2(8)}) |$
$S_0(x_{3(8)} \oplus RK_{3(8)}) = z_{0(8)} | z_{1(8)} | z_{2(8)} | z_{3(8)}$
단계 3. $y = M_1(z_{0(8)}, z_{1(8)}, z_{2(8)}, z_{3(8)})^T$

S_0, S_1은 비선형 8-비트 S-Box이고, 두 행렬 M_0과 M_1은 다음과 같이 정의한다.

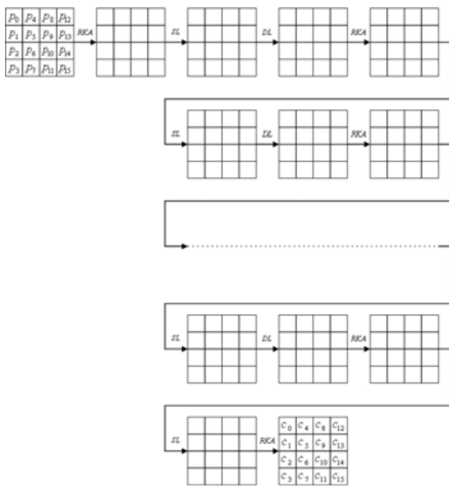
$$M_0 = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix},$$

$$M_1 = \begin{pmatrix} 0x01 & 0x08 & 0x02 & 0x0a \\ 0x08 & 0x01 & 0x0a & 0x02 \\ 0x02 & 0x0a & 0x01 & 0x08 \\ 0x0a & 0x02 & 0x08 & 0x01 \end{pmatrix}.$$

행렬과 벡터 연산은 $GF(2^8)$에서 수행되며, 기약다항식은 $x^8 + x^4 + x^3 + x^2 + 1$이다.

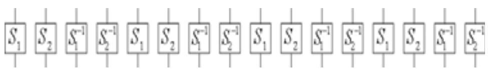
2.3 ARIA 블록 암호 소개

ARIA는 128-비트 블록 크기를 가지며, 128-비트, 192-비트, 그리고 256-비트의 비밀키 크기를 가진다. 그리고 키 길이에 따라 각각 12-라운드, 14-라운드, 16-라운드를 적용한다. ARIA는 SPN 구조를 설계 논리로 적용하였으며, 128-비트의 평문 $P=(p_0,\ldots,p_{15})$과 n-비트 키 $(n=128,\ 192,\ 256)$를 입력받아 128-비트의 암호문을 출력한다. ARIA의 라운드 함수는 Round Key Addition (RKA), Substitution Layer (SL), Diffusion Layer (DL) 의 3가지 기본 연산을 차례로 수행한다. 첫 라운드 전에 초기 RKA 연산이 먼저 수행되며, 마지막 라운드에는 DL 연산을 제외한다 ([그림 3] 참조).

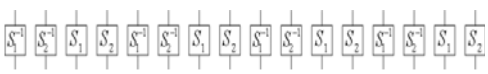


[그림 3] ARIA 전체 구조

1. RKA : 128-비트 라운드 키를 XOR 연산한다.
2. SL : 두 개의 8×8 S-box를 이용하여 연산한다. 홀수 라운드에서는 첫 번째 타입을 적용하고, 짝수 라운드에서는 두 번째 타입을 적용한다.



SL 타입 1



SL 타입 2

3. DL : $A^2=I$ 가 되는 16×16 행렬 A를 이용하여 연산한다. 행렬 A는 $A:(\{0,1\}^8)^{16}\rightarrow(\{0,1\}^8)^{16}$로 정의하며, 다음과 같다.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0\ 00\ 11\ 01\ 01\ 10\ 00\ 11\ 0 \\ 0\ 01\ 00\ 10\ 11\ 10\ 01\ 00\ 1 \\ 0\ 10\ 01\ 01\ 00\ 01\ 11\ 00\ 1 \\ 1\ 00\ 00\ 10\ 10\ 01\ 10\ 11\ 0 \\ 1\ 01\ 00\ 10\ 01\ 00\ 10\ 01\ 1 \\ 0\ 10\ 11\ 00\ 00\ 11\ 00\ 01\ 1 \\ 1\ 01\ 00\ 00\ 10\ 11\ 01\ 10\ 0 \\ 0\ 10\ 10\ 01\ 01\ 00\ 11\ 10\ 0 \\ 1\ 10\ 01\ 00\ 10\ 01\ 00\ 10\ 1 \\ 1\ 10\ 00\ 11\ 00\ 00\ 11\ 01\ 0 \\ 0\ 01\ 10\ 11\ 01\ 00\ 00\ 10\ 1 \\ 0\ 01\ 11\ 00\ 10\ 10\ 01\ 01\ 0 \\ 0\ 11\ 00\ 01\ 10\ 10\ 11\ 00\ 0 \\ 1\ 00\ 10\ 01\ 11\ 01\ 00\ 10\ 0 \\ 1\ 00\ 11\ 10\ 00\ 10\ 10\ 01\ 0 \\ 0\ 11\ 01\ 10\ 01\ 01\ 00\ 00\ 1 \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix}$$

$(x_0,x_1,\ldots,x_{15}) \mapsto (y_0,y_1,\ldots,y_{15})$일 때, 각 8-비트 출력값은 다음과 같다.

$y_0 = x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14}$,

$y_1 = x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15}$,

$y_2 = x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15}$,

$y_3 = x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14}$,

$y_4 = x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15}$,

$y_5 = x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15}$,

$y_6 = x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13}$,

$y_7 = x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13}$,

$y_8 = x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15}$,

$y_9 = x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14}$,

$y_{10} = x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15}$,

$y_{11} = x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14}$,

$y_{12} = x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12}$,

$y_{13} = x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13}$,

$y_{14} = x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14}$,

$y_{15} = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}$.

Ⅲ. 다중 불능 차분 공격

불능 차분 공격은 E. Biham, A. Biryukov, A. Shamir 등에 의해 1999년 처음 소개되었다[13]. 이후

XTEA와 TEA의 축소된 라운드[14], 일반적인 블록 암호 구조[15], 30-라운드 SHACAL-2[16], 6-라운드 AES[17] 등에 불능차분공격이 적용되었다. 이 공격법은 선택 평문 공격으로 31-라운드 SKIPJACK을 분석하는데 사용되었다. 불능 차분 공격은 전혀 일어날 수 없는 차분 특성을 이용하여 올바르지 않은 키 후보들을 걸러낸 후 올바른 키를 찾아낸다.

일반적인 r-라운드 불능 차분 공격은 다음과 같다.

1. 처음 r'-라운드에 대한 불능 차분 특성을 찾는다 ($r' < r$).
2. 찾아낸 불능 차분 특성의 입력 차분을 만족하는 선택 평문 쌍에 대한 암호문 쌍을 얻는다. 이 때, 불능 차분 특성의 출력 차분으로부터 마지막 k-라운드 후에 나올 수 있는 출력 차분 성질을 만족하는 암호문 쌍만 남긴다 ($k = r - r'$): 암호문 쌍 필터링 과정.
3. 마지막 k-라운드의 키를 추측하여 남은 암호문 쌍을 복호화한 후 불능 차분 특성의 출력 차분을 만족하는지 검사한다.
4. 출력 차분을 만족하는 라운드 키는 버리고, 이 과정을 반복하여 최종적으로 남은 키를 올바른 키로 택한다.

본 논문에서 제안하는 다중 불능 차분 공격은 동일한 입력 불능 차분에 대한 여러 종류의 출력 불능 차분을 찾고, 이를 순차적으로 고려하여 올바르지 않은 키를 다중으로 걸러낸다.

r-라운드 다중 불능 차분 공격은 다음과 같다.

1. 동일한 입력 불능 차분에 대한 여러 종류의 출력 불능 차분을 갖는 처음 r'-라운드 다중 불능 차분 특성을 찾는다 ($r' < r$). 다른 종류의 출력 불능 차분 형태는 암호문 복호화시 관여하는 키 비트가 겹치는 부분과 그렇지 않은 부분이 있다고 가정한다. 출력 불능 차분의 형태를 $A_1, A_2, \ldots, A_s$라 하자.
2. 찾아낸 다중 불능 차분 특성의 입력 차분을 만족하는 선택 평문 쌍에 대한 암호문 쌍을 얻는다. 이 때, 다중 불능 차분 특성의 출력 차분 $A_1, A_2, \ldots, A_s$ 각각에 대하여 마지막 k-라운드 후에 나올 수 있는 출력 차분 성질을 만족하는 암호문 쌍을 남긴다 ($k = r - r'$) : 암호문 쌍 필터링 과정.
3. $i = 1$부터 $i = s$까지 다음을 시행한다. A_i에 관여하는 키를 추측한 후 남아 있는 관계된 암호문 쌍을 복구하여 A_i 형태가 나오면 추측한 키를 버린다. 이 과정을 수행할 때, A_j의 관여된 키는 전 단계 A_{j-1}의 관여된 키 비트와 겹치는 부분은 제외하고 추측한다 ($2 \le j \le s$). 겹치는 키 비트는 A_{j-1}에서 남은 키에 대해서만 검사한다. A_s에 관여한 키가 한 개가 남을 때까지 이 과정을 수행한다.
4. 최종적으로 남은 키를 옳은 키로 택한다.

위 공격을 이용하면 불능 차분 공격보다 필요한 평문 쌍과 시간 복잡도를 줄일 수 있다.

Ⅳ. CLEFIA에 대한 다중 불능 차분 공격

[7]에서 사용한 불능 차분은 [표 3]의 $[0,00\gamma\beta,0,0] \rightarrow_{9R} [0,00\alpha0,0,0]$이다. 본 논문에서는 다중 불능 차분 공격을 적용하기 위해 [표 2]의 불능 차분 $[0,000\alpha,0,0] \rightarrow_{9R} [(0,00\beta\gamma,0,0),(0,0\beta0\gamma,0,0),(0,\beta00\gamma,0,0)]$을 사용한다. 다른 형태의 불능차분은 표 2의 불능 차분이 갖는 특성보다 그 확률이 같거나 좋지 않으므로 고려하지 않는다.

4.1 다중 불능 차분 공격 적용 방법

식 (1)는 [7]에서의 키 복구 계산 과정이다.

[표 2] 9-라운드 불능 차분 (1) [7]
(α, β, γ : 0이 아닌 임의의 8-비트 차분)

입력 불능 차분	출력 불능 차분
$[0,000\alpha,0,0]$	$[0,00\beta\gamma,0,0],[0,0\beta0\gamma,0,0],[0,\beta00\gamma,0,0]$
$[0,00\alpha0,0,0]$	$[0,0\beta\gamma0,0,0],[0,\beta0\gamma0,0,0],[0,00\beta\gamma,0,0]$
$[0,0\alpha00,0,0]$	$[0,\beta\gamma00,0,0],[0,0\gamma0\beta,0,0],[0,0\beta\gamma0,0,0]$
$[0,\alpha000,0,0]$	$[0,\gamma00\beta,0,0],[0,\gamma0\beta0,0,0],[0,\beta\gamma00,0,0]$
$[0,0,0,000\alpha]$	$[0,0,0,00\beta\gamma],[0,0,0,0\beta0\gamma],[0,0,0,\beta00\gamma]$
$[0,0,0,00\alpha0]$	$[0,0,0,0\beta\gamma0],[0,0,0,\beta0\gamma0],[0,0,0,00\gamma\beta]$
$[0,0,0,0\alpha00]$	$[0,0,0,\beta\gamma00],[0,0,0,0\gamma0\beta],[0,0,0,0\gamma\beta0]$
$[0,0,0,\alpha000]$	$[0,0,0,\gamma00\beta],[0,0,0,\gamma0\beta0],[0,0,0,\gamma\beta00]$

[표 3] 9-라운드 불능 차분 (2) [7]
(α, β, γ : 0이 아닌 임의의 8-비트 차분)

입력 불능 차분	출력 불능 차분
$[0,00\beta\gamma,0,0]$, $[0,0\beta0\gamma,0,0]$, $[0,\beta00\gamma,0,0]$	$[0,000\alpha,0,0]$
$[0,0\beta\gamma0,0,0]$, $[0,\beta0\gamma0,0,0]$, $\boldsymbol{[0,00\gamma\beta,0,0]}$	$\boldsymbol{[0,00\alpha0,0,0]}$
$[0,\beta\gamma00,0,0]$, $[0,0\gamma0\beta,0,0]$, $[0,0\gamma\beta0,0,0]$	$[0,0\alpha00,0,0]$
$[0,\gamma00\beta,0,0]$, $[0,\gamma0\beta0,0,0]$, $[0,\gamma\beta00,0,0]$	$[0,\alpha000,0,0]$
$[0,0,0,00\beta\gamma]$, $[0,0,0,0\beta0\gamma]$, $[0,0,0,\beta00\gamma]$	$[0,0,0,000\alpha]$
$[0,0,0,0\beta\gamma0]$, $[0,0,0,\beta0\gamma0]$, $[0,0,0,00\gamma\beta]$	$[0,0,0,00\alpha0]$
$[0,0,0,\beta\gamma00]$, $[0,0,0,0\gamma0\beta]$, $[0,0,0,0\gamma\beta0]$	$[0,0,0,0\alpha00]$
$[0,0,0,\gamma00\beta]$, $[0,0,0,\gamma0\beta0]$, $[0,0,0,\gamma\beta00]$	$[0,0,0,\alpha000]$

$$2^L(1-p)^N < 1. \qquad (1)$$

L은 추측하는 키와 계산하는 키의 총 비트 수이고, p는 입력 차분에 대한 F 함수의 총 확률이다. N은 필터링을 거친 후 남아있는 암호문 쌍이다. 식 (1)은 N개의 암호문 쌍을 불능 차분 특성에 적용하여 2^L-1개의 틀린 키를 걸러내고, 올바른 키만 남아있음을 의미한다. 표 2의 9-라운드 다중 불능 차분 특성을 보면 동일한 입력 차분에 대해 3가지 종류의 출력 불능 차분이 나타남을 알 수 있다. 그리고 출력 차분이 한 워드(32-비트)에서만 바이트의 자리가 다르며, 각 출력 차분에 대해 관여하는 키 부분은 8-비트를 제외하고 모두 동일하다. 따라서 식 (1)을 다음의 세 단계로 바꿀 수 있다. p_i와 N_i는 i번째 출력 불능 차분 형태에 대한 F 함수의 총 확률 및 필터링 후 남아 있는 암호문 쌍을 의미한다.

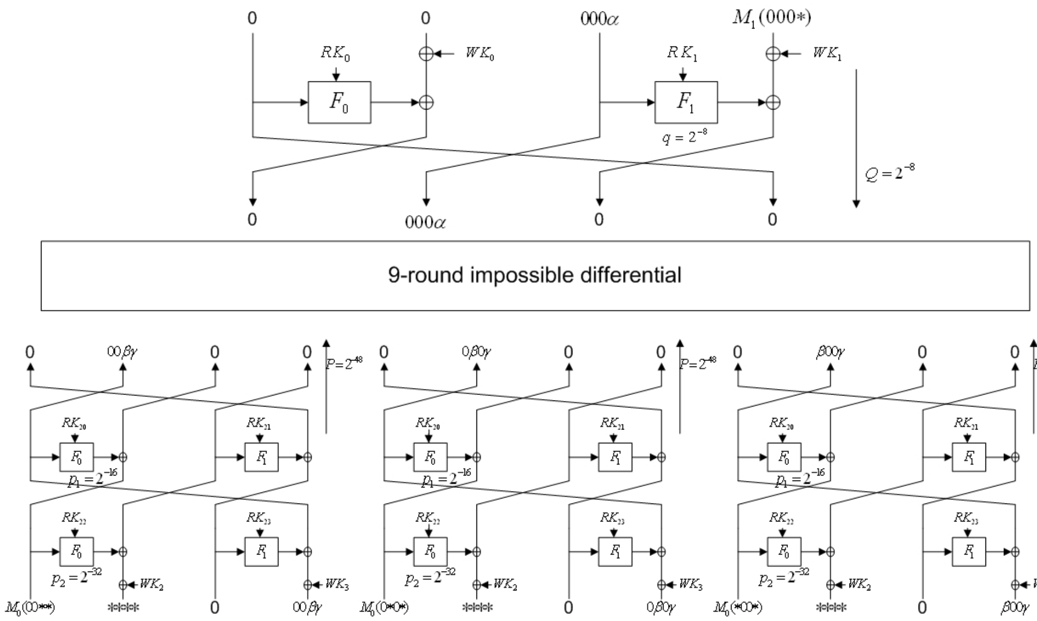
$$\begin{aligned} &2^L(1-p_1)^{N_1} \approx 2^{x_1} \\ &2^{8+x_1}(1-p_2)^{N_2} \approx 2^{x_2} \\ &2^{8+x_2}(1-p_3)^{N_3} < 1 \end{aligned} \qquad (2)$$

x_1은 첫 번째 출력 불능 차분에 대해 남은 키의 비트 수이고, x_2는 두 번째 출력 불능 차분에 대해 남은 키의 비트 수이다. 마지막으로 세 번째 출력 불능 차분에 대해 하나의 옳은 키가 남도록 필요한 평문 쌍의 개수를 계산한다 (기존의 단일 불능 차분 특성을 이용하는 것보다 적은 데이터량 요구).

4.2 12-라운드 CLEFIA에 대한 다중 불능 차분 공격

CLEFIA 및 ARIA에 대한 다중 불능 차분 공격에 다음의 정리를 사용한다.

[정리 1] F 함수 (F_0, F_1)에 대해, 두 입력값 (In, In')과 그에 대한 출력 차분 Δ_{out}을 알면, F 함수의 라운드



[그림 4] 12-라운드 CLEFIA 다중 불능 차분 공격

키를 계산할 수 있다[5].

12-라운드 공격은 [그림 4]와 같이 9-라운드 다중 불능 차분 특성 위에 한 라운드를 추가하고, 아래에 두 라운드를 추가하여, 추가된 라운드의 관여된 키 비트를 축출한다. 공격 방법은 다음과 같다.

1. 차분이 $\Lambda = (0,0,000\alpha, M_1(000*))$꼴이 되는 128-비트 평문을 원소로 갖는 structure를 구성한다.
   (α, $*$: 0이 아닌 임의의 8-비트 차분, structure = $\{C \oplus \lambda \mid C$: 상수, $\lambda \in \Lambda\}$)
   하나의 structure에서 원소의 개수는 $(2^8-1)^2 \approx 2^{16}$개이고, 만들 수 있는 쌍은 $\binom{2^{16}}{2} \approx 2^{31}$개다.
2. $2^{93.6}$개의 structure를 선택한다. 입력 차분 $(0,000\alpha,0,0)$에 대한 9-라운드 불능 차분의 첫 번째 출력 차분 $(0,00\beta\gamma,0,0)$에 대해 $(M_0(0*0*),****,0,0*0*)$의 차분 형태가 되는 암호문 쌍만 남긴다([그림 4] 참조). 따라서 필터링 후 남는 암호문 쌍의 개수는 $N_1 = 2^{124.6} \times 2^{-64} = 2^{60.6}$개가 된다. 두 번째 출력 차분 $(0,0\beta 0\gamma,0,0)$과 세 번째 출력 차분 $(0,\beta 00\gamma,0,0)$을 살펴보면, 첫 번째 출력 차분과 같은 워드에서만 차분의 바이트 자리가 다르다. 그러므로 두 번째 출력 차분 $(0,0\beta 0\gamma,0,0)$에 대해 $(M_0(0*0*),****,0,0*0*)$의 차분 형태가 되는 암호문 쌍만 남기면 $N_2 = N_1 = 2^{60.6}$개의 암호문 쌍이 남으며, 세 번째 출력 차분에 대해서도 $N_3 = N_2 = N_1 = 2^{60.6}$으로 동일하다.
3. 남아 있는 N_1개의 암호문 쌍에 대해 라운드 키 RK_{23} (32-비트)를 추측하면 정리 1에 의해 RK_{22}, $(\boldsymbol{WK_3 \oplus RK_{20}})_{\mathbf{2,3}}$, $(RK_1)_3$ (56-비트)의 키를 계산할 수 있다. 마찬가지로 N_2 및 N_3에 대해 라운드 키 RK_{23} (32-비트)를 추측하면 정리 1에 의해 각각 RK_{22}, $(\boldsymbol{WK_3 \oplus RK_{20}})_{\mathbf{1,3}}$, $(RK_1)_3$ (56-비트) 및 RK_{22}, $(\boldsymbol{WK_3 \oplus RK_{20}})_{\mathbf{0,3}}$, $(RK_1)_3$ (56-비트)를 계산할 수 있다. 각각의 출력 차분에 대해 계산하는 키가 8-비트씩만 다름을 알 수 있다 ($(WK_3 \oplus RK_{20})_{2,3}$, $(WK_3 \oplus RK_{20})_{1,3}$, $(WK_3 \oplus RK_{20})_{0,3}$). 식 (2)를 적용하면
   $2^{88}(1-2^{-56})^{N_1} \approx 2^{53.1}$
   $2^{8+53.1}(1-2^{-56})^{N_2} \approx 2^{26.2}$
   $2^{8+26.2}(1-2^{-56})^{N_3} < 1$
   이 된다 (확률 $2^{-56} = P \cdot Q$: [그림 4] 참조). 따라서 위 계산에 의해 남은 키가 옳은 키가 된다.

첫 번째 불능 차분 $(0,00\beta\gamma,0,0)$에 대한 F 함수의 총 확률은 다음과 같이 계산한다. 평문의 차분 $(0,0,000\alpha,M_1(000*))$에 대해 1-라운드의 F_1 함수의 입력 차분 000α에 대한 확률 q는 2^{-8}이 되고, 11-라운드의 F_0 함수의 입력 차분 $00\beta\gamma$에 대한 확률 p_1는 2^{-16}이 된다. 그리고, 12-라운드의 F_0 함수의 입력 차분 $M_0(00**)$에 대한 확률 p_2는 2^{-32}이 되므로, 첫 번째 불능 차분에 대한 F 함수의 총 확률은 $p_1 \cdot p_2 \cdot q = 2^{-48} \cdot 2^{-8} = 2^{-56}$이 된다 ($P = p_1 p_2$, $Q = q$). 1-라운드의 F_0 함수의 확률, 11-라운드 및 12-라운드의 F_1 함수의 확률은 입력 차분이 0이므로 모두 1이 된다. 두 번째와 세 번째의 불능 차분 형태에 대한 F 함수의 총 확률 $P \cdot Q$도 2^{-56}으로 모두 동일하게 계산된다.

시간 복잡도는 평문 $2^{109.6}$개의 암호화 과정과, 32-비트의 키를 추측하고 필터링 후 남은 암호문 쌍 N_1, N_2, N_3에 대한 F 함수 계산 과정 $\le 2^{32} \cdot 2^{60.6} \cdot 3 = 2^{93.9}$을 더하여 $2^{109.6}$이 된다.

4.3 13-라운드 CLEFIA에 대한 다중 불능 차분 공격

13-라운드 공격은 9-라운드 다중 불능 차분 특성 위에 두 라운드를 추가하고, 아래에 두 라운드를 추가하여, 추가된 라운드의 관여된 키 비트를 축출한다. 공격 방법은 다음과 같다.

1. 차분이 $\Lambda = (M_1(000*),****,0,000\alpha)$꼴이 되는 128-비트 평문을 원소로 갖는 structure를 구성한다.
   (하나의 structure에서 원소의 개수는 $(2^8-1)^6 \approx 2^{48}$개이고, 만들 수 있는 쌍은 $\binom{2^{48}}{2} \approx 2^{95}$개다)
2. $2^{62.3}$개의 structure를 선택한다. 입력 차분 $(0,000\alpha,0,0)$에 대한 9-라운드 후 첫 번째 출력 차분 $(0,00\beta\gamma,0,0)$에 대해 $(M_0(00**),****,0,00**)$의 차분 형태가 되는 암호문 쌍만 남긴다. 따라서 남아 있는 암호문 쌍의 개수는 $N_1 = 2^{157.3} \times 2^{-64} = 2^{93.3}$개가 된다. N_1과 N_2도 $2^{93.3}$이 된다.
3. 첫 번째 출력 불능 차분에 대해 남아 있는 N_1개의

암호문 쌍에 대해 라운드 키 RK_{25}, RK_1 (64-비트)를 추측하면 정리 1에 의해 RK_0, $(WK_1 \oplus RK_3)_3$, RK_{24}, $(\boldsymbol{WK_3 \oplus RK_{22}})_{2,3}$ (88-비트)의 키를 계산할 수 있다. 마찬가지로 N_2 및 N_3에 대해 라운드 키 RK_{25}, RK_1 (64-비트)를 추측하면 정리 1에 의해 각각 RK_0, $(WK_1 \oplus RK_3)_3$, RK_{24}, $(\boldsymbol{WK_3 \oplus RK_{22}})_{1,3}$ (88-비트) 및 RK_0, $(WK_1 \oplus RK_3)_3$, RK_{24}, $(\boldsymbol{WK_3 \oplus RK_{22}})_{0,3}$ (88-비트)를 계산할 수 있다. 축출하는 키가 각각 8-비트씩 다르므로, 식 (2)를 적용하여

$2^{152}(1-2^{-88})^{N_1} \approx 2^{95.2}$

$2^{8+95.2}(1-2^{-88})^{N_2} \approx 2^{46.4}$

$2^{8+46.4}(1-2^{-88})^{N_3} < 1$

을 얻는다 (앞 공격과 같이 확률 2^{-88}은 계산하는 키 비트에 관계된 확률의 합을 나타낸다). 따라서 위 계산에 의해 남은 키가 옳은 키가 된다.

시간 복잡도는 평문 $2^{110.3}$개의 암호화 과정과, 9-라운드 불능 차분 특성의 위 두 라운드에서 추측하는 키 32-비트와 9-라운드 불능 차분 특성의 아래 두 라운드에서 추측하는 키 32-비트에 대해 각 2^{32}개의 키를 키 테이블에 저장한 후 선택 평문 쌍 N_1, N_2, N_3으로 키를 계산하여 불능 차분이 나오는 키를 조합하여 버린다. 따라서 이 검사 과정은 $\le \{(2^{32}N_1 + 2^{32}N_1) + (2^{32}N_2 + 2^{32}N_2) + (2^{32}N_3 + 2^{32}N_3)\} = 2^{127.9}$의 암호화 과정을 요구하므로, 전체 시간 복잡도는 $2^{127.9}$이다.

4.4 14-라운드 CLEFIA에 대한 다중 불능 차분 공격

14-라운드 공격은 9-라운드 다중 불능 차분 특성 위에 두 라운드를 추가하고, 아래에 세 라운드를 추가하여, 추가된 라운드의 관여된 키 비트를 축출한다. 공격 방법은 다음과 같다.

1. 4.3절의 13-라운드 공격의 단계 1과 동일하게 수행한다.
2. $2^{62.8}$개의 structure를 선택한다. 입력 차분 $(0,000\alpha,0,0)$에 대한 9-라운드 후 첫 번째 출력 차분 $(0,00\beta\gamma,0,0)$에 대해 $(****,****,00**,M_0(00**) \oplus M_1(00**))$의 차분 형태가 되는 암호문 쌍만 남긴다. 따라서 필요한 선택 평문 쌍의 개수는 $N_1 = 2^{157.8} \times 2^{-16} = 2^{141.8}$개가 된다. $(N_1 = N_2 = N_3 = 2^{141.8})$
3. 남아 있는 N_1개의 암호문 쌍에 대해 라운드 키 RK_1, $(RK_{24} \oplus WK_3)$ (64-비트)를 추측하면 정리 1에 의해 RK_{26}, RK_{27}, $(RK_{25} \oplus WK_2)$, $(\boldsymbol{RK_{22}})_{2,3}$, RK_0, $(RK_3 \oplus WK_1)_3$ (152-비트)의 키를 계산할 수 있다. 마찬가지로 N_2 및 N_3에 대해 라운드 키 RK_1, $(RK_{24} \oplus WK_3)$ (64-비트)를 추측하면 정리 1에 의해 각각 RK_{26}, RK_{27}, $(RK_{25} \oplus WK_2)$, $(\boldsymbol{RK_{22}})_{1,3}$, RK_0, $(RK_3 \oplus WK_1)_3$ (152-비트)및 RK_{26}, RK_{27}, $(RK_{25} \oplus WK_2)$, $(\boldsymbol{RK_{22}})_{0,3}$, RK_0, $(RK_3 \oplus WK_1)_3$ (152-비트)를 계산할 수 있다. 각각의 출력 차분에 대해 계산하는 키가 8-비트씩만 다르다. 식 (2)를 적용하면

   $2^{216}(1-2^{-136})^{N_1} \approx 2^{135.7}$

   $2^{8+135.7}(1-2^{-136})^{N_2} \approx 2^{63.4}$

   $2^{8+63.4}(1-2^{-136})^{N_3} < 1$

   이 된다 (앞 공격과 마찬가지로 확률 2^{-136}은 계산하는 키와 관여된 라운드의 확률의 총합을 나타낸다). 따라서 위 계산에 의해 남은 키가 옳은 키가 된다.

시간 복잡도는 평문 $2^{110.8}$개의 암호화 과정과, 9-라운드 불능 차분 특성의 위 두 라운드에서 추측하는 키 32-비트와 9-라운드 불능 차분 특성의 아래 세 라운드에서 추측하는 키 32-비트에 대한 선택 평문 쌍 N_1, N_2, N_3의 검사 과정 $\le \{(2^{32}N_1 + 2^{32}N_1) + (2^{32}N_2 + 2^{32}N_2) + (2^{32}N_3 + 2^{32}N_3)\} = 2^{176.4}$(키 테이블 이용)을 더하여 $2^{176.4}$이 된다.

4.5 15-라운드 CLEFIA에 대한 다중 불능 차분 공격

15-라운드 공격은 9-라운드 다중 불능 차분 특성 위에 세 라운드를 추가하고, 아래에 세 라운드를 추가하여, 추가된 라운드의 관여된 키 비트를 축출한다. 공격 방법은 다음과 같다.

1. 차분이 $(000\alpha, M_0(000*) \oplus M_1(000*), ****, ****)$꼴이 되는 128-비트 평문을 원소로 갖는 structure를 구성한다.
   (하나의 structure에서 원소의 개수는 $(2^8-1)^{11} \approx 2^{88}$개이고, 만들 수 있는 쌍은 $\binom{2^{88}}{2} \approx 2^{175}$개다)

2. $2^{23.1}$개의 structure를 선택한다. 입력 차분 $(0,000\alpha,0,0)$에 대한 9-라운드 후 첫 번째 출력 차분 $(0,00\beta\gamma,0,0)$에 대해 $(****,****,00**,M_0(00**)\oplus M_1(00**))$의 차분 형태가 되는 암호문 쌍만 남긴다. 따라서 필터링 후 남는 암호문 쌍의 개수는 $N_1 = 2^{198.1}\times 2^{-16} = 2^{182.1}$개가 된다. N_2와 N_3도 N_1과 동일하다 ($N_1 = N_2 = N_3 = 2^{182.1}$).
3. 남아 있는 N_1개의 암호문 쌍에 대해 라운드 키 $(RK_3\oplus WK_1)$, $(RK_{27}\oplus WK_2)$ (64-비트)를 추측하면 정리 1에 의해 RK_{28}, RK_{29}, $(RK_{26}\oplus WK_3)$, $(\boldsymbol{RK_{24}})_{2,3}$, RK_0, RK_1, $(RK_2\oplus WK_0)$, $(RK_5)_3$ (216-비트)의 키를 계산할 수 있다. 마찬가지로 N_2 및 N_3에 대해 라운드 키 $(RK_3\oplus WK_1)$, $(RK_{27}\oplus WK_2)$ (64-비트)를 추측하면 정리 1에 의해 각각 RK_{28}, RK_{29}, $(RK_{26}\oplus WK_3)$, $(\boldsymbol{RK_{24}})_{1,3}$, RK_0, RK_1, $(RK_2\oplus WK_0)$, $(RK_5)_3$ (216-비트) 및 RK_{28}, RK_{29}, $(RK_{26}\oplus WK_3)$, $(\boldsymbol{RK_{24}})_{0,3}$, RK_0, RK_1, $(RK_2\oplus WK_0)$, $(RK_5)_3$ (216-비트)를 계산할 수 있다. 각각의 출력 차분에 대해 계산하는 키가 8-비트씩만 다르므로 식 (2)를 적용하여
$$2^{280}(1-2^{-176})^{N_1}\approx 2^{181.1}$$
$$2^{8+181.1}(1-2^{-176})^{N_2}\approx 2^{90.2}$$
$$2^{8+90.2}(1-2^{-176})^{N_3}<1$$
을 얻는다 (확률 2^{-176}은 라운드별 확률의 총합이다). 따라서 위 계산에 의해 남은 키가 옳은 키가 된다.

시간 복잡도는 평문 $2^{111.1}$개의 암호화 과정과, 9-라운드 불능 차분 특성의 위 세 라운드에서 추측하는 키 32-비트와 9-라운드 불능 차분 특성의 아래 세 라운드에서 추측하는 키 32-비트에 대한 선택 평문 쌍 N_1, N_2, N_3의 검사 과정 $\le\{(2^{32}N_1+2^{32}N_1)+(2^{32}N_2+2^{32}N_2)+(2^{32}N_3+2^{32}N_1)+(2^{32}N_2+2^{32}N_2)+(2^{32}N_3+2^{32}N_3)\}=2^{216.7}$ (키 테이블 이용)을 더하여 $2^{216.7}$이 된다.

Ⅴ. ARIA에 대한 다중 불능 차분 공격

[11]에서 사용한 불능 차분은 [표 4]의 $[\alpha,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0]\rightarrow_{4R}[0,\gamma,0,0,0,\gamma,0,0,0,0,0,0,\gamma,0,\gamma,0]$이다. 본 논문에서는 다중 불능 차분 공격을 적용하기 위해 다른 형태의 불능 차분 특성과 비교하여 가장 좋은 표 4의 불능 차분$[\alpha,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0]\rightarrow_{4R}[(0,\gamma,0,0,0,\gamma,0,0,0,0,0,0,\gamma,0,\gamma,0),(\gamma,\gamma,0,0,0,0,0,0,0,0,0,0,0,\gamma,0,\gamma,0)]$을 사용한다(본 기호의 처음 4-바이트는 첫 번째 열을 나타낸다).

5.1 다중 불능 차분 공격 적용 방법

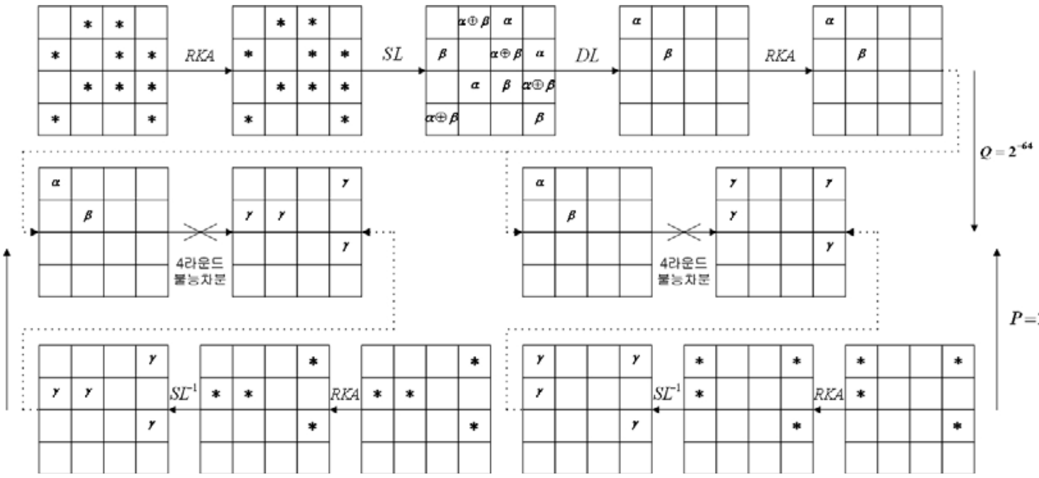
ARIA에 대한 불능 차분 공격은 위의 CLEFIA에 대한 공격과 같은 방법으로 적용한다.

$$2^L(1-p_1)^{N_1}\approx 2^{x_1}$$
$$2^{8+x_1}(1-p_2)^{N_2}<1 \qquad (3)$$

L은 추측하는 키와 계산하는 키의 총 비트 수이다. ARIA에 대한 위 두 개의 출력 불능 차분을 비교해 보면 관여하는 키 부분은 8-비트를 제외한 나머지가 모두 동일함을 알 수 있다. p_i는 i번째 불능 차분 형태에 대한 총 확률이고 N_i는 필터링 후 남아 있는 암호문을 의미하며, 따라서 키 복구 계산 과정을 식 (3)과 같이 두 단계로 계산할 수 있다. 여기서 x_1은 첫 번째 출력 불능 차분 검증 이후에 남은 키의 비트 수이다.

[표 4] 4-라운드 불능 차분 (1) [11] (α, β, γ : 0이 아닌 임의의 8-비트 차분)

입력 불능 차분	출력 불능 차분
$[\alpha,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0]$	$[0,\gamma,0,0,0,\gamma,0,0,0,0,0,0,\gamma,0,\gamma,0]$
	$[\gamma,\gamma,0,0,0,0,0,0,0,0,0,0,\gamma,0,\gamma,0]$
	$[0,\gamma,0,0,0,0,0,0,0,\gamma,\gamma,\gamma,0,0,0,\gamma,0]$
	$[0,\gamma,0,0,0,0,0,0,0,\gamma,0,0,0,\gamma,\gamma,\gamma]$
	$[0,0,\gamma,0,\gamma,0,0,0,0,0,0,0,\gamma,\gamma,\gamma,0]$
	$[0,0,0,0,\gamma,0,0,\gamma,\gamma,0,0,0,\gamma,0,\gamma,0]$



[그림 5] 6-라운드 ARIA 다중 불능 차분 공격

5.2 6-라운드 ARIA에 대한 다중 불능 차분 공격

6-라운드 공격은 그림 5와 같이 4-라운드 다중 불능 차분 특성 위에 한 라운드를 추가하고, 아래에 한 라운드를 추가하여, 추가된 라운드의 관여된 키 비트를 축출한다. 공격 방법은 다음과 같다.

1. 차분이 (0,*,0,*,*,0,*,0,*,*,*,0,0,*,*,*)꼴이 되는 128-비트 평문을 원소로 갖는 structure를 구성한다. (하나의 structure에서 원소의 개수는 $(2^8-1)^{10}$ $\approx 2^{80}$개이고, 만들 수 있는 쌍은 $\binom{2^{80}}{2}\approx 2^{159}$개이다)
2. $2^{30.4}$개의 structure를 선택한다. 4-라운드 입력 차분 $(\alpha,0,0,0,0,\beta,0,0,0,0,0,0,0,0,0,0)$에 대한 불능 차분의 첫 번째 출력 차분 $(0,\gamma,0,0,0,\gamma,0,0,0,0,0,0,0,\gamma,0,\gamma,0)$에 대해 $(0,*,0,0,0,*,0,0,0,0,0,0,0,*,0,*0)$의 차분 형태가 되는 암호문 쌍만 남긴다. 따라서 필터링 후 남는 암호문 쌍의 개수는 $N_1 = 2^{189.4}\times 2^{-96} = 2^{93.4}$개가 된다. 두 번째 출력 차분 $(\gamma,\gamma,0,0,0,0,0,0,0,0,0,0,0,\gamma,0,\gamma,0)$에 대해 (*,*,0,0,0,0,0,0,0,0,0,0,*,0,*0)의 차분 형태가 되는 암호문 쌍만 남기면 필터링 후 남는 암호문 쌍의 개수도 N_1과 마찬가지로 $2^{93.4}$이 된다.
3. 남아 있는 N_1개의 암호문 쌍에 대해 라운드 키 $k_{0,1}$, $k_{0,6}$, $k_{6,1}$ (24-비트)를 추측하면 정리 1에 의해 $k_{0,3}$, $k_{0,4}$, $k_{0,8}$, $k_{0,9}$, $k_{0,10}$, $k_{0,13}$, $k_{0,14}$, $k_{0,15}$, $\boldsymbol{k_{6,5}}$, $k_{6,12}$, $k_{6,14}$ (88-비트)의 키를 계산할 수 있다. 마찬가지로 N_2에 대해 라운드 키 $k_{0,1}$, $k_{0,6}$, $k_{6,1}$ (24-비트)를 추측하면 정리 1에 의해 $k_{0,3}$, $k_{0,4}$, $k_{0,8}$, $k_{0,9}$, $k_{0,10}$, $k_{0,13}$, $k_{0,14}$, $k_{0,15}$, $\boldsymbol{k_{6,0}}$, $k_{6,12}$, $k_{6,14}$ (88-비트)의 키를 계산할 수 있다. 각각의 출력 차분에 대해 계산하는 키가 8-비트씩만 다르다($k_{6,5}$와 $k_{6,0}$). 식 (3)을 적용하면
   $2^{112}(1-2^{-88})^{N_1}\approx 2^{51.1}$
   $2^{8+51.1}(1-2^{-88})^{N_2}<1$
   이 된다. 따라서 위 계산에 의해 남은 키가 옳은 키가 된다.

첫 번째 불능 차분 $(0,\gamma,0,0,0,\gamma,0,0,0,0,0,0,0,\gamma,0,\gamma,0)$에 대한 총 확률 2^{-88}은 다음과 같이 계산한다. (0,*,0,*,*, 0,*,0,*,*,*,0,0,*,*,*)차분 형태가 되는 평문 쌍 (P,P^*)에 대해 (1, 3, 4, 6, 8, 9, 10, 13, 14, 15)번째 바이트의 초기 키 k_0 연산 후 SL 연산을 거쳤을 때, 각각 (1, 10, 15), (6, 8, 13), (3, 4, 9, 14)번째 바이트의 차분이 같아야 한다. 확률은 $(2^{-8})^2\times(2^{-8})^2\times(2^{-8})^4=2^{-64}$이 된다. 그리고 $(0,*,0,0,0,*,0,0,0,0,0,0,0,*,0,*0)$ 차분 형태가 되는 암호문 쌍 (C,C^*)에 대해 (1, 5, 12, 14)번째 바이트의 키 k_6 연산과 SL^{-1} 연산을 거쳤을 때, (1, 5, 12, 14)번째 바이트의 차분이 같아야 한다. 확률은 $(2^{-8})^3=2^{-24}$이다. 그러므로 총 확률은 2^{-88}이 되고, 이는 두 번째 불능 차분 $(\gamma,\gamma,0,0,0,0,0,0,0,0,0,0,0,\gamma,0,\gamma,0)$에 대해서도 동일하다.

시간 복잡도는 평문 $2^{110.4}$개의 암호화 과정과, 4-라운

드 불능 차분 특성의 위 한 라운드에서 추측하는 키 8-비트와 4-라운드 불능 차분 특성의 아래 한 라운드에서 추측하는 키 8-비트에 대한 선택 평문 쌍 N_1, N_2의 검사 과정 $\leq \{(2^{16}N_1 + 2^8 N_1) + (2^{16}N_2 + 2^8 N_2)\} = 2^{110.5}$(키 테이블 이용)을 더하여 $2^{111.5}$이 된다.

Ⅵ. 결 론

본 논문에서는 입력 불능 차분에 대한 여러 개의 출력 불능 차분을 한 번에 고려하여, 불능 차분 공격에서 필요한 평문 쌍과 시간 복잡도를 줄일 수 있는 다중 불능 차분 공격을 제안하였다. 그리고 CLEFIA와 ARIA에 다중 불능 차분 공격을 적용함으로써 본 공격이 기존의 불능 차분 공격보다 더 좋은 효율성을 가짐을 보였다. 본 논문의 공격 결과는 다음과 같다. 본 논문에서 제안한 공격 방법을 CLEFIA에 적용하여 $2^{109.6}$개의 선택 평문과 $2^{109.6}$의 시간 복잡도로 12-라운드 CLEFIA-128의 비밀키를 복구하였으며, $2^{110.3}$개의 선택 평문과 $2^{127.9}$이하의 시간 복잡도로 13-라운드의 CLEFIA-192 및 CLEFIA-256의 비밀키를 복구하였다. 14-라운드와 15-라운드의 CLEFIA-256에 대한 공격은 각각 $2^{110.8}$의 선택 평문과 $2^{176.4}$이하의 시간 복잡도, $2^{111.1}$의 선택 평문과 $2^{216.7}$이하의 시간 복잡도로 비밀키를 복구하였다. 그리고 ARIA에 대한 공격은 $2^{110.4}$의 선택 평문과 $2^{111.5}$ 이하의 시간 복잡도로 6-라운드 ARIA의 비밀키를 복구할 수 있었다. 본 논문에서 제시한 다중 불능 차분 공격이 다른 블록 암호의 분석에 유용한 도구로 사용되기를 기대한다.

참고문헌

[1] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit Blockcipher CLEFIA,” Fast Software Encryption, LNCS 4593, pp. 181-195, 2007.

[2] Sony Corporation, “The 128-bit Blockcipher CLEFIA: Algorithm Specification,” Revision 1.0, June 2007.

[3] Sony Corporation, “The 128-bit Blockcipher CLEFIA: Security and Performance Evaluation,” Revision 1.0, June 2007.

[4] H. Chen, W. Wu, and D. Feng, “Differential Fault Analysis on CLEFIA,” International Conference on Information and Communications Security, LNCS 4861, pp. 284-295, 2007.

[5] W. Wang and X.Y. Wang, “Improved Impossible Differential Cryptanalysis of CLEFIA,” IACR ePrint 2007-466, Dec. 2007.

[6] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki, and H. Kubo, “Impossible Differential Cryptanalysis of CLEFIA,” Fast Software Encryption, LNCS 5086, pp. 398-411, 2008.

[7] B. Sun, R. Li, M. Wang, P. Li, and C. Li, “Impossible Differential Cryptanalysis of CLEFIA,” IACR ePrint 2008-151, Apr. 2008.

[8] D. Kwon, J. Kim, S. Park, S. Sung, Y. Shon, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han, and J. Hong, “New Block Cipher: ARIA,” International Conference on Information and Communications Security, LNCS 2971, pp. 432-445, 2004.

[9] A. Biryukov, C. Canniere, J. Lano, S. Ors, and B. Preneel, “Security and Performance Analysis of Aria,” Version 1.2, Jan. 2004.

[10] W. Wu, W. Zhang, and D. Feng, “Impossible differential cryptanalysis of ARIA and Camellia,” IACR ePrint 2006-350, Oct. 2006.

[11] R. Li, B. Sun, P. Zhang, and C. Li, “New Impossible Differential Cryptanalysis of ARIA,” IACR ePrint 2008-227, May 2008.

[12] 서정갑, 김창균, 하재철, 문상재, 박일환, “블럭 암호 ARIA에 대한 차분전력분석공격,” 정보보호학회논문지, 15(1), pp. 99-107, 2005년 2월.

[13] E. Biham, A. Biryukob, and A. Shamir, “Cryptanalysis of Skipjack Reduced to 31 round using impossible differential,” Advances in Cryptology, EUROCRYPT'99, LNCS 1592, pp. 12-23, 1999.

[14] 문덕재, 황경덕, 이원일, 이상진, 홍석희, “XTEA와 TEA의 축소된 라운드에 대한 불능 차분 공격,” 정보보호학회논문지, 12(4), pp. 77-85, 2002년 8월.

[15] 김종성, 홍석희, 이상진, 임종인, 은희천, “블록 암호 구조에 대한 불능 차분 공격,” 정보보호학회논문지, 13(3), pp. 119-127, 2003년 6월.

[16] 홍석희, 김종성, 김구일, 이창훈, 성재철, 이상진, “30 라운드 SHACAL-2의 불능 차분 공격,” 정보보호학회논문지, 14(3), pp. 1079-115, 2004년 6월.

[17] 김종성, 홍석희, 이상진, 은희천, “6 라운드 AES에 대한 향상된 불능 차분 공격,” 정보보호학회논문지, 15(3), pp. 103-107, 2005년 6월.

<著 者 紹 介>

최 준 근 (Joonggeun Choi) 학생회원
2004년 2월: 고려대학교 수학과 학사
2007년 2월～현재: 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 대칭키 암호의 분석 및 설계

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～현재: 고려대학교 정보보호기술연구센터 연구교수
<관심분야> 대칭키 암호의 분석 및 설계

성 재 철 (Jaechul Sung) 종신회원
1997년 8월: 고려대학교 수학과 학사
1999년 8월: 고려대학교 수학과 석사
2002년 8월: 고려대학교 수학과 박사
2002년 8월～2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월～현재: 서울시립대학교 수학과 조교수
<관심분야> 암호 알고리즘 설계 및 분석

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