Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.3, Jun. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.3.427

MILP를 이용한 ARX 기반 블록 암호의 불능 차분 특성 탐색*

이 호 창,[1†] 강 형 철,[1] 홍 득 조,[2‡] 성 재 철,[3] 홍 석 희[1]
[1]고려대학교, [2]전북대학교, [3]서울시립대학교

Searching for Impossible Differential Characteristics of ARX-Based Block Cipher Using MILP*

HoChang Lee,[1†] HyungChul Kang,[1] Deukjo Hong,[2‡] Jaechul Sung,[3] Seokhie Hong[1]
[1]Korea University, [2]Chonbuk National University, [3]University of Seoul

요 약

불능 차분 특성은 해당 블록 암호를 랜덤 치환과 구별을 해주며, 키 복구 공격에도 사용될 수 있다. 최근 Cui 등이 ARX 기반 블록 암호에 대해 Mixed-Integer Linear Programming(MILP)를 이용해서 자동화된 불능 차분 특성 탐색 방법을 제안하였다. Cui 등이 제안한 방법을 개선하면 기존에 방법에 비해 더 적은 선형 제약식을 이용하여 기존 방식으로 찾지 못하는 불능 차분 특성까지 추가 되어 더 많은 불능 차분 특성을 찾아 낼 수 있다. 수정한 방법을 이용하여 SPECK family와 LEA에 적용하였다. SPECK32, SPECK48, SPECK64, SPECK96에 대해 7-라운드, SPECK128에 대해 8-라운드 불능 차분 특성을 찾아내었다. 이 불능 차분 특성은 모두 새롭게 발견된 것이다. LEA에 대해서는 기존의 10-라운드인 불능 차분 특성을 포함한 새로운 10-라운드 불능 차분 특성을 찾아내었다.

ABSTRACT

Impossible differential characteristics distinguish the corresponding block cipher from random substitution and can also be used for key recovery attack. Recently Cui et al. proposed an automatic method for searching impossible differential characteristics of several ARX - based block ciphers using Mixed Integer Linear Programming(MILP). By optimizing the method proposed by Cui et al., It was possible to find new impossible differential characteristics which could not be founded by the method by using less linear constraint expression than the existing method. It was applied to the SPECK family and LEA using the modified method. We found 7-rounds for SPECK32, SPECK48, SPECK64, SPECK96 and 8-rounds impossible differential characteristics of SPECK128. These impossible differential characteristics are all newly found. We also found existing 10-rounds of impossible differential characteristic and new 10-rounds of impossible differential characteristics of LEA.

Keywords: MILP, ARX, Impossible Differential Cryptanalysis, Impossible Differential Characteristic, Automatic Search, SPECK, LEA

Received(03. 13. 2017), Modified(04. 26. 2017), Accepted(04. 28. 2017)

* 본 논문은 2016년 동계 학술대회에 발표한 우수논문을 개선 및 확장한 것임

* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대 정보x컴퓨팅기술개발사업의 지원을 받아 수 행된 연구입니다(No. NRF-2014M3C4A7030649)

* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.B0722-16-0006, (창조씨앗-2단계)암호와 물리계층보안을 결합한 IoT 네트워크 보안 기술 개발)

† 주저자, lhc254@korea.ac.kr

‡ 교신저자, deukjo.hong@jbnu.ac.kr(Corresponding author)

Table 1. Impossible Differential Characteristics Searched Using MILP Method

block cipher	number of round	number of found char.	reference
SPECK32	7	1	[this paper]
SPECK48	7	1	[this paper]
SPECK64	7	21	[this paper]
SPECK96	7	426	[this paper]
SPECK128	8	8	[this paper]
LEA	10	1 / 3	[21] / [this paper]

I. 서 론

불능 차분 특성(Impossible Differential Characteristic)은 절대로 성립하지 않는 차분 특성이다. 즉, 0이 아닌(Nonzero) 입력 차분 $\Delta\alpha$로부터 어떤 출력 차분 $\Delta\beta$가 유도될 확률이 0일 때, 해당 차분 특성은 불능 차분 특성이 된다. 이것을 $\Pr(\Delta\alpha \rightarrow \Delta\beta) = 0$으로 표시하자. n비트 입·출력 길이를 갖는 랜덤 치환은 임의의 입·출력 쌍 $(\Delta\alpha, \Delta\beta)$에 대해 확률 $\Pr(\Delta\alpha \rightarrow \Delta\beta) = 2^{-n}$을 갖기 때문에, 만약 n비트 블록 길이를 갖는 어떤 블록 암호에서 불능 차분 특성이 발견된다면 그것은 그 자체로 랜덤 치환에 대한 구별자(Distinguisher)로서 의미가 있다. 불능 차분 특성은 또한 블록 암호의 키 복구 공격에 활용 될 수도 있다[1,2]. 불능 차분 특성을 찾기 위한 방법으로 $\mathscr{U}$-method[3], UID-method[4]와 두 방법을 포함하는 확장 툴[5] 등 여러 불능 차분 특성 탐색 방법들이 존재한다. 하지만 이러한 방법들은 비선형 부분의 성질을 고려하지 않았다는 한계가 있다.

프로그래밍 문제(Programming Problem)는 주어진 조건을 만족하는 목적 함수(Objective Function)의 최댓값 또는 최솟값을 구하는 수학적 최적화 문제이다. Mixed-Integer Linear Programming(MILP)는 제약식이 선형이고 일부 변수가 정수인 프로그래밍 문제이다. Mouha 등[6]과 Wu 등[7]은 MILP 방법을 이용해 word-based 블록 암호에 대해서 active S-Box의 수를 세는 방법을 소개하였다. 이를 확장시켜 Sun 등[8]은 SPN 구조를 사용하는 블록 암호에 MILP를 적용하는 방법을 소개하였다.

최근에는 MILP 방법을 이용해서 많은 연구들이 이루어 지고 있다. Fu 등[9]이 ARX 기반 블록 암호에 대해 차분 확률과 선형 근사식(Linear Approximation)을 계산 하는 방법을 제시하였다. Cui 등[10]과 Sasaki 등[11]은 블록 암호에 대한 불능 차분 특성 자동 탐색법을 제안하였다. Xiang 등[12]과 Sun 등[13,14]은 Integral 공격을 확장한 Division 성질을 찾는 법을 제안하였다.

Cui 등이 제안한 MILP를 이용해 불능 차분 특성을 탐색하는 방법은 $\mathscr{U}$-method, UID-method와 같은 기존의 방법들과 달리 비선형 부분의 성질까지 고려하여 더 많은 불능 차분 특성을 찾을 수 있도록 도와준다. 하지만 Cui 등이 제안한 방법은 Fu 등이 제안한 차분 확률 계산 방법을 응용한 것으로, 모델링 방법에서 불능 차분 특성 탐색에는 필요없는 차분 확률 계산 부분이 들어있다. 따라서 Cui 등이 제안한 방법을 불능 차분 특성 탐색에 대해 최적화 할 경우, 탐색 성능의 향상을 가져올 수 있다. Cui 등이 제안한 mycallback 함수 때문에 찾아내지 못하는 불능 차분 특성도 탐색 할 수 있다.

본 논문에서는 주어진 ARX 구조 블록 암호에 대해 가장 긴 불능 차분 특성을 최대한 효율적으로 찾는 방법을 연구한다. 그 목적을 위하여 Cui 등이 제안한 MILP 기반 이용한 ARX 구조 블록 암호 블록 암호 불능 차분 특성 자동 탐색법을 개선한다. 개선된 방법을 SPECK family와 LEA에 적용한 결과, SPECK32, SPECK48, SPECK96에 대해 7-라운드, SPECK128에 대해 8-라운드 불능 차분 특성이 발견되었다. 이것들은 모두 새롭게 발견된 것들이다. LEA에 대해서는 기존의 10-라운드 불능 차분 특성[21]과 새로운 10-라운드 불능 차분 특성이 찾아내었다. 적용 결과를 정리하면 Table 1.과 같다.

본 논문은 2장에서는 표기법과 MILP를 이용한 불능 차분 특성 탐색 방법에 대해서 소개하고, 3장에서는 이를 향상 시킬 수 있는 방법에 대해서 소개하고, 4장에서 SPECK family와 LEA에 적용 방

법 및 결과를 소개한다. 마지막으로 5장에서는 결론을 맺는다.

II. 사전 지식

2.1 표기법

- ¬: 보수(complement)
- ⊕: 배타적 논리합(eXclusive OR, XOR)
- ⊞: 법 덧셈(modular addition)
- $\gg a$($\ll a$): 오른쪽(왼쪽)으로 a번 비트 순환이동
- $\alpha[i]$: a의 i번째 최하위 비트
- α_j^i: i-라운드 워드 α의 j번째 최하위 비트
- Δ_x: 설정한 입력 차분의 집합
- Δ_y: 설정한 출력 차분의 집합
- $a\%m$: $a \pmod m$
- W_A^i: 집합 A의 원소위치에 차분이 있는 i번째 워드
- W_Z^i: 차분이 없는 i번째 워드
- $(\alpha,\beta \rightarrow \gamma)$: 입력 차분 α,β와 출력 차분 γ
- $d_{\oplus}$: 더미 비트

2.2 MILP를 이용한 ARX 기반 블록 암호의 차분 특성 탐색 방법

Fu 등[9]은 Lipmaa와 Moriai[15]의 정리를 이용하여 Mixed-Integer Linear Programming(MILP)로 자동화된 차분 특성 탐색 방법을 제안하였다. MILP 문제는 선형 제약식을 만족하는 값에 대해서 목적 함수의 값을 최적화 하는 문제이다. MILP를 이용하여 차분 특성을 탐색하기 위해서는 목적 함수를 차분 특성의 확률을 나타내는 식으로 설정하고, 선형 제약식은 해당 암호 시스템을 구성하는 식으로 구성한다. 구성한 암호 시스템에 대해서 MILP 문제의 답으로 해당 암호 시스템을 구성하는 최적의 차분 특성 확률을 구하는 것이다.

목적 함수를 설정하기 위해서 Lipmaa와 Moriai가 제안한 두 가지 정리를 이용하였고, 그 정리는 다음과 같다.

정리 1. 법 덧셈 차분의 존재성 [15]
법 덧셈 차분 $(\alpha,\beta \rightarrow \gamma)$가 0이 아닌 확률을 가질 필요 충분 조건은 아래의 2가지 조건이다.

1. $\alpha[0] \oplus \beta[0] \oplus \gamma[0] = 0$
2. $\alpha[i-1] = \beta[i-1] = \gamma[i-1] = \alpha[i] \oplus \beta[i] \oplus \gamma[i]$
($\alpha[i-1] = \beta[i-1] = \gamma[i-1], i \in [1, n-1]$ 경우)

정리 2. 법 덧셈 차분의 확률 [15]
법 덧셈 차분 $(\alpha,\beta \rightarrow \gamma)$가 0이 아닌 확률을 가질 경우, 그 확률은 $2^{-\sum_{i=0}^{n-2} \neg eq(\alpha[i],\beta[i],\gamma[i])}$ 이다. 이 때 함수 $eq(\alpha[i],\beta[i],\gamma[i])$는 아래와 같다.

$$eq(\alpha[i],\beta[i],\gamma[i]) = \begin{cases} 1 & (\alpha[i] = \beta[i] = \gamma[i]) \\ 0 & (else) \end{cases}$$

Lipmaa와 Moriai가 제안한 두 가지 정리로 법 덧셈에 대한 차분 특성의 표현 및 확률 계산이 가능하다. 또한 $eq(\alpha[i],\beta[i],\gamma[i])$을 이용해 MILP 문제의 목적 함수로 표현할 수 있다.

ARX 기반 블록 암호를 선형 제약식으로 구성하기 위해서는 ARX 기반 블록 암호를 이루는 연산인 법 덧셈, 비트 순환이동, XOR을 표현하면 된다. 그 표현 방법은 다음과 같다.

법 덧셈: Lipmaa와 Moriai가 제안한 정리를 만족하는 점들을 이용해 벡터 $(\alpha[i],\beta[i],\gamma[i],\alpha[i+1],\beta[i+1],\gamma[i+1],\neg eq(\alpha[i],\beta[i],\gamma[i]))$로 표현할 경우 56개의 벡터가 생긴다. 56개의 벡터만을 표현할 수 있는 13개의 선형 제약식으로 법 덧셈의 성질을 표현하였다.

비트 순환이동: 법 덧셈에서 사용되는 비트 인덱스를 수정해서 사용한다.

XOR: 세 비트의 XOR 상태인 $a \oplus b \oplus c = 0$을 표현하기 위해서 아래의 5개의 선형 제약식으로 표현한다.

$$\begin{cases} d_{\oplus} \geqq a, d_{\oplus} \geqq b, d_{\oplus} \geqq c \\ a+b+c \geqq 2d_{\oplus} \\ a+b+c \leqq 2 \end{cases}$$

위 방법으로 목적 함수, 선형 제약식을 설정하여 ARX 기반 블록 암호에 대해서 MILP 기반 차분 특성 탐색이 가능하다.

Algorithm 1. Searching for Impossible Differential Characteristic[10]

Input:
Δ_x: A set of input differences to search for
Δ_y: A set of output differences to search for
r: The number of rounds you want to find
Output:
r-round impossible differential characteristics
1. // impossible differential characteristic searching function
2. procedure find_impossible_diff(Δ_x, Δ_y, r) do
3. for $i_x \in \Delta_x$ do
4. for $o_y \in \Delta_y$ do
5. create a r-round model "idc.lp" with input and output differences i_x and o_y respectively.
6. m = read("idc.lp") // open the model
7. m.optimize(mycallback) // optimize using the mycallback function
8. if (m.status == 3) then // if model's status is infeasible
9. save i_x and o_y

2.3 MILP를 이용한 불능 차분 특성 탐색 방법

[10]에서는 [9]에서 제안하는 차분 확률 계산 방법을 이용하여 ARX 기반 블록 암호에 대해서 불능 차분 특성 탐색 방법을 제안하였다. [9]의 방법을 이용해 특정 입·출력 차분 값에 대해서 MILP 문제의 답을 구할 수 없는 경우, 해당 입·출력 차분 값으로 주어진 암호 시스템으로 차분 특성을 구성 할 수 없는 것을 의미한다. 따라서 해당 입·출력 차분 값은 주어진 암호 시스템의 해당 라운드의 불능 차분 특성이 된다. 즉, [10]에서 제안하는 불능 차분 특성 탐색 방법은 [9]의 차분 확률 계산 방법에서 MILP 문제의 답을 구할 수 없는 입·출력 차분 값을 찾는 것으로 MILP 기반 불능 차분 특성 탐색이 진행된다.

[10]에서 ARX 기반 블록 암호를 선형 제약식으로 구성하기 위해서 XOR을 표현 하는 방식을 제외하고 [9]의 선형 제약식을 그대로 이용하였으며 XOR을 표현한 방법은 다음과 같다.

XOR) 세 비트의 XOR 상태인 $a \oplus b \oplus c = 0$을 표현하기 위해서 1개의 더미 변수를 이용하여 다음과 같이 1개의 선형 제약식으로 표현한다.

$$a + b + c = 2d_{\oplus}$$

[10]에서 암호 시스템을 최적화 프로그램인 Gurobi[16]의 입력 형태인 LP파일 형태로 만들어 MILP 문제를 푸는 것으로 불능 차분 특성을 찾는 알고리즘을 Algorithm 1.과 같이 제안하였다. 7번 줄의 mycallback 함수의 경우 Gurobi 프로그램이 해당 입·출력 차분 값으로 구성되는 하나의 경로라도 찾을 경우 최적화를 종료하기 위한 콜백 함수이다.

III. 불능 차분 특성 탐색 능력 향상 방법

[10]에서 제안한 불능 차분 특성 탐색 방법은 [9]의 차분 확률 계산 방법을 이용하는 MILP 기반 불능 차분 특성 탐색 방법이다. 하지만 불능 차분 특성 탐색의 경우 가능 여부만 확인하면 되는 반면 [10]에서 제안한 방법은 [9]의 방법을 이용하였기 때문에 차분 확률 계산 부분도 포함되어있다. 즉, 가능의 여부만 고려하면 되는 불능 차분 특성 탐색에 대한 최적의 방법이 아니다. 따라서 차분 확률 계산 부분을 고려하지 않는 방법으로 최적화하여 가능해를 하나라도 구하는 경우 불능 차분 특성이 아닌 것을 이용하는 방법으로 [10]에서 제안한 방법의 성능을 향상 시킬 수가 있다.

3.1 선형 제약식 최적화

[10]에서 [9]의 방법을 이용하여 불능 차분 특성 방법을 제안하였다. 하지만 불능 차분 특성 탐색에만

초점을 맞출 경우 [10]에서 제안한 방법보다 제약식의 수를 줄일 수 있다. [10]에서는 i번째 라운드 법 덧셈 차분의 성질을 56개의 벡터 $(\alpha[i],\beta[i],\gamma[i],\alpha[i+1],\beta[i+1],\gamma[i+1],\neg eq(\alpha[i],\beta[i],\gamma[i]))$로 표현하였고, 이를 해로 가지는 13개의 선형 제약식을 사용하였다. 하지만 벡터의 $\neg eq(\alpha[i],\beta[i],\gamma[i])$ 부분은 확률 계산을 위한 부분으로 불능 차분 특성 탐색에는 필요하지 않다. 따라서 벡터의 해당 부분을 고려하지 않을 경우 i번째 법 덧셈 차분의 가능성을 표현하는 56개의 벡터 $(\alpha[i],\beta[i],\gamma[i],\alpha[i+1],\beta[i+1],\gamma[i+1])$로 Table 2.와 같다.

Table 2.의 벡터를 [17]에서 소개 된 방법으로 SAGE 프로그램[18]의 inequality_generator() 함수를 이용해서 20개의 선형 제약식을 얻을 수 있다. 또한 greedy algorithm[19]을 통해 Table 2.를 해로 가지는 8개의 선형 제약식만을 남길 수 있다. 입력 차분 α,β와 출력 차분 γ인 8개의 선형 제약식은 Table 3.과 같다.

법 덧셈 차분 확률까지 고려하는 13개의 선형 제약식에서 가능 여부만을 표현하는 8개의 선형 제약식으로 줄임으로써 [10]에서 사용되는 선형 제약식의 개수보다 약 30% 줄일 수 있다. 또한 확률 계산 부분을 제외하였으므로 기존 방법의 목적 함수 부분이 더 이상 필요 없으므로 null로 변경한다.

Table 2. Vectors of Modular Difference

(0,0,0,0,0,0),(0,0,0,0,1,1),(0,0,0,1,0,1),(0,0,0,1,1,0),
(0,0,1,0,0,0),(0,0,1,0,0,1),(0,0,1,0,1,0),(0,0,1,0,1,1),
(0,0,1,1,0,0),(0,0,1,1,0,1),(0,0,1,1,1,0),(0,0,1,1,1,1),
(0,1,0,0,0,0),(0,1,0,0,0,1),(0,1,0,0,1,0),(0,1,0,0,1,1),
(0,1,0,1,0,0),(0,1,0,1,0,1),(0,1,0,1,1,0),(0,1,0,1,1,1),
(0,1,1,0,0,0),(0,1,1,0,0,1),(0,1,1,0,1,0),(0,1,1,0,1,1),
(0,1,1,1,0,0),(0,1,1,1,0,1),(0,1,1,1,1,0),(0,1,1,1,1,1),
(1,0,0,0,0,0),(1,0,0,0,0,1),(1,0,0,0,1,0),(1,0,0,0,1,1),
(1,0,0,1,0,0),(1,0,0,1,0,1),(1,0,0,1,1,0),(1,0,0,1,1,1),
(1,0,1,0,0,0),(1,0,1,0,0,1),(1,0,1,0,1,0),(1,0,1,1,0,0),
(1,0,1,1,0,0),(1,0,1,1,0,1),(1,0,1,1,1,0),(1,0,1,1,1,1),
(1,1,1,0,0,1),(1,1,1,0,1,0),(1,1,1,1,0,0),(1,1,1,1,1,1)

Table 3. Linear Inequalities Only for Possibility of Modular Difference

$\alpha[i]+\beta[i]+\gamma[i]-\alpha[i+1]+\beta[i+1]+\gamma[i+1]\geqq 0$
$\alpha[i]+\beta[i]+\gamma[i]+\alpha[i+1]-\beta[i+1]+\gamma[i+1]\geqq 0$
$\alpha[i]+\beta[i]+\gamma[i]+\alpha[i+1]+\beta[i+1]-\gamma[i+1]\geqq 0$
$\alpha[i]+\beta[i]+\gamma[i]-\alpha[i+1]-\beta[i+1]-\gamma[i+1]\leqq 2$
$\alpha[i]+\beta[i]+\gamma[i]-\alpha[i+1]-\beta[i+1]-\gamma[i+1]\geqq -2$
$\alpha[i]+\beta[i]+\gamma[i]-\alpha[i+1]+\beta[i+1]+\gamma[i+1]\leqq 4$
$\alpha[i]+\beta[i]+\gamma[i]+\alpha[i+1]-\beta[i+1]+\gamma[i+1]\leqq 4$
$\alpha[i]+\beta[i]+\gamma[i]+\alpha[i+1]+\beta[i+1]-\gamma[i+1]\leqq 4$

3.2 새로운 불능 차분 특성 탐색

[10]에서 제안하는 mycallback 함수는 MILP 문제를 풀기위한 모델과 모델의 연산 진행 상황을 입력으로 받아 Branch-and-Bound 방법으로 최적화 중의 목적 함수 값의 바운드에 따라 종료하는 함수이다. 따라서 mycallback 함수로 종료될 경우 해당 모델의 상태가 infeasible이 아닌, 해를 가지는 것을 이용하여 불능 차분 특성인지 아닌지를 판단하는 방식으로 Algorithm 1.이 진행된다.

[10]에서 불능 차분 특성을 탐색하는 방법은 특정 입·출력 차분 값으로 최적화를 진행하였을 때, 모델의 상태가 'infeasible'일 경우 해당 입·출력 차분 값으로 차분 특성이 이루어지지 않는 것으로 불능 차분 특성으로 결정한다. 하지만 [10]에서 제안하는 mycallback 함수의 경우 특정 불능 차분 특성의 입·출력 차분 값에 대해서 최적화 과정을 종료하여 해당 상태가 infeasible을 만들지 않아 불능 차분 특성으로 결정하지 않는다. 본 논문에서는 불능 차분 특성 탐색에 최적화 함에 따라 목적 함수를 넣을 필요가 없어졌고, 따라서 mycallback 함수를 사용하지 않다도 된다. 그 결과 최적화한 선형 제약식과 mycallback 함수를 사용하지 않고 탐색하여 불능 차분 특성 탐색을 실행할 경우 기존 방법으로 찾지 못하는 불능 차분 특성을 탐색할 수 있다. 예를 들어 SPECK64의 경우 mycallback 함수로 152개의

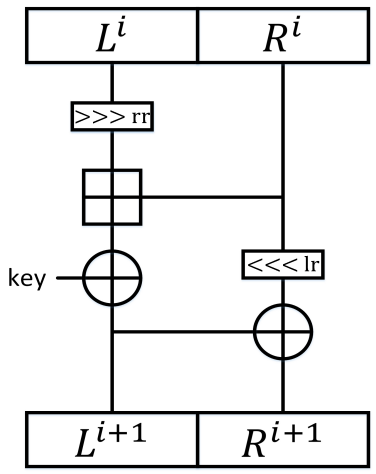

Fig. 1. ith Round of SPECK

불능 차분 특성을 찾을 수 있었다. 하지만 본 논문에서 최적화한 선형 제약식으로 콜백 함수 없이 실행할 경우 157개의 불능 차분 특성을 얻을 수 있었다.

IV. ARX 기반 블록 암호에 적용

본 장에서는 3장의 방법을 이용해 ARX 기반 블록 암호인 SPECK family와 LEA에 적용한 방법과 불능 차분 특성을 소개한다.

4.1 SPECK family에 적용

4.1.1 SPECK family

SPECK[20]은 NSA가 2013년에 공개한 10가지 종류를 가진 ARX 기반 경량 블록 암호이다. i번째 라운드 함수는 Fig.1.과 같다.

rr과 lr값 각각 왼쪽 워드와 오른쪽 워드의 비트 순환이동을 의미하며 SPECK family에 대한 라운드는 Table 4.와 같다.

Table 4. rr,lr and Round Number of SEPCK family

type	key size	rr	lr	number of rounds
SPECK32	64	7	2	22
SPECK48	72	8	3	22
	96			23
SPECK64	96			26
	128			27
SPECK96	96			28
	144			29
SPECK128	128			32
	192			33
	256			34

4.1.2 SPECK family의 불능 차분 특성

SPECK2m ($m\in\{16,24,32,48,64\}$)의 i번째 라운드를 선형 제약식으로 표현하면 아래와 같다.

$$L^i_{rr\%m}+R^i_0+L^{i+1}_0=2d^i$$

$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}+R^i_{j+1}+L^{i+1}_{j+1}\geqq 0$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j+L^i_{(j+1+rr)\%m}-R^i_{j+1}+L^{i+1}_{j+1}\geqq 0$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j+L^i_{(j+1+rr)\%m}+R^i_{j+1}-L^{i+1}_{j+1}\geqq 0$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}-R^i_{j+1}-L^{i+1}_{j+1}\leqq 2$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}-R^i_{j+1}-L^{i+1}_{j+1}\leqq -2$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}+R^i_{j+1}+L^{i+1}_{j+1}\leqq 4$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}+R^i_{j+1}+L^{i+1}_{j+1}\leqq 4$$
$$L^i_{(j+rr)\%m}+R^i_j+L^{i+1}_j-L^i_{(j+1+rr)\%m}+R^i_{j+1}+L^{i+1}_{j+1}\leqq 4$$
$$R^i_{(j-lr)\%m}+L^{i+1}_j+R^{i+1}_j=2d^i_j \qquad (j\in[0,m-1])$$

d의 경우 dummy bit를 의미하며 $j+rr(\mathrm{mod}m)$과 $j-lr(\mathrm{mod}m)$으로 비트 순환이동까지 한 번에 표현할 있다.

SPECK2m의 경우 한 라운드를 표현하는데 총 $9m-7(=1+8(m-1)+m)$개의 선형 제약식이 필요하게 되며 n-라운드 불능 차분 특성을 찾기 위해서는 입·출력 차분 값에 대한 제약식($4m$개)까지 포함하여 총 $n(9m-7)+4m$개의 제약식이 필요하다. 이러한 필요량은 Cui 등이 제안한 방법의 선형 제약식의 수보다 약 35% 적은 양이다.

최적화한 선형 제약식을 이용해 Cui 등의 탐색방법으로 탐색한 SPECK family의 불능 차분 특성은 Table 5.와 같다. 모든 입·출력 차분 범위는 탐색의 경우의 수가 계산 능력을 뛰어넘기 때문에 입·출력 차분의 헤밍 무게를 각각 1또는 2로 제한하여 탐색 범위를 계산 가능한 범위로 설정하였다. Table 5.의 weight 열은 순서대로 입력 차분의 헤밍 무게, 출력 차분의 헤밍 무게를 의미한다. number of rounds 열은 해당 weight 조건에서 가장 긴 불능 차분 특성을 의미한다.

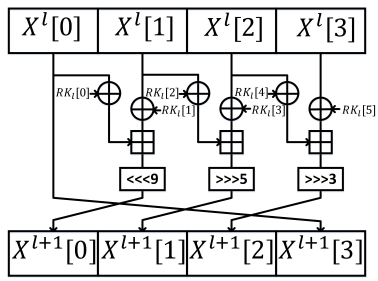

Fig. 2. lth Round of LEA

Table 5. Impossible Differential Characteristic of SPECK family and LEA

block cipher	number of linear inequalities in one round [3]	number of linear inequalities in one round [this paper]	weight	number of rounds	number
SPECK32	212	137	1 / 1	6	3
			2 / 1		1
SPECK48	324	209	1 / 1	6	20
			2 / 1		27
SPECK64	436	281	1 / 1	6	157
			2 / 1		410
SPECK96	660	425	1 / 1	7	12
			2 / 1		35
SPECK128	884	559	1 / 1	7	160
			2 / 1		503
LEA	11,164	6,979	1 / 1	9	27

4.2 LEA에 적용

4.2.1 LEA

LEA는 Hong 등[21]이 제안한 ARX 기반 경량 블록 암호이다. 128bit의 블록 사이즈와 128, 192, 256bit의 키 사이즈로 구성되며 각각 24, 28, 32라운드이다. LEA의 i번째 라운드 함수는 Fig.2.와 같다.

4.2.2 LEA의 불능 차분 특성

LEA의 i번째 라운드를 선형 제약식으로 표현하기 위해서는, 각 법 덧셈마다 249개($=8\times31+1$)개의 선형 제약식이 필요하다. 따라서 n-라운드 불능 차분 특성을 찾기 위해서는 입·출력 차분 값에 대한 선형 제약식(256개)까지 포함하여 총 $747n+256$개($=(3\times249)n+256$)의 선형 제약식이 필요하다. 이러한 필요량은 Cui 등이 제안한 방법의 선형 제약식의 수보다 약 38% 적은 양이다.

새로 구성한 선형 제약식을 이용해 Cui 등의 탐색 방법으로 탐색한 LEA의 불능 차분 특성은 Table 5. 와 같다.

4.3 특성의 연장

Table 5.의 결과는 모두 입·출력 차분의 헤밍 무게를 1 또는 2로 제한하여 탐색한 결과이다. 계산능력의 한계로 헤밍 무게가 1 또는 2가 아닌 불능 차분 특성에 대해서는 탐색이 어렵기 때문이다. 하지만 SPECK family와 LEA의 헤밍 무게를 1 또는 2로 제한하여 탐색한 불능 차분 특성을 1의 확률로 연장하여 헤밍 무게가 1 또는 2가 아닌 불능 차분 특성을 만들 수 있다. 본 절에서는 탐색 범위를 확장하지 않고, 헤밍 무게가 1 또는 2가 아닌 더 긴 라운드의 불능 차분 특성을 만드는 해당 조건과 연장된 불능 차분 특성을 소개한다.

4.3.1 SPECK의 불능 차분 특성 연장

SPECK의 오른쪽 워드의 최상위 비트에만 차분이 있을 경우, 확률 1로 한 라운드 암호화한 차분의 형태를 얻을 수 있다. 즉, m비트 워드에 대해서 각각의 SPECK family는 다음을 만족한다.

$$(W_Z^1\ W_{m-1}^2)\xrightarrow[encrypt]{1-round}(W_{m-1}^1\ W_{\{m-1,\alpha\}}^2)$$

따라서 MILP를 이용하여 찾은 SPECK family의 불능 차분 특성에 대해서 입력 차분이 해당 성질을 만족시킬 경우 한 라운드를 연장 시킬 수 있으며, 역시 불능 차분 특성을 만족한다. MILP를 이용해 찾은 SPECK32, SPECK48, SPECK64, SPECK128의 가장 긴 라운드 불능 차분 특성을 각각 1개, 1개, 21개, 7개를 한 라운드 연장 시킬 수 있었다. SPECK96의 경우 7-라운드 불능 차분 특성에서 연

Table 6. 8-round Impossible Differential Characteristic of SPECK128

$$(W_Z^1\ W_{38}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_Z^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_{\{45,37\}}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_{\{46,38\}}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_{\{46,39\}}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_{\{47,38\}}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$
$$(W_4^1\ W_{\{48,38\}}^2) \not\rightarrow_8 (W_{63}^1\ W_{\{63,3\}}^2)$$

장 할 수 있는 특성은 존재하지 않았지만 6-라운드 불능 차분 특성에서 391개의 차분 특성을 한 라운드 연장 할 수 있었다. SPECK128의 8-라운드 불능 차분 특성은 Table 6.과 같다.

4.3.2 LEA의 불능 차분 특성 연장

LEA의 마지막 워드의 최상위 비트에만 차분이 있을 경우, 확률 1로 한 라운드 복호화 한 차분은 각 워드 최상의 비트에만 차분이 있다. 즉, 32비트 워드에 대해서 다음을 만족한다.

$$(W_Z^1\ W_Z^2\ W_Z^3\ W_{\{31\}}^4) \xrightarrow[decrypt]{1-round} (W_{\{31\}}^1\ W_{\{31\}}^2\ W_{\{31\}}^3\ W_{\{31\}}^4)$$

따라서 MILP를 이용하여 찾은 LEA의 불능 차분 특성에 대해서 입력 차분이 해당 성질을 만족시킬 경우 한 라운드를 연장 시킬 수 있으며, 역시 불능 차분 특성을 만족한다. LEA의 9-라운드 불능 차분 특성의 탐색 결과 중 연장 시킬 수 있는 불능 차분 특성 3개를 발견했고, 연장한 3개의 10-라운드 불능 차분 특성은 Table 7.과 같다. 이중 3번째 특성이 [21]에서 소개된 10-라운드 불능 차분 특성이고 이외 2개는 새롭게 발견된 것이다.

Table 7. 10-round Impossible Differential Characteristic of LEA

$$(W_{\{31\}}^1\ W_{\{31\}}^2\ W_{\{31\}}^3\ W_{\{31\}}^4) \not\rightarrow_{10} (W_Z^1\ W_Z^2\ W_{\{27\}}^3\ W_Z^4)$$
$$(W_{\{31\}}^1\ W_{\{31\}}^2\ W_{\{31\}}^3\ W_{\{31\}}^4) \not\rightarrow_{10} (W_Z^1\ W_Z^2\ W_{\{28\}}^3\ W_Z^4)$$
$$(W_{\{31\}}^1\ W_{\{31\}}^2\ W_{\{31\}}^3\ W_{\{31\}}^4) \not\rightarrow_{10} (W_Z^1\ W_Z^2\ W_{\{29\}}^3\ W_Z^4)$$

V. 결 론

본 논문에서는 Cui 등[10]이 제안한 ARX 기반 블록 암호에 대한 MILP를 이용한 자동화된 불능 차분 특성 탐색 방법을 최적화하는 방법을 제안하였다. 그 결과를 이용해 더 적은 선형 제약식을 이용하여 SPECK32, SPECK48, SPECK64, SPECK96에 대해 7-라운드 불능 차분 특성을 각각 1개, 1개, 21개, 426개, SPECK128에 대해서 8-라운드 불능 차분 특성 8개를 찾아내었다. 이 불능 차분 특성은 모두 새롭게 발견된 것들이고 기존 결과에 비해 더 긴 라운드를 갖는다. LEA에 대해서는 기존의 10-라운드 불능 차분 특성과 새로운 10-라운드 불능 차분 특성 2개를 찾아내었다. 발견된 불능 차분 특성을 이용하여 키 복구 공격에 적용을 차후 연구 계획 중이다.

References

[1] Knudsen, Lars. "DEAL-a 128-bit block cipher." complexity vol.258, no.2, pp. 216, 1998.

[2] Biham, Eli, Alex Biryukov, and Adi Shamir. "Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials." International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, pp. 12-23, May. 1999.

[3] Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S. "Impossible differential cryptanalysis for block cipher structures." In International Conference on Cryptology in India Springer Berlin Heidelberg, pp. 82-96, Dec. 2003.

[4] Luo, Y., Lai, X., Wu, Z., & Gong, G. "A unified method for finding impossible differentials of block cipher structures." Information Sciences, vol.263, pp. 211-220, 2014.

[5] Wu, Shengbao, and Mingsheng Wang. "Automatic search of truncated impossible differentials for word-oriented block ciphers." International Conference

on Cryptology in India. Springer Berlin Heidelberg, pp. 283-302, Dec. 2012.

〔6〕 Mouha, N., Wang, Q., Gu, D., & Preneel, B. "Differential and linear cryptanalysis using mixed-integer linear programming." In International Conference on Information Security and Cryptology pp. 57-76, Springer Berlin Heidelberg. Nov. 2011.

〔7〕 Wu, Shengbao, and Mingsheng Wang. "Security Evaluation against Differential Cryptanalysis for Block Cipher Structures." IACR Cryptology ePrint Archive vol.2011, pp. 551, 2011.

〔8〕 Sun, S., Hu, L., Song, L., Xie, Y., & Wang, P. "Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks." In International Conference on Information Security and Cryptology. Springer International Publishing, pp. 39-51, Nov. 2013.

〔9〕 Fu, K., Wang, M., Guo, Y., Sun, S., & Hu, L. "Milp-based automatic search algorithms for differential and linear trails for speck." In International Conference on Fast Software Encryption. Springer Berlin Heidelberg, pp. 268-288, Mar. 2016.

〔10〕 Cui, T., Jia, K., Fu, K., Chen, S., & Wang, M. "New automatic search tool for impossible differentials and zero-correlation linear approximations." Cryptology ePrint Archive, Report 2016/689. 2016.

〔11〕 Sasaki, Yu, and Yosuke Todo. "New impossible differential search tool from design and cryptanalysis aspects." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Cham, pp. 185-215, Apr. 2017.

〔12〕 Xiang, Z., Zhang, W., Bao, Z., & Lin, D. "Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers." In Advances in Cryptology - ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22. Springer Berlin Heidelberg, pp. 648-678, Dec. 2016.

〔13〕 Sun, Ling, Wei Wang, and Meiqin Wang. "Milp-aided bit-based division property for primitives with non-bit-permutation linear layers." IACR Cryptology ePrint Archive, 2016: 811, 2016.

〔14〕 Sun, Ling, et al. "MILP-Aided Bit-Based Division Property for ARX-Based Block Cipher." IACR Cryptology ePrint Archive vol.2016, pp. 1101, 2016.

〔15〕 Lipmaa, Helger, and Shiho Moriai. "Efficient algorithms for computing differential properties of addition." International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, pp. 336-350, Apr. 2001.

〔16〕 http://www.gurobi.com/

〔17〕 Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., & Song, L. "Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers." In International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, pp. 158-178, Dec. 2014.

〔18〕 http://www.sagemath.org/

〔19〕 Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X.,& Fu, K.. "Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties." Cryptology ePrint Archive, Report 2014/747. 2014

〔20〕 Beaulieu, R., Treatman-Clark, S., Shors,

D., Weeks, B., Smith, J., & Wingers, L. "The SIMON and SPECK lightweight block ciphers." In Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE, pp. 1-6, IEEE. Jun. 2015.

[21] Hong, D., Lee, J. K., Kim, D. C., Kwon, D., Ryu, K. H., & Lee, D. G. "LEA: A 128-bit block cipher for fast encryption on common processors." In International Workshop on Information Security Applications. Springer International Publishing, pp. 3-27, Aug. 2013.

〈저자소개〉

이 호 창 (HoChang Lee) 학생회원
2016년 2월: 서울시립대학교 수학과 졸업
2016년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호 알고리즘 설계 및 분석, 자동화 탐색

강 형 철 (Hyngchul Kang) 학생회원
2010년 2월: 고려대학교 산업시스템정보공학과 학사 졸업
2010년 3월~현재: 고려대학교 정보보호대학원 석박사통합과정
〈관심분야〉 블록 암호화 해쉬 함수 설계 및 분석, 인증 암호화 모드 설계

홍 득 조(Deukjo Hong) 종신회원
1999년 8월: 고려대학교 수학과 학사
2001년 8월: 고려대학교 수학과 석사
2006년 2월: 고려대학교 정보보호대학원 박사
2006년 3월~2007년 12월: 고려대학교 정보보호기술연구센터 연구교수
2007년 12월~2015년 8월: 국가보안기술연구소 선임연구원
2015년 9월~현재: 전북대학교 IT정보공학과 조교수
〈관심분야〉 암호 알고리즘 설계 및 분석

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 전임강사, 조교수, 부교수, 교수
〈관심분야〉 암호 알고리즘 설계 및 분석

홍 석 희 (SeokHie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식