Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.3, Jun. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.3.533

효율적인 MILP-Espresso 기반 차분 특성 자동 탐색 방법*

박 연 지,[1†] 이 호 창,[2] 홍 득 조,[3‡] 홍 석 희[1]
[1]고려대학교, [2]국가보안기술연구소, [3]전북대학교

MILP-Espresso-Based Automatic Searching Method for Differential Charactertistics*

YeonJi Park,[1†] HoChang Lee,[2] Deukjo Hong,[3‡] Seokhie Hong[1]
[1]Korea University, [2]National Security Research Institute, [3]Chonbuk National University

요 약

본 논문은 Sasaki 등이 2018년도에 제안한 차분 특성 자동화 탐색 방법을 개선하여 MILP Solver로 정확하고 간편하게 S-box 기반 암호의 차분 특성을 탐색하는 방법을 제안한다[13]. Sasaki 등이 제안한 방식은 차분 특성에 대한 제약식 설계에 입력 차분과 출력 차분만을 변수로 포함하여 확률을 별도로 계산한 반면, 논문에서 제안하는 탐색 방법은 입력 차분, 출력 차분, 확률 변수를 하나의 제약식으로 구성하여 한 번의 프로그램 실행으로 특성과 확률을 동시에 확인할 수 있도록 탐색 절차를 간소화 하였다. 또한 본 논문에서는 Sasaki 등이 활용했던 Espresso 알고리즘과 Quine- McCluskey 알고리즘 중에서 제약식이 보다 대폭 축소되는 Espresso 알고리즘을 이용해 제약식을 간소화 하였다. 본 논문에서 제안하는 탐색 방법은 다양한 구조와 블록 사이즈에 적용 가능함을 입증하기 위해 블록암호 GIFT-64, GIFT-128, SKINNY-64에 적용하였다. 적용 결과, GIFT의 경우 기존의 제안 논문에서 4라운드의 최적의 차분 특성이 5개의 활성 S-box를 가진다고 제시한 반면, 본 논문을 적용한 결과에서는 활성 S-box의 개수는 6개이지만 기존보다 좋은 $2^{-11.415}$의 확률을 갖는 최적의 차분 특성을 찾는 성과가 있었다. SKINNY-64의 경우 기존보다 개선된 결과는 아니지만 제안 논문에서의 분석과 동일한 활성 S-box를 갖는 차분 특성을 찾을 수 있었다.

ABSTRACT

In this paper, we propose an MILP-based method for Optimal Probability of Bit-based Differential Characteristic in SP(Substitution-permutation) ciphers based on Automatic Differential Characteristic Searching Method of Sasaki, et al[13]. In [13], they used input/output variables and probability variables seperatably, but we simplify searching procedure by putting them(variables) together into linear inequalities. Also, In order to decrease the more linear inequalities, we choose Espresso algorithm among that used by Sasaki, et al(Quine-McCluskey algorithm & Espresso algorithm). Moreover, we apply our method to GIFT-64, GIFT-128, SKINNY-64, and we obtained results in the GIFT(Active S-boxs : 6, Probabilities : $2^{-11.415}$) compared with the existing one.(Active S-boxs : 5, Probabilities : unknown). In case of SKINNY-64, we can't find better result, but can find same result compared with the existing one.

Keywords: MILP, Bit-based, Differential, Automatic search, Logic minimization algorithm, Espresso, GIFT, SKINNY

Received(02. 09. 2018), Modified(05. 23. 2018), Accepted(05. 24. 2018)

* 본 논문은 2017년 동계 학술대회에 발표한 우수논문을 개선 및 확장한 것임

* 본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00520, (ICT 기초연구실) SCR-Friendly 대칭키 암호 및 응용모드 개발 / No.B0722-16-0006, 암호와 물리계층보안을 결합한 IoT 네트워크 보안 기술 개발)

† 주저자, yeonta555@naver.com

‡ 교신저자, deukjo.hong@gmail.com(Corresponding author)

I. 서 론

차분 분석(differential cryptanalysis)은 Eli Biham과 Adi Shamir에 의해 제안되어 현재까지 블록암호의 안전성 분석에 보편적으로 사용되는 방법이다.[1] 차분 분석은 평문 쌍의 차분 값(ΔP)을 알면 그에 대응되는 암호문의 차분 값(ΔC)을 확률적으로 예측할 수 있다는 사실에 기반하며, 이를 통해 차분 특성(differential characteristic)을 구성하여 키(key)를 찾아낸다.

차분 특성을 탐색하는데 있어 자동화 도구는 매우 효율적으로 사용될 수 있다. 자동화 도구는 대칭키 암호의 분석과 디자인 설계 등에 사용되는 도구인데 그 중 Mouha, Wu, Wang등에 의해 Mixed Integer Linear Programming(MILP) Solver가 차분 특성 탐색에 사용될 수 있음이 알려지면서 블록 암호의 분석에 이용되기 시작하였다[2]. MILP 문제는 선형부등식 형태의 제약식과 주어진 식의 최댓값 또는 최솟값을 구해주는 목적 함수로 구성되며 MILP Solver는 목적 함수를 만족하는 최적의 결과를 찾아준다.

MILP와 관련된 기존 연구로는 Mouha등이 워드 단위로 MILP를 사용해 차분 특성 및 선형 특성(linear characteristic)을 탐색하였으며[2] 이후 Sun등에 의해 비트 단위로 차분 특성을 탐색함으로써 차분의 확산 형태를 보다 정확하게 분석 가능함을 보였다[3]. Fu등은 ARX구조 암호에 대해 MILP Solver를 이용한 모델링을 적용하여 차분 및 선형 특성을 탐색하였으며[4], Sasaki, Todo등은 SPN 구조에서 불능차분을 찾는 방법을[5], Xiang등은 디비젼 성질(division property)를 이용하여 인테그랄 구별자(integral distinguisher)를 찾는 방법을 제안하였다[6]. 이렇듯 많은 논문에서 MILP Solver를 이용하여 암호학적 특성에 대해 탐색하는 다양한 방법을 제시했지만 탐색 범위 및 깊이에 따라 증가하는 탐색 시간을 줄이는 문제는 아직 풀어야할 숙제로 남아있다. 일반적으로 라운드가 길어지거나, 분석해야할 암호의 크기가 커지거나, 제약식이 많아질수록 탐색에 많은 시간이 걸리며 특히 비트 기반 탐색은 연산량이 지수 단위로 증가하여 보다 정확한 특성과 확률을 계산할 수 있음에도 널리 쓰이지 못하였다.

Sasaki 등은 앞에서 언급한 문제를 해결하여 최초로 8비트 이상의 S-box에 대해 MILP Solver를 이용한 차분 특성 탐색 방법을 2018년 제안하였다[13]. Sasaki 등이 제안한 방법은 차분 특성 탐색을 위해 가능한 각 확률별로 차분 분포표를 만들어 목적 함수를 구성한 뒤, 각 차분 분포표에 대응되는 입·출력 차분이 발생하면 이에 대응되는 확률에 값을 부여하여 최종적으로 확률이 최대화 되는 차분 특성을 출력하는 방법이다. 또한 차분 특성을 탐색하는데 사용하는 제약식을 Espresso 알고리즘을 이용하여 간소화 하였으며 Quine- McClusky 알고리즘을 사용한 논리식 최소화 결과와 비교하였다[13].

본 논문에서 소개할 탐색 방법은 앞의 방법을 개선하여 보다 간단한 절차를 통해 탐색하기 위한 것으로 Sasaki 등이 제안한 방법이 차분 특성을 탐색하는데 사용되는 제약식 내에 입·출력 변수만을 포함하고 확률과 관련된 테이블을 별도로 사용하여 조건을 부여하는 반면, 본 논문에서 제시하는 방법은 제약식 내에 입·출력 변수와 확률 변수를 포함하여 동시에 처리함으로써 차분 특성의 탐색과 확률 계산을 별도로 할 필요 없이 한 번에 최적화 된 차분 특성과 그에 대한 확률을 구할 수 있다. 제약식의 간소화는 Espresso 알고리즘을 이용하였으며 이를 통해 절차와 제약식을 간소화하여 효율성을 개선하였다.

본 논문에서 제안하는 방법은 블록암호 GIFT-64에 적용하였으며 다양한 블록크기에서의 결과를 보이기 위해 GIFT-128에도 적용하였다[7]. 또한 GIFT와 다른 permutation 구조를 갖는 S-box기반 암호에 대해서도 적용 가능함을 보이기 위해 SKINNY-64에 대한 탐색 결과도 제시하였다[10]. 차분 특성 탐색 결과, SKINNY-64의 경우 제안 논문에서의 결과와 동일한 활성 S-box를 갖는 차분 특성을 탐색할 수 있었으며 이에 대한 정확한 확률을 계산할 수 있었다. GIFT에 대해서는 제안 논문에서 명시된 기존 결과보다 개선된 결과를 확인할 수 있었는데, 4라운드에서 최적의 차분 특성이 5개의 활성 S-box를 가진다고 제시한 기존 결과와 달리 본 논문의 방법을 적용한 결과 활성 S-box의 개수는 6개이지만 기존보다 좋은 $2^{-11.415}$의 확률을 갖는 최적의 차분 특성을 찾을 수 있었다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 표기법과 차분 특성 탐색 방법 등 배경 지식에 대해 소개하고, Ⅲ장에서는 이를 이용한 S-box 기반 암호의 차분 특성 탐색 방법을 소개한다. Ⅳ장에서는 제시한 방법을 GIFT-64, GIFT-128, SKINNY-64에 적용한 것과 그 결과를 제시하며, 마지막으로 Ⅴ장에서

결론을 맺는다.

II. 배경 지식

2.1 표기법

본 논문에 사용된 표기법은 다음과 같다.

- o Δ: 차분 값($\Delta P = P \oplus P'$)
- o $S(P)$: S-box를 통과한 P의 결과 값
- o $\oplus$: 배타적 논리합(exclusive-OR, XOR)
- o $\wedge$: 논리곱(logical conjunction, AND)
- o $\vee$: 논리합(logical disjunction, OR)
- o $\overline{X}$: 변수 X에 대한 논리부정(logical complement, NOT)
- o x_i: x의 오른쪽에서 i번째 비트(MSB)
- o per[i]: S-box들의 전체 출력 중 i번째 비트를 전치 함수에 입력한 결과 값
- o X[r][0~15]: 변수 X의 r번째 라운드 0~15번 비트 (총 16비트)

2.2 차분 특성

차분 분석은 암호 분석의 한 방법으로 입력 값의 변화(입력 차분)에 따른 출력 값의 변화(출력 차분)를 일정 확률로 예측할 수 있음을 이용하는 방법이다. 선형 구조에서는 입·출력간 차분의 변화를 1의 확률로 예측할 수 있으며 비선형 구조에 대해서는 일정 확률로 차분의 변화를 예측할 수 있다.

비선형 구조에서의 차분 특성을 S-box를 예로 살펴보자. S-box는 특정 비트의 값을 입력 받아 이에 대응되는 특정 비트의 값을 출력한다. 차분은 여러 가지 형태를 가질 수 있으나 통상적으로 사용하는 차분은 XOR 차분이며 본 논문에서도 차분으로 명시되는 것은 XOR 차분을 의미한다. Fig. 1.을 통해 보면 S-box의 두 입력값이 P와 P' 이고, 그에 대응하는 두 출력값이 C와 C'일 때 S-box에서의 입력 차분은 $\Delta P = P \oplus P'$, 출력 차분은 $\Delta C = C \oplus C'$ 이다.

n비트가 입력되어 n비트가 출력되는 S-box에서 모든 입·출력 차분이 발생할 확률이 편차가 없이 균일하다면 특정 입력 차분에 대해 특정 출력 차분이 나타날 확률은 $1/2^n$로 동일해야 하지만 실제로는 이러한 분포가 나타나지는 않는다. 그렇기 때문에 이러한 차분의 성질이 차분 분석에 이용되며, 분석을 위해 분석 대상의 가능한 모든 입력 차분과 가능한 모든 출력 차분을 표로 만들어 가능한 입·출력 차분의 분포를 나타낸 것이 차분 분포표(Differential Distribution Table, DDT)이다. Table. 1.은 Fig. 2.의 3비트 S-box에 대한 차분 분포표로 세로축은 입력 차분을 나타내며 가로축은 출력 차분을 나타낸다. 예를 들자면 $001_{(2)}$이라는 차분을 가지는 쌍이 입력되는 경우 가능한 모든 8개의 경우에 대해 출력될 수 있는 차분은 8번 중 4번은 $010_{(2)}$, 4번은 $101_{(2)}$이라는 차분을 가지는 쌍이 출력된다는 것을 의미한다. 즉, 입력 차분값(ΔP)이 $001_{(2)}$로 S-box에 입력되면 출력 차분값(ΔC)이 $010_{(2)}$이 될 확률은 4/8 = 1/2 이다.

이러한 차분 성질을 바탕으로 각 라운드 별 발생할 확률이 높은 입·출력 차분을 연결한 것을 차분 특성(Differential Characteristic)이라고 한다. 차분 특성의 확률 계산은 각 라운드의 확률이 독립적으로 발생한다는 가정 하에 각 라운드 별 모든 S-box의 예상 입력 차분에서 예상 출력 차분이 나올 확률을

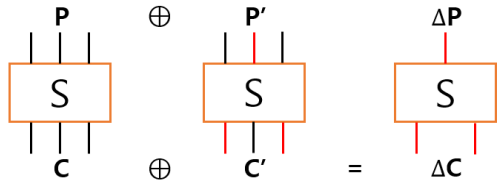

Fig. 1. XOR differential in S-box

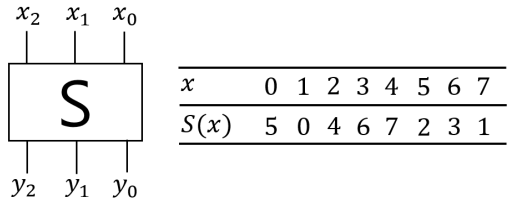

Fig. 2. 3×3-bit S-box example

Table 1. 3bit S-box DDT example

ΔP \ ΔC	000	001	010	011	100	101	110	111
000	8	0	0	0	0	0	0	0
001	0	0	4	0	0	4	0	0
010	0	2	0	2	2	0	2	0
011	0	2	0	2	2	0	2	0
100	0	0	4	0	0	0	0	4
101	0	0	0	0	0	4	0	4
110	0	2	0	2	2	0	2	0
111	0	2	0	2	2	0	2	0

곱하여 계산한다. 그렇기 때문에 높은 확률을 가진 입·출력 차분을 연결하여 차분 특성을 구성하면 효율적으로 차분 분석을 할 수 있게 된다.

2.3 워드 기반 / 비트 기반 차분 특성 탐색 방법

MILP로 차분 특성을 탐색하려면 입력과 출력에 대한 제약식을 설정하여 해당 알고리즘에 있어 변수들이 가질 수 없는 불가능한 값들을 제약식을 통해 걸러내고 목적 함수를 통해 가능한 조합을 최적화하도록 암호를 모델링하는 작업이 필요하다. 모델링 방법은 Fig. 3.에서 나타내는 것과 같이 워드 기반(word-based)과 비트 기반(bit-based) 두 가지 방식으로 나뉜다.

워드 기반 방식은 최소한의 활성 S-box를 지나는 차분 특성이 가장 높은 확률을 갖는 차분 특성이라는 가정에서 시작한다. 활성 S-box란 입력 차분이 존재하는 S-box로 활성 S-box가 많아질 경우 차분이 예상한 경로를 따라 확산될 확률을 계산하는데 있어 곱해지는 1보다 작은 확률 값이 많아지므로 대체로 전체 확률 값이 낮아지는 결과를 가져온다. 그렇기 때문에 최소한의 활성 S-box를 만드는 차분 특성이 가장 높은 확률을 가질 가능성이 크다. 활성 S-box를 표현하기 위해 S-box의 입·출력은 입·출력 유무에 따라 워드 단위로 표현하고, 차분 분포표에 기반하여 출력 차분이 발생할 가능성에 따라 S-box를 최소한으로 거치는 차분 특성을 구성한다. 워드 기반 방식은 입력과 출력을 단순화하여 연산시간이 줄어든다는 장점이 있지만 차분의 정확한 확산을 표현할 수 없다는 단점이 있다.

비트 기반 방식은 입력 차분과 출력 차분을 비트단위로 표현하는 방식이다. 실제 차분 분포표에 따라 차분의 입·출력 경로를 구성하며 이에 따라 차분의 정확한 확산 및 차분 특성의 확률을 구할 수 있다는 장점이 있는 반면 탐색에 필요한 연산시간이 비트의 지수 단위로 증가한다는 단점이 있다. 본 논문에서는 보다 정확한 값을 표현하기 위해 비트 기반 방식을 활용하였으며 이를 식으로 표현하는데 부울(boolean) 함수와 합의 곱(product-of-sum)을 이용했다.

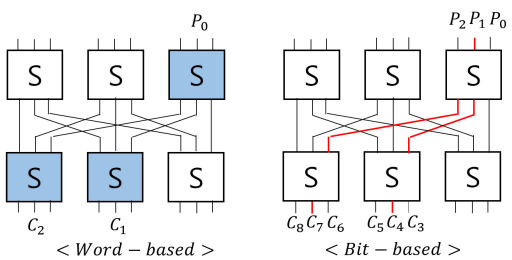

Fig. 3. word-based and bit-based differential propagation

2.4 부울(Boolean) 함수 표현

부울식은 2진 변수들간의 관계를 논리곱(AND), 논리합(OR), 논리부정(NOT)을 사용하여 나타낸 식으로 곱의 합(Disjunctive Normal Form, DNF) 또는 합의 곱(Conjunctive Normal Form, CNF)의 형태로 표현할 수 있다. 곱의 합 형태는 참(true)의 결과를 가지는 변수들의 논리곱을 각각의 절로 만들어 각 절들을 논리합으로 연결한 것이며, 합의 곱 형태는 거짓(false) 결과를 가지는 변수들의 부정값의 논리합을 각각의 절로 만들어 각 절들을 논리곱으로 연결한 것이다. 예를 들면 Table. 2.에서 Value가 1이면 결과가 참이고 0이면 거짓이라고 하자. Table. 2.가 나타내는 참의 결과를 곱의 합 형태로 표현하면 $(\overline{X}\wedge\overline{Y})\vee(\overline{X}\wedge Y)$으로 표현할 수 있으며, 합의 곱 형태로 표현하면 $(\overline{X}\vee Y)\wedge(\overline{X}\vee\overline{Y})$으로 표현할 수 있다.

본 논문에서는 합의 곱 형태를 지원하는 논리식 최소화 알고리즘을 사용하므로 합의 곱 형태의 부울식을 부울함수로 사용할 것이다. S-box에 대한 입·출력을 합의 곱 형태로 표현해 보자. Table. 2.에서 X와 Y를 입력으로, Value를 출력으로 본다면 n비트 입력과 n비트 출력을 갖는 S-box는 부울 함수로 표현했을 때 2n비트의 입력과 1비트의 출력을 갖는 CNF 형태의 $f(x,y)$로 표현할 수 있다.

$$f(x,y)=\bigwedge_{c\in 0,1^{2n}}((\bigvee_{i=1}^{n}(x_i\oplus c_i))\bigvee(\bigvee_{i=1}^{n}(y_i\oplus c_i)))$$

각각의 입·출력 값은 논리합으로 연결된 절로 구성되며, 차분 분포표에 의해 가능한 입력 차분과 출력 차분의 조합은 결과값을 1로 두고($c_i=1$) 불가능한 차분들의 결과값은 0으로($c_i=0$) 분류한다. 각 절은 논리곱 연산에 의해 연결되어 각 절 중 하나라도 만족하지 않는 경우 값이 참이 될 수 없으므로

Table 2. truth-table example

X	Y	Value
0	0	1
1	0	0
0	1	1
1	1	0

$f(x,y) \geq 1$이라는 제약식을 통해 결과 값이 1이 되는 조합만 남기고 불가능한 출력은 쉽게 걸러낼 수 있게 된다. 이렇듯 부울 함수 표현을 통해 n비트 입력과 n비트 출력을 갖는 S-box연산을 표현할 수 있는 제약식을 만들어 낼 수 있다.

2.5 Espresso 알고리즘[12]

Espresso 알고리즘은 IBM의 Robert Brayton에 의해 제안된 논리식 최소화 알고리즘으로 경험적(Heuristic) 알고리즘이다. 결정적(Deterministic) 알고리즘인 Quine-McCluskey 알고리즘보다 더 간략한 형태로 최소화가 가능하며 최소화에 적용되는 변수의 개수에 제한이 없어 변수가 아무리 많더라도 비교적 쉽게 처리된다는 장점이 있으나 최소화 결과가 항상 최솟값이라는 보장이 없다는 단점이 있다. 이러한 단점에도 불구하고 메모리 사용과 계산시간이 몇 배로 줄어든다는 점과 실제 최솟값과 결과값의 차이가 근소하여 표준 논리식 최소화 알고리즘으로 쓰이고 있다.

최소화는 큐브(cube)와 카누맵(karnaugh map)의 개념을 활용하여 3단계에 거쳐 진행된다. 첫 번째 단계에서는 n개의 변수를 갖는 부울함수를 2^n개의 꼭지점을 갖는 n차원 큐브로 표현한다. n차원 큐브는 여러 개의 저차원 큐브로 표현할 수 있으며 저차원 큐브들은 n차원 큐브를 표현 가능하도록 확장된다.

2, 3단계에서는 확장된 큐브들을 카누맵으로 나타내어 잉여를 제거하고 통합한다. 카누맵을 이용하는 방법은 각 항을 포함된 1의 개수에 따라 분류한 뒤 중복되는 부분을 서로 통합하는 것이다. 예를 들어 A는 2진 벡터로 표시하면 '1000'이고 B는 '1001'이면 둘을 합쳐서 '100-' 이라는 값을 만든다. 여기서 '-'로 명시된 부분은 0이나 1 무엇이든 상관이 없는 자리임을 의미한다.

위와 같은 방법으로 잉여 큐브를 제거하고 더 이상 제거 및 통합할 것이 없어지면 이를 이용하여 논리식을 간소화하여 표현할 수 있다.

III. S-box 기반 암호의 차분 특성 탐색 방법

3.1 S-box의 bit-based 차분 특성 표현

앞서 Ⅱ장에서 제시된 Table. 1.의 결과를 보면 입력 차분 3비트와 출력 차분 3비트에 대해 가능한 확률은 $000_{(2)} \rightarrow 000_{(2)}$인 경우를 제외하고 1/4 또는 1/2의 확률을 가짐을 알 수 있다. 입·출력 차분이 두 확률 중 하나의 확률을 가질 때 해당 변수가 1의 값을 갖도록 변수화하여 입력 차분, 출력 차분, 확률을 하나의 벡터로 표현하면 Table. 1.의 차분 분포는 다음과 같은 형태로 표현할 수 있다.

$$v = (x_2, x_1, x_0, y_2, y_1, y_0, p^{1/2}, p^{1/4})$$

벡터v를 Ⅱ장 2.4절에서 나타낸 것과 같이 합의 곱으로 표현하려면 각 절을 0(false)의 결과를 가지는 변수들의 부정 값의 논리합으로 구성하면 된다. Table. 1.에서 0의 출력을 갖는 값 전부를 벡터v를 이용해 제약식을 만들면 다음과 같은 형태의 부울 함수가 된다.

$$f(x,y,p) = (x_2 \vee x_1 \vee x_0 \vee y_2 \vee \overline{y_1} \vee y_0 \vee \overline{p^{1/2}} \vee p^{1/4}) \\ \ldots \wedge (\overline{x_2} \vee \overline{x_1} \vee \overline{x_0} \vee \overline{y_2} \vee \overline{y_1} \vee y_0 \vee p^{1/2} \vee \overline{p^{1/4}})$$

위 식에서 논리합으로 연결된 각 절은 벡터v와 1:1로 대응되어 $(x_2 \vee x_1 \vee x_0 \vee y_2 \vee \overline{y_1} \vee y_0 \vee \overline{p^{1/2}} \vee p^{1/4})$는 '00001010', 즉, Table. 1.에서 나올 수 없는 조합 중 하나인 P(0x0→0x2) = 1/2을 나타낸다. 이러한 방법으로 S-box에 대한 제약식을 만들 수 있으며 동시에 벡터에 포함된 확률 변수 $p^{1/2}, p^{1/4}$에 각각 가중치를 적용하여 목적 함수를 만들면 목적 함수를 이용하여 최적화된 차분 특성의 확률 값을 바로 구할 수 있게 된다.

3.2 논리식 최소화 알고리즘 적용

Ⅲ장 3.1절의 결과는 차분 분포표상의 불가능한 조합이 모두 표현된다. Table. 1.의 S-box에서 0이 아닌 출력을 갖는 값은 23개 이므로 합의 곱으로 표현되어 제약식에 포함되는 벡터 수는 $2^8 - 23$개다. 제약식의 수가 많아지면 통상적으로 탐색 속도 또한

느려지는 경향이 있기 때문에 제약식을 단순화하기 위해 논리식 최소화 알고리즘이 활용되며 그 중 Espresso 알고리즘을 거친 제약식은 다음과 같은 형태의 벡터 모음으로 출력된다.

10-01--0 1

이 벡터는 Ⅲ장 3.1절의 벡터v에서 각 자리가 나타내는 변수와 대응되며 0은 $+x_i$, 1은 $+(1-x_i)$을 나타낸다. '-'는 0이든 1이든 어떠한 값이 들어가도 상관없음을 의미하고 맨 끝의 1은 벡터의 결과가 1비트 결과 값으로 출력됨을 의미한다. 즉, '10-01-0'는 $(1-x_2)+x_1+y_2+(1-y_1)+p^{1/4} \geq 1$, 상수를 정리하면 $-x_2+x_1+y_2-y_1+p^{1/4} \geq -1$이라는 제약식 절이 되며, 이를 통해 원래 사용되어야 하는 양보다 적은 제약식 절을 사용하여 S-box의 가능한 입·출력 차분만을 표현할 수 있게 된다.

3.3 MILP를 이용한 최적화

MILP 문제를 만들기 위해서는 목적 함수 설정, 제약식 작성, 변수 선언이 필요하다. 본 논문에서는 MILP Solver 프로그램으로는 Gurobi Optimizer를 활용하였으며 MILP 문제는 LP파일로 만들어 실행하였다.

입력 차분과 출력 차분, 한 S-box에서 출력 가능한 확률을 변수로 놓고 Ⅲ장 3.1절 및 Ⅲ장 3.2절의 과정을 거쳐 제약식을 작성, r라운드 모델을 만든다.

이후 해당 모델에 가능한 모든 비트 조합을 기 선언한 변수들에 대입하여 목적 함수를 가장 최적화 해주는 비트 조합을 찾아낸다. S-box 기반 암호의 차분 특성 탐색에서 목적 함수는 차분 특성에서 나타나는 확률들의 가중치(확률을 2^{-n}이라고 할 때, n값) 합이 최소가 되는 값이다.

목적 함수에 대한 최적화가 끝나면 각 라운드별 입력 차분과 출력 차분, 확률을 나타내는 변수에 대입된 비트 값을 확인하여 최적화 된 차분 특성을 확인할 수 있다. Algorithm. 1.은 Gurobi Optimizer로 제약식 작성, 목적 함수 최적화, 결과 출력을 하는데 사용한 알고리즘의 의사코드이다.

Algorithm. 1. efficient method of searching optimal differential characteristics

```
Input: ΔX, ΔY, p
Output: Optimal Differential trail
1. Procedure find Differential trail(ΔX, ΔY, p) do
2.   for i∈ΔX do
3.     for o∈ΔY do
4.       for s∈p do
5.         creat r-ROUND model "DC.lp"
6.         m = read("DC.lp")
7.         m.optimize()
8.         for i 0 to r-1 do
9.         write ΔX, ΔY, p
```

Ⅳ. 탐색 방법 적용

4.1 GIFT-64의 차분 특성 탐색

GIFT는 2017년 Subhadeep Banik 등에 의해 제안된 경량 블록 암호이다[7]. PRESENT와 같은 디자인 전략을 가져왔지만 선형공격에서 PRESENT의 단점을 보완하고 보다 빠르고 작게 구현 가능한 형태로 효율성을 높인 암호이다. GIFT는 64, 128 두 종류의 블록 사이즈를 가지며 각각 28라운드, 40라운드를 갖는다. S-box는 공통적으로 4bit S-box를 사용하며 SubCell, PermBits, AddRound Key 함수로 구성된다. 전체 구조는 Fig. 5.와 같다.

Fig. 4.의 GIFT S-box는 4bit의 6-uniform S-box로 입·출력 차분에 대해 가능한 확률은 $2^{-1.415}, 2^{-2}, 2^{-3}$ 3가지가 있다.

S-box에 대한 입력 차분 4비트, 출력 차분 4비트와 가능한 확률 3가지는 다음과 같은 형태의 벡터로 표현하였다.

$$v = (x_3, x_2, x_1, x_0, y_3, y_2, y_1, y_0, p^{3/8}, p^{1/4}, p^{1/8})$$

Ⅲ장 3.1절의 방법으로 불가능한 조합의 벡터를 모두 출력한 결과 1950개의 벡터를 얻을 수 있었으며, 그 결과를 Espresso 알고리즘과 Quine-McCluskey 알고리즘으로 최소화하여 Table. 3.과 같은 결과 값을 얻을 수 있었다. 결과적으로 경험적 알고리즘인 Espresso 알고리즘을 이용하여 결정적 알고리즘인 Quine-McCluskey 알고

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$GS(x)$	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

Fig. 4. GIFT S-box(4-bit)

Table 3. minimized linear inequalities (boolean vectors) with logic minimization algorithm

	Initial count	QM	Espresso
GIFT S-box	1950	105	48

리즘보다 2배 이상 제약식을 축소시킬 수 있었다.

MILP를 이용한 최적화를 위해서 LP파일을 생성하였으며 LP파일에는 목적 함수, 제약식, 변수 순으로 필요항목이 작성되었다. 변수는 각 라운드별 입·출력 차분과 가능한 확률을 2차원 배열로 표현하였으며 배열의 행은 라운드, 열은 비트의 위치(확률의 경우 S-box의 위치)를 의미한다. 즉 입·출력 차분은 $X[r][i]$로($0 \le i < 64$), 총 16개의 4비트 S-box에 대응되는 각각의 확률은 $p^i[r][j]$($i \in 3/8,1/4,1/8$, $0 \le j < 16$)로 표현하여 아래와 같이 LP파일 내에 작성하였다.

Binary

$X[0][0] \ldots X[0][63], p^{3/8}[0][0], \ldots p^{1/8}[0][15]$

$p^{3/8}[r-1][0] \ldots p^{1/8}[r-1][15], X[r][0] \ldots X[r][63]$

목적 함수는 아래 식과 같이 작성하였으며 이때 각 확률 변수가 나타내는 확률 값이 다르므로 확률 변수에 가중치를 준 뒤에 가중치를 준 확률 변수 값들의 합을 최소화하도록 하였다.(차분 특성의 확률 2^{-n}에서 n값을 최소화 하는 것)

$$Obj = 1.415\,p^{3/8}[0][0] + 2p^{1/4}[0][0] + \ldots\ldots$$
$$+ 2p^{1/4}[r-1][15] + 3p^{1/8}[r-1][15]$$

제약식은 논리식 최소화 알고리즘을 통해 도출된 제약식들이 각 S-box마다 적용되므로 Espresso 알고리즘을 적용한 경우 제약식의 총 개수는 48×16+1(입력 차분이 0이 아님을 나타내는 식 포함)개 이다. 입력 변수는 $X[r][i]$ 출력 변수는 y대신 $X[r+1][per[i]]$를 대입하며 $+(1-x_i)$형태는 MILP의 특성상 상수를 모두 부등식의 우변으로 옮겨서 처리하였다.

$$-X[0][3]-X[0][2]-X[0][1]+X[0][0]-X[1][17] \ge -3$$
$$\vdots$$
$$X[r-1][63]-X[r-1][62]-X[r-1][61]+$$
$$X[r-1][60]+X[r][62]-X[r][28]+p^{3/8}[r-1][15] \ge -2$$

GIFT-64에 대한 새로운 MILP 모델 적용 결과는 Table. 4.와 같다.

총 7라운드에 대해 라운드별 가장 높은 확률의 차분 특성을 확인할 수 있었으며 해당 특성에 대한 입력 차분 및 출력 차분의 정확한 위치도 확인할 수 있었다. GIFT에 대해 차분 특성과 확률을 정확히 탐색한 이전 연구가 없기 때문에 라운드별 차분 특성에 대한 비교 대상은 없으나 GIFT 제안 논문인 [7]의

Table 4. optimal probability of differential characteristic(GIFT-64)

Round	$-\log_2 p$	time	Active S-box	Active S-box[7]
1	1.415	0.07s	1	1
2	3.415	0.24s	2	2
3	7	3s	3	3
4	11.415	1.7m	6	5
5	17	4.5m	7	7
6	22.415	58m	10	10
7	28.415	4h	13	13

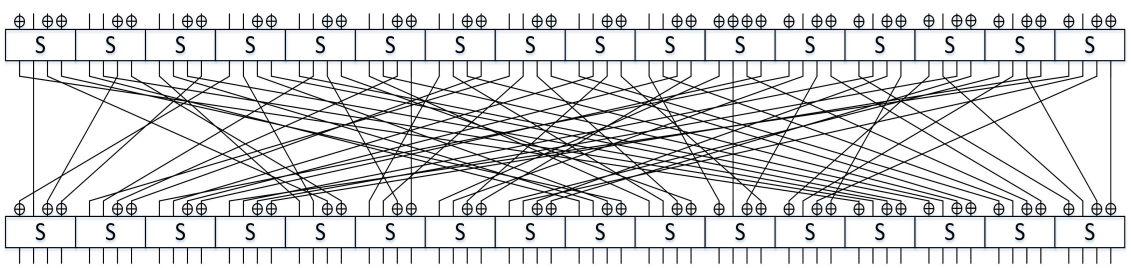

Fig. 5. GIFT-64 structure

결과와 비교하였을 때 활성 S-box 개수는 제안 논문보다 많음에도 목적 함수에 의해 최적화된 가장 높은 확률을 갖는 특성을 찾을 수 있었다.

4.2 GIFT-128의 차분 특성 탐색

GIFT-128은 GIFT-64에서 블록 사이즈만 2배로 커진 암호이다 S-box는 동일하게 Fig. 4.를 사용하는 것으로 탐색하였기 때문에 입·출력 차분을 나타내는 변수의 개수와 전치함수(permutation)외에 나머지는 GIFT-64와 동일하게 적용하였다.

GIFT-128에 대한 새로운 MILP 모델 적용 결과 총 5라운드에 대해 Table. 5.와 같은 결과를 확인할 수 있었으며 해당 특성에 대한 입력 차분 및 출력차분의 정확한 위치도 알 수 있었다. GIFT-64와 라운드별 차분 특성의 확률 및 활성 S-box 개수는 동일하였으며, 블록 크기가 커진 만큼 라운드가 증가할수록 탐색 시간이 점점 큰 차이를 보이는 것을 확인할 수 있었다.

Table 5. optimal probability of differential characteristic(GIFT-128)

Round	$-\log_2 p$	time	Active S-box	Active S-box[7]
1	1.415	0.12s	1	1
2	3.415	0.4s	2	2
3	7	7.67s	3	3
4	11.415	11m	6	5
5	17	3.5h	7	7

4.3 SKINNY-64의 차분 특성 탐색

SKINNY는 2016년 Christof Beierle 등에 의해 제안된 경량 블록 암호이다[10]. AES와 유사한 구조지만 Tweakable 블록 암호로 key외에 계속해서 변화하는 Tweak이 XOR되며 블록, 키, Tweak의 사이즈를 유동적으로 변형시킬 수 있다. SKINNY는 64, 128 두 종류의 블록 사이즈를 가지며 Tweak은 블록 사이즈에 따라 64, 128, 192 또는 128, 256, 384의 크기를 갖는다. 라운드는 SKINNY-64의 경우 Tweak의 크기에 따라 32, 36, 40라운드를 가지며, SKINNY-128의 경우 Tweak의 크기에 따라 40, 48, 56라운드를 갖는다. S-box는 SKINNY-64는 4bit S-box, SKINNY-128는 8bit S-box를 사용하며 SubCells, AddConstants, AddRound TweaKey, Shiftrows, Mixcolumns 함수로 구성된다. 전체 구조는 Fig. 6.과 같다.

본 논문에서 탐색한 SKINNY-64 S-box는 4bit의 4-uniform S-box로 입·출력 차분에 대해 가능한 확률은 2^{-2},2^{-3} 2가지가 있다.

입·출력 차분을 나타내는 변수 8개와 가능한 확률 2가지를 이용하여 제약식을 만들면 총 927개의 불가능한 조합이 있으며 이를 Espresso 알고리즘을 이용하여 최소화 하면 37개의 식으로 축소된다.

GIFT와 SKINNY 모델링의 차이점은 Shiftrows와 Mixcolumns를 포함한 제약식을 구성하는 것이다. Shiftrows는 단순히 출력 차분을 나타내는 변수의 위치를 바꾸어주면 되지만 Mixcolumns은 유한체 $GF(2^4)$상에서의 다항식 곱셈 연산을 이용하여 구현해야 한다. Mixcolumns을 구현하기 위해 S-box를 통과하여 Shiftrows까지 거친 입·출력 차분을 변수 X와 Y로, Mixcolumns을 거친 입·출력 차분을 Y와 X로 분할하였으며 Mixcolumns의 과정에서 값을 저장하기 위해 임시변수 Z를, XOR을 처리하기 위해 더미변수 d를 사용하였다. Mixcolumns 연산에 사용되는 행렬은 수식(1)과 같다. 이 때 연산은 수식(2)와 같은 형태로 진행되며 이를 전개하면 수식(3)과 같은 형태의 제약식으로 Mixcolumns을 표현할 수 있다.

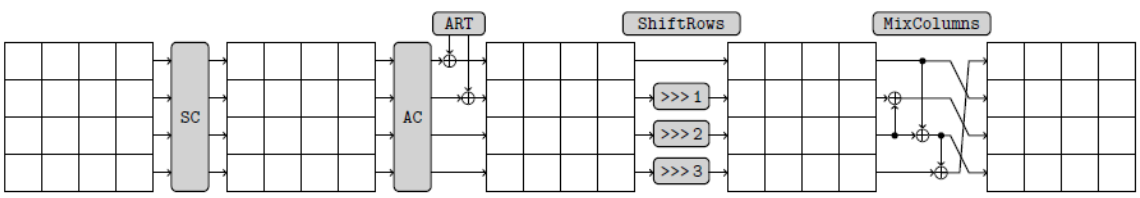

Fig. 6. SKINNY structure

$$M=\begin{pmatrix}1&0&1&1\\1&0&0&0\\0&1&1&0\\1&0&1&0\end{pmatrix} \quad (1)$$

$$M\times\begin{pmatrix}Y[r][0\sim15]\\Y[r][16\sim31]\\Y[r][32\sim47]\\Y[r][48\sim63]\end{pmatrix}=\begin{pmatrix}X[r+1][0\sim15]\\X[r+1][16\sim31]\\X[r+1][32\sim47]\\X[r+1][48\sim63]\end{pmatrix} \quad (2)$$

$$\begin{cases}Z[r][i]=Y[r][i]\oplus Y[r][i+32]\\X[r+1][i]=Z[r][i]\oplus Y[r][i+48]\\\qquad(0\le i<16)\\X[r+1][i]=Y[r][i-16](16\le i<32)\\X[r+1][i]=Y[r][i-16]\oplus Y[r][i]\\\qquad(32\le i<48)\\X[r+1][i]=Y[r][i-48]\oplus Y[r][i-16]\\\qquad(48\le i<64)\end{cases} \quad (3)$$

수식 (3)에 포함된 XOR 연산은 [11]에서 사용한 방식을 차용하여 더미변수 d를 통해 수식 (4)와 같이 1개의 식으로 표현하였다.

$$X\oplus Y=Z\rightarrow X+Y+Z-2d=0 \quad (4)$$

SKINNY-64에 대한 새로운 MILP 모델 적용 결과 총 6라운드에 대해 Table. 6.과 같은 결과를 확인할 수 있었으며 각 라운드별 최적의 결과로 탐색된 차분 특성에 대한 입력 차분 및 출력 차분의 정확한 위치도 알 수 있었다. SKINNY-64의 경우 탐색 결과 제안 논문과 비교하였을 때 활성 S-box 개수에 변화가 있는 특성을 찾아내지는 못하였으나, GIFT와 비교해본 결과 매 라운드를 지날수록 Mixcolumn을 사용한 SKINNY가 GIFT보다 많은 활성 S-box 개수를 가지며 차분 특성의 확률 또한 낮은 것을 알 수 있었다.

Table. 6. optimal probability of differential characteristic(SKINNY-64)

Round	$-\log_2 p$	time	Active S-box	Active S-box [10]
1	2	0.04s	1	1
2	4	0.2s	2	2
3	10	7.7s	5	5
4	16	3m	8	8

V. 결 론

본 논문에서에서 제안한 방법은 MILP Solver 프로그램을 이용한 차분 특성 탐색 방법으로 확률과 차분 특성에 대한 입·출력 변수들을 제약식에 포함시키고 Espresso 알고리즘을 적용하여 제약식을 간소화함으로써 탐색 절차의 간소화, 효율성 개선을 도모한 방식이다. 이를 GIFT-64, GIFT-128, SKINNY-64 등의 블록암호에 적용한 결과, 간편한 방법으로 이전 연구 결과 보다 더욱 정교한 탐색 결과를 제시할 수 있었다. 특히 GIFT에 대해서는 이전 연구 결과보다 많은 활성 S-box를 가지지만 더 높은 확률을 갖는 새로운 특성을 찾아냄으로써 보다 개선된 결과를 찾아내었다. 향후 해당 모델을 이용해 선형 관계식이나 불능 차분, 디비젼 성질 등 다양한 성질을 분석하는데 확률변수를 추가하는 방식을 적용할 수 있을 것으로 판단되며, Espresso 알고리즘의 경우 SAT Solver를 활용한 자동 탐색에도 적용해본다면 향후 유용한 결과를 도출 가능할 것으로 판단된다.

References

[1] Biham, Eli, and Adi Shamir. "Differential cryptanalysis of DES-like crypto systems." Journal of CRYPTOLOGY 4.1, 3-72. 1991.

[2] Mouha, Nicky, et al. "Differential and linear cryptanalysis using mixed-integer linear programming." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2011.

[3] Sun, Siwei, et al. "Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014.

[4] Fu, Kai, et al. "Milp-based automatic search algorithms for differential and

linear trails for speck." International Conference on Fast Software Encryption. Springer, Berlin, Heidelberg, 2016.

〔5〕 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, 2017.

〔6〕 Xiang, Zejun, et al. "Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2016.

〔7〕 Banik, Subhadeep, et al. "GIFT: a small PRESENT." International Conference on Cryptographic Hardware and Embedded Systems. Springer, Cham, 2017.

〔8〕 Sun, Siwei, et al. "Analysis of AES, SKINNY, and others with constraint programming." IACR Transactions on Symmetric Cryptology, 2017.

〔9〕 Sun, Ling, Wei Wang, and Meiqin Wang. "Automatic Search of Bit-Based Division Property for ARX Ciphers and Word-Based Division Property." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017.

〔10〕 Beierle, Christof, et al. "The SKINNY family of block ciphers and its low-latency variant MANTIS." Annual-Cryptology Conference. Springer, Berlin, Heidelberg, 2016.

〔11〕 HoChang Lee, HyungChul Kang, Deukjo Hong, Jaechul Sung, Seokhie Hong, "Searching for Impossible Differential Characteristics of ARX-Based Block Cipher Using MILP" Journal of KIISC,VOL.27, NO.1, Feb. 2017.

〔12〕 Brian Lawless, "Fundamental digital electronics, Unit 17 Espresso minimization algorithm", http://www.physics.dcu.ie/~bl/digi/unitd17.pdf

〔13〕 Abdelkhalek, Ahmed, et al. "MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics." IACR Transactions on Symmetric Cryptology 2017.4, 99-129. 2017.

〈저자소개〉

박 연 지 (YeonJi Park) 학생회원
2011년 2월: 육군사관학교 영어과 졸업
2017년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호 알고리즘 설계 및 분석, 자동탐색

이 호 창 (HoChang Lee) 정회원
2016년 2월: 서울시립대학교 수학과 졸업
2018년 2월: 고려대학교 정보보호대학원 석사
2017년 12월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 암호 알고리즘 설계 및 분석, 자동탐색

홍 득 조 (Deukjo Hong) 종신회원
1999년 8월: 고려대학교 수학과 학사
2001년 8월: 고려대학교 수학과 석사
2006년 2월: 고려대학교 정보보호대학원 박사
2007년 12월~2015년 8월: 국가보안기술연구소 선임연구원
2015년 9월~현재: 전북대학교 IT정보공학과 조교수
〈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 박사 후 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