Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.6, Dec. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.6.1259

BOGI 전략으로 설계된 블록 암호의 차분 공격에 대한 안전성 분석*

이 상 협,[1†] 김 성 겸,[1] 홍 득 조,[2‡] 성 재 철,[3] 홍 석 희[1]
[1]고려대학교, [2]전북대학교, [3]서울시립대학교

Security Analysis of Block Ciphers Designed with BOGI Strategy against Differential Attacks*

Sanghyeop Lee,[1†] Seonggyeom Kim,[1] Deukjo Hong,[2‡] Jaechul Sung,[2] Seokhie Hong[1]
[1]Korea University, [2]Chonbuk National University, [3]University of Seoul

요 약

블록 암호를 설계할 때, 설계자는 주로 차분 특성 확률의 상한을 이용하여 라운드 수를 결정한다. 라운드 수는 블록 암호의 성능에 영향을 미치므로, 더 적은 라운드를 갖기 위해 차분 특성 확률의 상한을 정밀하게 계산하는 것이 중요하다. 이전까지의 활성 S-box의 최소 개수를 탐색하는 방법들은 비선형 연산과 선형 연산을 각각 제약식으로 구성하여 차분 특성 확률의 상한을 계산하였다. 하지만 선형 연산이 비선형 연산에 의존적으로 선택되는 BOGI 설계전략(Bad-Output Good-Input Design Strategy)의 경우 이전 탐색방법으로 구한 상한은 정밀하지 않을 수 있다. 본 논문에서는 BOGI 전략의 성질을 이용하여 기존의 방법보다 더 정밀한 차분 특성 확률의 상한을 구하는 새로운 방법을 제안한다. 그리고 이 방법을 이용하여 구한 상한의 타당성을 수학적으로 증명한다. 제안한 방법을 BOGI가 사용된 GIFT-64와 GIFT-128에 각각 적용하여 9라운드까지 차분 특성 확률의 상한을 탐색하였다. GIFT-64의 7라운드와 GIFT-128의 9라운드에 대해 기존의 방법을 적용하면 차분 특성 확률의 상한이 각각 $2^{-18.395}$와 $2^{-26.885}$이었으나, 제안한 방법을 적용하면 각각 $2^{-19.81}$과 $2^{-28.3}$으로 더 정밀하게 계산된다.

ABSTRACT

The upper bound of differential characteristic probability is mainly used to determine the number of rounds when constructing a block cipher. As the number of rounds affects the performance of block cipher, it is critical to evaluate the tight upper bound in the constructing process. In order to calculate the upper bound of differential characteristic probability, the previous searching methods for minimum number of active S-boxes constructed constraint equations for non-linear operations and linear operations, independently. However, in the case of BOGI design strategy, where linear operation is dependent on non-linear operation, the previous methods may present the less tight upper bound. In this paper, we exploit the properties of BOGI strategy to propose a new method to evaluate a tighter upper bound of differential characteristic probability than previous ones. Additionally, we mathematically proved the validity of our method. Our proposed method was applied to GIFT-64 and GIFT-128, which are based on BOGI strategy, and the upper bounds of differential characteristic probability were derived until 9 round. Previously, the upper bounds of differential characteristic probability for 7-round GIFT-64 and 9-round GIFT-128 were $2^{-18.395}$ and $2^{-26.885}$, respectively, while we show that the upper bounds of differential characteristic probability are more tight as $2^{-19.81}$ and $2^{-28.3}$, respectively.

Keywords: BOGI, Differential Cryptanalysis, Active S-box, GIFT, MILP

Received(10. 08. 2019), Modified(11. 08, 2019), Accepted(11. 10. 2019)

* 본 연구는 고려대 암호기술 특화연구센터(UD170109ED)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.

† 주저자, tkekd91@korea.ac.kr
‡ 교신저자, deukjo.hong@jbnu.ac.kr(Corresponding author)

I. 서 론

차분 분석(Differential Cryptanalysis)은 현재까지 블록 암호의 안전성을 분석하는데 널리 사용되는 분석법이다[1]. 차분 분석을 통해 블록 암호의 안전성을 분석하는 방법은 크게 2가지가 있다. 첫 번째 방법은 활성 S-box의 최소 개수를 구하여 차분 특성(differential characteristic) 확률의 상한을 구하는 방법이다. 활성 S-box는 입력 차분이 0이 아닌 S-box를 말한다. 두 번째 방법은 차분 특성의 최대 확률을 구하는 방법이다. 일부 암호들의 경우에는 가장 큰 실제 차분 특성의 확률을 구하는 것이 어렵다. 따라서 활성 S-box의 최소 개수를 구하여 S-box의 최대 차분 확률(MDP, Maximum Differential Probability)을 곱하는 것으로 차분 특성 확률의 상한을 계산한다.

안전한 암호를 설계하기 위해서는 알려진 공격들에 대한 안전성을 보이는 것이 중요하다. 그러므로 암호 설계자들은 차분 분석에 안전성을 보이기 위해 주어진 시간 안에 실제 차분 특성의 확률을 구하는 것이 어려운 경우에는 차분 특성 확률의 상한을 계산하는 것으로 안전성을 보인다. 이 방법을 이용하여 안전성을 보인 블록 암호로는 AES, PRESENT, CLEFIA 등이 있다[2,3,4]. 예를 들어 CLEFIA의 경우 12-라운드 활성 S-box의 최소 개수는 28개이고, 여러 가지의 S-box 중 가장 큰 확률을 S-box 개수에 곱하여 차분 특성 확률의 상한을 계산하였다. 차분 특성의 최대 확률 또는 차분 특성 확률의 상한을 구하기 위해서 수학적 문제를 해결하는 자동화 도구를 이용한다.

수학적 문제를 해결하기 위한 자동화 도구들은 암호의 설계 또는 분석에 유용하게 사용되고 있다. 수학적 도구를 암호분석에 이용하기 위해서는 암호의 비선형 요소와 선형 요소를 각각 제약식으로 구성하여 최대 확률을 갖는 차분 특성을 탐색하거나 활성 S-box의 최소 개수를 탐색한다. 수학적 문제 중 MILP(Mixed Integer Linear Programming) 문제는 선형부등식의 변수 일부가 정수인 제약식을 만족하는 값 중 목적함수의 최댓값 또는 최솟값을 구하는 문제이다. 이를 이용하여 Mouha 등은 활성 S-box의 최소 개수를 구하는 문제를 MILP Solver로 해결하였다[5]. Mouha의 방법은 워드(word) 단위로 연산 되는 암호에 적용할 수 있도록 제약식을 구성하여 탐색하는 방법이다. 하지만 이 방법은 비트 순열(bit-permutation)과 같은 비트(bit) 단위 연산을 사용하는 암호에 적용하기 힘든 한계점이 있다. 이후 Sun 등은 이 방법을 확장하여 비트 단위 연산을 사용하는 암호에도 적용할 수 있도록 제약식을 구성하여 활성 S-box의 최소 개수를 탐색하였다[6,7]. 또한, MILP를 이용한 차분 특성의 최대 확률을 구하는 방법을 제안하였다[8].

현재까지 알려진 차분 특성을 탐색하는 방법은 비선형 연산과 선형 연산을 각각 제약식으로 만들어 분석한다. 그러나 2017년 Subhadeep Banik 등에 의해 제안된 경량 블록 암호인 GIFT[9]는 BOGI 설계전략(Bad-Output Good-Input Design Strategy)을 적용하여, 차분 공격에 대한 안전성이 낮은 비선형 연산을 선형 연산을 통해 보완하는 방법으로 설계되었다. 이 설계기법은 GIFT에서 처음 제안된 설계 원리로 S-box의 출력 차분의 활성 비트 수가 1일 경우(Bad-Output, BO) 비트 순열을 적용한 다음 S-box의 출력 차분의 활성 비트 수가 1이 나올 수 없는 입력 차분(Good-Input, GI)을 만드는 설계기법이다. 이를 통하여 GIFT는 PRESENT의 S-box보다 차분 분석의 관점에서 안전하지 않은 경량 S-box를 사용하지만, 차분 분석에 더 우수한 안전성을 갖는다. 예를 들어 GIFT와 PRESENT의 9라운드 차분 경로 확률을 비교하였을 때, GIFT는 $2^{-44.415}$이고 PRESENT는 $2^{-40.702}$로 GIFT가 더 낮은 확률로 차분 관점에서 우수한 안전성을 갖는 것을 확인했다. GIFT의 비선형 연산에 대한 제약식과 선형 연산에 대한 제약식을 각각 구성하면 정밀한 상한을 구할 수 없다. 왜냐하면, 기존 방법으로 탐색한 활성 S-box들 중 BO를 갖는 낮은 확률의 활성 S-box가 포함될 수 있기 때문이다.

본 논문은 BOGI를 적용한 암호의 차분 특성 확률의 상한을 계산할 때, 비선형 연산과 선형 연산을 모두 고려한 제약식을 통해 더 정밀한 상한을 계산하는 방법을 제안한다. 제안하는 방법은 BOGI의 1비트 입력 차분에서 1비트 출력 차분이 나온 후에 다시 1비트 입력 차분이 나오지 않는 성질을 고려한다. 이 방법으로 기존 상한보다 더 정밀한 차분 특성 확률의 상한을 구할 수 있다. 예를 들어, BOGI가 적용된 블록 암호를 기반으로 한 Sponge[10] 구조를 사용하는 인증 암호화를 설계할 때와 같이 블록의 크기가 큰 경우 차분 특성의 최대 확률을 가용시간 내에 구하기 어려워 제안하는 방법으로 구한 정밀한 상한이 중요할 수 있다. 또한, 제안하는 방법으로 구한

차분 특성 확률의 상한이 타당한지 수학적으로 증명한다. BOGI를 적용한 블록 암호인 GIFT를 대상으로 기존 활성 S-box의 최소 개수를 탐색하는 제약식에 낮은 확률을 갖는 BOGI S-box를 구분할 수 있도록 제약식을 추가하고, MILP Solver를 이용하여 더 정밀한 차분 특성 확률의 상한을 계산하고 기존 방법의 결과와 비교한다. GIFT-64의 7라운드와 GIFT-128의 9라운드에 대해 기존의 방법을 적용하면 차분 특성 확률의 상한이 각각 $2^{-18.395}$와 $2^{-26.885}$이었으나, 제안한 방법을 이용하면 각각 $2^{-19.81}$과 $2^{-28.3}$으로 더 정밀하게 계산된다. BOGI를 적용한 블록 암호는 본 논문에서 제안한 방법으로 차분 특성 확률의 상한을 탐색하면 기존 방법보다 정밀한 상한을 계산할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 차분 분석, 부울 함수의 논리식 표현, GIFT에 관해 설명하고 BOGI에 대해 설명한다. 3장에서는 관련 연구로 MILP를 이용하여 차분 특성을 탐색하는 방법을 설명한다. 4장에서는 본 논문에서 제안하는 BOGI를 고려한 새로운 차분 특성 확률의 상한 탐색방법을 설명하고 논문에서 제안한 상한의 타당성을 보인다. 5장에서는 제안한 방법을 GIFT에 적용하여 얻은 결과를 소개한다. 마지막으로 6장에서 결론을 맺는다.

II. 배경 지식

2.1 차분 분석

차분 분석은 입력 차분 값에 따른 출력 차분 값의 확률 분포를 이용한다. 선형 구조에서는 차분의 변화를 1의 확률로 알 수 있지만, 비선형 구조는 특정 확률로 차분의 변화를 알 수 있다.

비선형 구조의 대표적인 예로는 S-box가 있다. S-box는 특정 값을 입력받아 대응되는 특정 값을 출력하는 비선형 함수이다. n비트가 입력되어 n비트가 출력되는 S-box에서 모든 입·출력 차분 값이 발생할 확률이 균일하다면 특정 입력 차분 값에 대하여 특정 출력 차분 값이 나올 확률은 $1/2^n$로 같아야 한다. 하지만 이러한 분포는 S-box에서 존재할 수 없다. 이런 성질을 이용하여 차분 분석이 진행되며 분석 대상의 모든 입력 차분 값과 가능한 모든 출력 차분 값을 표로 만들어 사용한다. 가능한 입·출력 차분의 분포를 표로 나타낸 것을 차분 분포표(Difference Distribution Table, DDT)라 정의한다. n비트를 입력하여 m비트를 출력하는 즉, $f: F_2^n \rightarrow F_2^m$인 S-box f가 존재하면 차분 분포표 $D(f)$는 $2^n \times 2^m$인 행렬로 정의한다.

$$D(f)[\delta,\Delta] = \#\left\{x \in F_2^n, f(x \oplus \delta) \oplus f(x) = \Delta\right\}$$

행렬의 원솟값은 입력 차분이 δ인 입력값 $x, x \oplus \delta$를 각각 S-box에 대입한 출력값 $f(x), f(x \oplus \delta)$의 차분이 Δ를 만족하는 쌍의 개수이다.

Nyberg는 함수 $f: F_2^n \rightarrow F_2^n$에서 입력 차분과 출력 차분에 대하여 입력 차분이 0이 아닌 값 중 원소의 값이 가장 큰 값을 차분 균일성(differential uniformity)이라 정의하였다. 또한, 행렬 $D(f)$ 차분 균일성을 Δ_f-uniform이라 표기한다[11].

$$\Delta_f = \max\left\{D(f)[\delta,\Delta] : \delta \in F_{2^n}^*, \Delta \in F_{2^n}\right\}$$

2.2 부울 함수의 논리식 표현

부울(boolean) 함수의 논리식은 두 가지 방법으로 표현할 수 있다. 첫 번째 방법은 논리곱(and)들의 논리합(or)으로 표현되는 DNF(Disjunctive Normal Form)이고 두 번째 방법은 논리합들의 논리곱으로 표현되는 CNF(Conjunctive Normal Form)이다. 곱의 합 형태(DNF)는 참(true)의 결과를 갖는 변수들의 논리곱을 각각의 절(clause)로 만들어 각 절을 논리합으로 연결한 것이며, 합의 곱 형태(CNF)는 거짓(false) 결과를 갖는 변수들의 부정 값을 논리합으로 연결하여 각각의 절로 만들고 각 절을 논리곱으로 연결한 것이다.

예를 들어, x_0에 어떠한 연산을 하여 x_1이 되는지에 대한 결괏값 True/False(1/0)를 y라고 하면

Table 1. Truth table example

x_0	x_1	y	*DNF*	*CNF*
0	0	1	$(\overline{x_0} \wedge \overline{x_1})$	
0	1	0		$(x_0 \vee \overline{x_1})$
1	0	0		$\wedge(\overline{x_0} \vee x_1)$
1	1	1	$\vee(x_0 \wedge x_1)$	

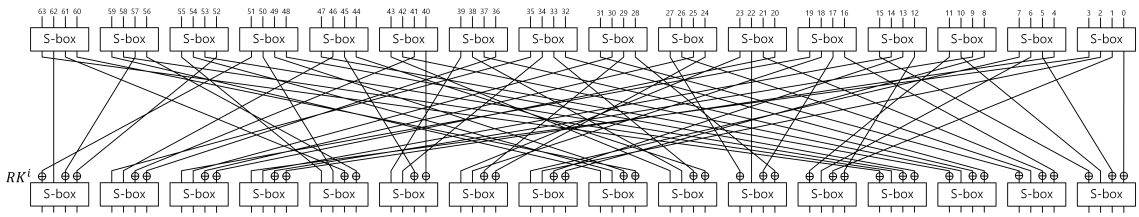

Fig. 1. One round of GIFT-64

Table 1.과 같이 나타낼 수 있다. DNF는 y값이 1인 변수들로 표현할 수 있는데, 0인 값에는 부정(not) 연산자를 붙여 논리곱으로 묶고 이렇게 묶인 절끼리는 논리합으로 연결한다. 따라서 Table 1.의 DNF는 $(\overline{x_0} \wedge \overline{x_1}) \vee (x_0 \wedge x_1)$이 된다. 비슷한 방법으로 CNF는 y값이 0인 변수들로 표현할 수 있는데, 1인 값에 부정 연산자를 붙여 논리합으로 묶고 이렇게 묶인 절끼리 논리곱으로 연결하여 $(x_0 \vee \overline{x_1}) \wedge (\overline{x_0} \vee x_1)$으로 표현할 수 있다.

2.3 GIFT

2017년에 제안된 GIFT는 2007년에 제안된 PRESENT[3]와 비슷한 SPN(Substitution Permutation Network) 구조이다. GIFT는 블록 크기에 따라 GIFT-64와 GIFT-128로 나눌 수 있고, 각 라운드 수는 28라운드, 40라운드이다. 키(key)는 모두 128비트를 사용한다.

GIFT-64	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$P_{64}(i)$	0	17	34	51	48	1	18	35	32	49	2	19	16	33	50	3
	i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	$P_{64}(i)$	4	21	38	55	52	5	22	39	36	53	6	23	20	37	54	7
	i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	$P_{64}(i)$	8	25	42	59	56	9	26	43	40	57	10	27	24	41	58	11
	i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	$P_{64}(i)$	12	29	46	63	60	13	30	47	44	61	41	31	28	45	62	15
GIFT-128	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$P_{128}(i)$	0	33	66	99	96	1	34	67	64	97	2	35	32	65	98	3
	i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	$P_{128}(i)$	4	37	70	103	100	5	38	71	68	101	6	39	36	69	102	7
	i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	$P_{128}(i)$	8	41	74	107	104	9	42	75	72	105	10	43	40	73	106	11
	i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	$P_{128}(i)$	12	45	78	111	108	13	46	79	76	109	14	47	44	77	110	15
	i	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	$P_{128}(i)$	16	49	82	115	112	17	50	83	80	113	18	51	48	81	114	19
	i	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
	$P_{128}(i)$	20	53	86	119	116	21	54	87	84	117	22	55	52	85	118	23
	i	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
	$P_{128}(i)$	24	57	90	123	120	25	58	91	88	121	26	59	56	89	122	27
	i	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
	$P_{128}(i)$	28	61	94	127	124	29	62	95	92	125	30	63	60	93	126	31

Fig. 2. Specification of GIFT Bit Permutation

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. 3. S-box of GIFT

전체적인 구조는 Fig 1. 과 같고, 각 라운드는 키를 XOR 연산하는 AddRoundKey, Fig. 2.의 GIFT S-box를 사용하여 4비트 단위의 비선형 연산을 하는 SubCell, 비트를 Fig. 3.에 따라 섞어주는 Permubits로 총 3단계로 구성된다. GIFT-64는 16개의 워드로 구성되고 GIFT-128은 32개의 워드로 구성된다.

2.4 BOGI

BOGI는 GIFT에서 처음 소개된 설계기법으로 S-box의 출력 차분의 활성 비트 수가 1개일 경우(Bad-Output) 비트 순열을 적용한 다음 S-box에 출력 차분의 활성 비트 수가 2개 이상이 나오는 입력 차분(Good-Input)을 만드는 방법이다.

S-box의 DDT를 생성한 후 입력과 출력 차분의 해밍 웨이트가 1인 차분으로 구성된 1-1비트 DDT를 생성하면 Table 2.와 같이 나타낼 수 있다. GI, GO, BI, BO는 각각 Good Input, Good Output, Bad input, Bad Output을 의미한다. 1-1비트 DDT에서 GI는 행을 기준으로 원소가 모두 0인 것을 말하고 반대로 BI는 적어도 하나는 0이 아닌 것을 의미한다. 마찬가지로 GO와 BO는 열을 기준으로 위와 같은 조건을 적용한 것이다. 예를 들어, Table 2.에서 GI={0100,0010}, GO={1000,0001}, BI={1000,0001}, BO={0100,0010}임을 알 수 있다.

BOGI는 $|BO| \leq |GI|$ 를 만족하는 S-box와 BO가 GI가 되는 순열을 갖는 구조를 만드는 것을 의미한다. BOGI를 이용하면 1비트의 출력 차분이 나오면 다음 라운드의 출력 차분은 2비트 이상임을 보장할 수 있다. 따라서 BOGI를 이용하면 S-box의 브랜치 넘버가 2임에도 1비트에서 1비트로 계속 이어지는 차분 특성을 만들 수 없게 된다.

또한, BOGI에 적합한 S-box는 [9]에서 제안한 5가지 조건 중 2와 3을 만족하는 S-box이고 1-1비

Table 2. 1-1 bit DDT Example

Δx \ Δy	1000	0100	0010	0001
1000	0	2	2	0
0100	0	0	0	0
0010	0	0	0	0
0001	0	2	2	0

트 DDT의 원소는 어떠한 확률이든 가질 수 있다. 다만 GIFT의 경우 조건 4와 5가 추가로 적용하였기 때문에, 1-1비트 DDT의 원소는 낮은 확률을 갖는다.

III. MILP를 이용한 차분 특성 탐색방법

MILP Solver는 주어진 식의 해를 찾아주는 SAT/ SMT Solver와 달리, 사용자가 설정한 식의 최댓값 또는 최솟값을 찾는다. 또한, DDT의 모든 원소가 거듭제곱 꼴이 아니라면 SAT/SMT는 확률 계산을 위한 소수점 연산을 구현해야 하지만 MILP Solver는 연산을 지원하기 때문에 다양한 암호에 간편하게 적용할 수 있다. MILP Solver로 차분 특성을 탐색하기 위해선 제약식을 통해 가능한 조합을 탐색하도록 암호를 모델링(modeling)하는 작업이 필요하다. 모델링 작업은 최소화 또는 최대화하고 싶은 변수들로 구성된 목적함수 설정, 비선형 연산과 선형 연산의 차분 특성을 부등식으로 표현한 제약식과 변수를 선언하는 것을 말한다. 본 장에서는 활성 S-box의 최소 개수를 탐색하는 두 가지 모델링 방법, 차분 특성의 최대 확률을 탐색하는 모델링 방법에 관해 설명한다.

3.1 활성 S-box의 최소 개수 탐색방법

비트 단위 연산을 고려한 활성 S-box의 최소 개수를 구하기 위해 암호의 구성 요소를 부등식으로 표현해야 한다. 암호의 특성을 부등식으로 표현하기 전에 차분 벡터를 아래와 같이 정의한다.

정의 1. n개의 비트로 이루어진 차분을 $\Delta = (\Delta_0, \Delta_1, \cdots, \Delta_{n-1})$으로 표현하면 Δ의 차분 벡터 $x = (x_0, x_1, \cdots, x_{n-1})$를 식(1)과 같이 정의할 수 있다.

$$x_i = \begin{cases} 0 & \text{if } \Delta_i = 0, \\ 1 & \text{if } \Delta_i = 1. \end{cases} \tag{1}$$

차분 벡터를 비트 단위로 정의하였으므로 식(2)와 같이 활성 S-box를 구분하기 위한 제약식이 필요하다. 활성 S-box를 나타내기 위해서 n-비트의 S-box에서 입력 차분 벡터 x의 원소 중 하나라도 1이 존재하면 S-box가 활성($A=1$)이 되는 제약식이다. 따라서 식(2)를 워드의 개수만큼 사용하고 그 값을 모두 더하면 한 라운드의 활성 S-box의 개수를 계산할 수 있다. 목적함수는 모든 A를 더한 값이 최솟값이 되도록 작성한다.

$$\begin{cases} A - x_i \geq 0, i \in \{0, \ldots, n-1\} \\ \sum_{i=0}^{n-1} x_i - A \geq 0 \end{cases} \tag{2}$$

S-box의 차분 특성을 제약식으로 구성하는 방법은 두 가지로 나눌 수 있다. 첫 번째 방법은 식(3)과 같이 S-box의 브랜치 넘버(branch number, B_S)를 사용하는 방법이다[6]. S-box의 브랜치 넘버란 0이 아닌 입력 차분(x)의 해밍 웨이트와 출력 차분(y)의 해밍 웨이트의 합의 최솟값을 말하고 $B_S = \min_{x \neq y}\{wt((x \oplus y) \| (S(x) \oplus S(y)) : x, y \in F_2^n\}$로 정의한다. d^S는 더미(dummy) 변수로 $\{0,1\}$의 값을 갖는다.

$$\begin{cases} \sum_{i=0}^{n-1} x_i^S + \sum_{i=0}^{n-1} y_i^S - B_S d^S \geq 0 \\ d^S - x_i^S \geq 0,\ d^S - y_i^S \geq 0,\ i \in \{0, \ldots, n-1\} \end{cases} \tag{3}$$

두 번째 방법은 MILP Solver가 S-box의 DDT에서 불가능한 입·출력 차분을 구분할 수 있도록 제약식을 구성하는 방법이다[7]. 먼저, S-box의 제약식을 구성하기 위해서는 앞서 2.1장에서 설명한 DDT를 생성한다. 생성된 DDT는 가능한 결괏값(참, ≥ 2)이 되는 입·출력 차분과 불가능한 결괏값(거짓, $=0$)이 되는 입·출력 차분으로 구분할 수 있다. S-box의 DDT에서 불가능한 입·출력 차분을 MILP에서 사용할 수 있도록 부등식으로 만들기 위해서 변수들을 CNF로 표현한다. n-비트 입력과 n-비트 출력을 갖는 S-box에서 거짓이 되는 입력과 출력을 변수로 사용하면 아래와 같은 식으로 DDT

의 불가능한 입·출력 차분을 CNF로 표현할 수 있다.

$$f(in,out) = \bigwedge_{c\in\{0,1\}^{2n}} \left(\left(\bigvee_{i=1}^{n} in_i \oplus c_i \right) \vee \left(\bigvee_{i=1}^{n} out_i \oplus c_{n+i} \right) \right)$$

c는 DDT의 결괏값이 0인 불가능한 입력과 출력을 나타낸다. 논리합으로 구성된 절은 논리곱으로 연결되어 있으므로 위의 식이 참이 되는 경우는 각각의 절이 모두 참인 경우를 말한다. 따라서 S-box에 대한 제약식이 참이 되는 것을 부등식으로 나타내면 $f(input,output) \geq 1$이고 이 식을 만족하는 입력과 출력은 DDT에서 참이 되는 값이다.

선형 연산의 경우 두 방법 모두 비트 단위 연산을 사용할 수 있고 선형 연산의 종류는 크게 비트 순열과 XOR 연산으로 나눌 수 있다. 비트 순열과 같은 연산은 변수의 순서를 변경하여 연산하지만, XOR 연산은 입력(x_{in1}, x_{in2})과 출력(x_{out})에 대해 다음과 같은 제약식을 사용한다. d는 더미 변수로 $\{0,1\}$의 값을 갖는다.

$$\begin{cases} x_{\mathrm{in1}} + x_{\mathrm{in2}} + x_{out} \geq 2d \\ d - x_{\mathrm{in1}} \geq 0,\ d - x_{\mathrm{in2}} \geq 0,\ d - x_{out} \geq 0 \\ x_{in1} + x_{in2} + x_{out} \leq 2 \end{cases}$$

[12]에 따르면 일반적으로 논리식을 간소화하는 것이 탐색 시간을 줄여주지만, 논리식의 개수를 최소화한다고 탐색 시간이 비례하여 줄어드는 것이 아님을 알 수 있다. 그러나 일반적으로 논리식을 간소화하는 것이 시간을 줄여주기 때문에, 본 논문에서는 논리식을 간소화하는 알고리즘 중 Espresso[13]를 사용한다.

3.2 차분 특성의 최대 확률 탐색방법

MILP Solver를 이용하면 차분 특성의 최대 확률을 구하는 것 또한 가능하다. 차분 특성의 최대 확률을 구하기 위해서는 활성 S-box가 DDT의 어떤 확률을 갖는지 구분하기 위한 변수가 추가로 필요하다. DDT의 확률의 종류에 따라 변수의 개수가 정해지고, 제약식의 개수도 정해지게 된다. 예를 들어, 3비트 S-box의 DDT가 3가지의 확률을 갖는다고 하면 각각의 확률 변수를 p^1, p^2, p^3로 정의한 후 아래와 같은 벡터로 입·출력 차분과 확률을 표현할 수 있다.

$$(x_0, x_1, x_2, y_0, y_1, y_2, p^1, p^2, p^3)$$

앞서 설명한 3.1의 두 번째 방법과 같이 입·출력 차분이 불가능한 확률을 갖는 경우를 모두 제거하도록 S-box의 제약식을 구성하고 선형 연산의 경우 앞서 설명한 방법과 같이 구성한다.

IV. BOGI를 고려한 차분 특성 확률의 상한

앞서 3.1절에서 언급한 Sun의 방법[7]을 이용하면 비트 순열을 사용하는 암호까지 다양한 암호의 활성 S-box의 최소 개수를 구할 수 있다. 일반적으로 이렇게 구한 활성 S-box의 최소 개수에 DDT의 최대 차분 확률을 곱하여 차분 특성 확률의 상한을 계산한다. 그러나 BOGI 가 적용된 GIFT와 같은 암호의 활성 S-box의 최소 개수에 최대 차분 확률을 곱하여 차분 특성의 상한을 계산하는 것은 정밀하지 않을 수 있다. 왜냐하면, 활성 S-box들 중에 구분이 가능한 낮은 확률을 갖는 활성 S-box를 포함할 수 있기 때문이다. BOGI가 적용되지 않은 기존 암호의 활성 S-box 중에도 최대 확률을 갖지 않는 S-box가 존재할 수 있지만 이를 구분하는 것은 확률을 계산하는 것과 복잡도가 비슷하다. 하지만 BOGI가 적용된 암호에서 1비트 차분에서 1비트 차분으로 가는 경우 낮은 확률을 가질 수 있으므로 이러한 활성 S-box를 구분하여 탐색을 진행할 수 있다면 개선된 차분 특성의 상한을 구할 수 있다.

4.1 BOGI를 고려한 목적함수 및 S-box 제약식

GIFT의 1-1 비트 DDT는 Table 3.과 같고 입력 차분이 0001이고 출력 차분이 1000만 존재한다. 이러한 입력 차분과 출력 차분을 갖는 활성 S-box를 BOGI S-box라고 정의한다. 앞서 3장에서 설명한 Sun의 방법에서 BOGI S-box에 해당하는 제약식을 추가하면 BOGI S-box를 구별할 수 있다. 따라서 DDT에서 입력 차분이 1이고 출력 차분이 8인 경우 1이고 나머지 가능한 입·출력 차분에서는 0인 결괏값을 갖는 변수를 추가하여 전체 S-box의 제약식을 생성한다. GIFT는 4비트 S-box를 사용하여

Table 3. 1-1 bit DDT of GIFT S-box

Δx \ Δy	1000	0100	0010	0001
1000	0	0	0	0
0100	0	0	0	0
0010	0	0	0	0
0001	2	0	0	0

입·출력에 해당하는 변수 8개와 BOGI S-box를 나타내는 변수 1개를 사용하여 식(4)과 같은 형태의 벡터로 표현할 수 있다.

$$v=(x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3,B) \tag{4}$$

식 (4)의 벡터를 2.2절의 설명과 같이 부울 함수를 논리식으로 표현하여 BOGI를 고려한 GIFT S-box의 논리식을 생성한다. 예를 들어, 1에서 8로 가는 경우 변수 B의 값이 1인 경우 참이고 0이면 거짓이 된다. 이것을 CNF로 나타내면 식 (5)와 같다.

$$x_0\vee x_1\vee x_2\vee \overline{x_3}\vee \overline{y_0}\vee y_1\vee y_2\vee y_3\vee B \tag{5}$$

이와 같은 방법으로 DDT에서 불가능한 모든 조합을 논리곱으로 연결하면 부울 함수 f를 식(6)과 같이 표현할 수 있다. 8개의 변수로 나타낼 수 있는 DDT의 모든 불가능한 조합은 157개의 절로 존재하고 Espresso 알고리즘을 통하여 제약식을 간소화하면 34개의 절이 존재하고 식 (7)와 같다.

$$\begin{aligned} f(x,y)=&\\ &(x_0\vee x_1\vee x_2\vee \overline{x_3}\vee y_0\vee \overline{y_1}\vee y_2\vee \overline{y_3})\wedge\\ &\cdots\wedge(x_0\vee x_1\vee x_2\vee \overline{x_3}\vee y_0\vee \overline{y_1}\vee y_2\vee y_3)\wedge\\ &\cdots\wedge(\overline{x_0}\vee \overline{x_1}\vee \overline{x_2}\vee \overline{x_3}\vee \overline{y_0}\vee \overline{y_1}\vee y_2\vee \overline{y_3}) \end{aligned} \tag{6}$$

$$\begin{aligned} f(x,y)=&(\overline{x_0}\vee \overline{x_2}\vee \overline{x_3}\vee \overline{y_0}\vee y_1\vee \overline{y_0}\vee y_3)\wedge\\ &\cdots\wedge(x_0\vee x_1\vee y_0\vee \overline{y_2}\vee \overline{y_3})\wedge\\ &\cdots\wedge(\overline{x_0}\vee \overline{x_1}\vee \overline{x_2}\vee x_3\vee \overline{y_2}) \end{aligned} \tag{7}$$

CNF로 나타낸 부울 함수는 MILP Solver에서 사용할 수 있도록 부등식의 형태로 변경해야 한다. 부정 연산자가 붙은 변수는 $1-Var$와 같이 표현하고 부정 연산자가 붙지 않은 변수는 Var로 나타낸다. 부울 함수 f는 절들이 논리곱으로 연결되어 있어 각 절이 모두 참이면 f가 참이 되고 참이 되는 부등식은 ≥ 1로 표현한다. 따라서 각 절이 모두 1 이상이 되어야 한다. 식 (5)를 MILP Solver에서 사용할 수 있도록 좌변에는 변수가 존재하고 우변에는 정수만 존재하도록 부등식으로 변환하면 식(8)과 같다.

$$x_0+x_1+x_2-x_3-y_0+y_1+y_2+y_3+B\geq -1 \tag{8}$$

S-box 차분 확률의 종류가 t개이고 해당하는 로그$(-\log_2)$ 값을 $p_1,p_2,p_3,\ldots,p_n(p_1<p_2<\cdots<p_t)$이라 하자. 활성 S-box의 변수에는 p_1을 가중치로 주고 BOGI S-box가 갖는 차분 확률의 로그 값을 p_b $(p_b\in\{p_2,p_3,\ldots,p_t\})$라고 하면 BOGI S-box의 변수에는 p_b-p_1을 가중치로 주어 차분 특성 확률의 상한을 최소화하는 목적함수를 식(9)과 같이 작성하였다. Sun 방법의 활성 S-box를 나타내는 변수 A와 BOGI S-box를 나타내는 변수 B에 각각의 가중치를 두어 최소가 되도록 하였다. (r: 라운드, w:워드 개수)

$$\begin{aligned} object=&p_1A[0][0]+(p_b-p_1)B[0][0]+\\ &\cdots+p_1A[r-1][w-1]+(p_b-p_1)B[r-1][w-1] \end{aligned} \tag{9}$$

GIFT의 경우 한 라운드의 한 워드에 대한 제약식의 개수는 Table. 4.와 같다. GIFT의 1-1 비트 DDT에서 구별해야 할 입·출력 차분이 1개이므로 제약식은 1개가 추가된다. 일반적으로는 1-1 비트 DDT의 0이 아닌 원소의 개수에 따라 제약식이 추가된다.

Table 4. Constraints of word

	3.1[7]	3.2	This
Constraints	39	48	40

4.2 제안하는 방법으로 구한 상한의 타당성

본 절에서는 4.1절에서 구한 차분 특성 확률의 상한이 실제 차분 특성의 최대 확률 보다 크거나 같고

기존 방법으로 구한 상한보다 작거나 같은 것을 정리 1과 같이 나타냈다.

정리 1. BOGI S-box를 고려하여 구한 확률은 활성 S-box의 최소 개수를 구하여 S-box의 최대 확률을 곱한 값보다 항상 작거나 같고 차분 특성의 최대 확률보다 항상 크거나 같다.

(증명) 사용하는 표기법은 다음과 같다.

p_k : S-box 차분 확률의 로그 값 ($k \in \{1,2,...,t\}$)

$(p_1 < p_2 < \cdots < p_t)$(최대 확률 : 2^{-p_1})

p_b : BOGI S-box 확률의 로그 값($b \in \{1,2,...,t\}$)

$obj1 = A[0][0] + A[0][1] + \cdots + A[r-1][w-1]$

$obj2 = p_1 A[0][0] + (p_b - p_1) B[0][0] + \cdots$
$+ p_1 A[r-1][w-1] + (p_b - p_1) B[r-1][w-1]$

$A: \{A[i][j] | 0 \le i \le r-1, 0 \le j \le w-1\}$

$B: \{B[i][j] | 0 \le i \le r-1, 0 \le j \le w-1\}$

$\{A^*, B^*\} \in \arg\min_{\{A,B\}}(obj2)$

$N_A = \min(obj1)$

$N_{A^{new}} = \sum_{i=0}^{r-1}\sum_{j=0}^{w-1} A^*[i][j]$

$N_{B^{new}} = \sum_{i=0}^{r-1}\sum_{j=0}^{w-1} B^*[i][j]$

N_{p_k} : 확률이 최대인 차분 특성에서 확률의 로그 값이 p_k인 S-box의 개수($k \in \{1,2,...,t\}$)

위의 정의들은 다음의 성질을 만족한다.

1. $N_A \le N_{A^{new}}$
2. $N_A \le \sum_{k=1}^{t} N_{p_k}$
3. $N_{B^{new}} \le N_{p_b}$

활성 S-box의 최소 개수와 MDP에 로그($-\log_2$)를 취해 곱한 값을 $p_1 N_A$로, 제안하는 방법으로 구한 차분 특성 확률의 상한은 $p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}}$로 나타낼 수 있고, 최대 확률의 로그 값은 $p_1 N_{p_1} + p_2 N_{p_2} + \cdots + p_t N_{p_t}$이다. 따라서 식(10)과 같은 부등식이 성립함을 보이면 된다.

$$\begin{aligned} p_1 N_A &\le p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}} \\ &\le p_1 N_{p_1} + p_2 N_{p_2} + \cdots + p_t N_{p_t} \end{aligned} \quad (10)$$

식(10)이 타당함을 보이기 위해 먼저, $p_1 N_A \le p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}}$를 만족하는지 확인한다. 성질 1로부터 $p_1 N_A \le p_1 N_{A^{new}}$임이 자명하여 $p_1 N_A \le p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}}$가 만족함을 알 수 있다. 다음으로는 $p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}} \le p_1 N_{p_1} + p_2 N_{p_2} + \cdots + p_t N_{p_t}$을 만족하는지 확인하면 된다. 위 식이 성립함을 보이기 위해 먼저 $N_A \le N_{p_1} + N_{p_2} + \cdots + N_{p_t}$ 이 성립하는 것을 보이면 된다. 이것은 성질 2로 자명하다. 다음으로 $p_1 N_{p_1} + p_2 N_{p_2} + \cdots + p_t N_{p_t}$이 $p_1(N_{p_1} + N_{p_2} + \cdots + N_{p_t}) + (p_b - p_1) N_{p_b}$보다 크거나 같은 것을 알 수 있고, $p_1 N_A + (p_b - p_1) N_{p_b}$보다 크거나 같은 것을 알 수 있다. 이제 $p_1 N_{A^{new}} + (p_b - p_1) N_{B^{new}}$보다 $p_1 N_A + (p_b - p_1) N_{p_b}$가 크거나 같은 것을 보이면 되고 이는 성질 3을 통해 자명한 것을 알 수 있다. □

본 논문에서 제안한 방법으로 탐색할 때, 만약 p_b가 p_1과 같다면 기존 방법의 상한과 같은 값을 갖는다. 즉, $p_1 \ne p_b$인 경우 개선된 상한을 얻을 수 있다.

V. 탐색방법 적용

본 장에서는 BOGI가 적용된 암호 GIFT에 대해서 앞서 설명한 BOGI를 고려한 제약식과 목적함수를 GIFT-64와 GIFT-128에 적용하여 각각의 차분 특성 확률의 상한을 탐색하고 결과에 대해 논한다. 또한, BOGI를 만족하는 S-box와 비트 순열을 생성하여 새로운 암호를 만든 후 상한을 탐색하고 탐색한 상한을 이용해 라운드를 결정해 보았다.

5.1 GIFT-64의 적용 결과

GIFT-64는 앞서 2장의 설명과 같이 총 16개의 S-box와 비트 순열을 사용한다. Table 5.는 MILP Solver를 이용하여 구한 차분 특성의 확률(Best Prob), 기존 방법으로 계산한 차분 특성의 상한(Bound(Old))과 본 논문에서 제안한 방법으로 탐색한 결과(Bound(New))를 표로 정리한 것이다.

Table 5. Best probability and bound of differential characteristic(GIFT-64)

Round	Best Prob.	Bound (Old)	Bound (New)
1	-1.415	-1.415	-1.415
2	-3.415	-2.83	-2.83
3	-7	-4.245	-5.66
4	-11.415	-7.075	-8.49
5	-17	-9.905	-11.49
6	-22.415	-14.15	-16.98
7	-28.415	-18.395	-19.81
8	-38	-22.64	-22.64
9	-42	-25.47	-25.47

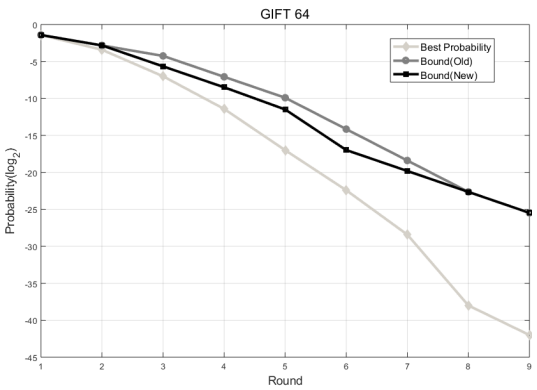

Fig. 4. Graphical representation of the experimental results. (GIFT-64)

3라운드부터 7라운드까지 기존 방법보다 개선된 결과를 얻었다. 이후 라운드부터는 기존과 동일한 결과를 얻었는데, 그 이유는 8라운드 이후 활성 S-box에 BOGI S-box가 포함되지 않고 반복 차분이 발생하여 같은 결괏값이 나오기 때문이다.

5.2 GIFT-128의 적용 결과

GIFT-128은 GIFT-64와 같은 S-box를 사용하며 워드 크기는 32이고 비트 순열이 달라진다. GIFT-64와 마찬가지로 MILP Solver를 통해 얻은 결과를 Table 6.와 같이 정리하였다. 그 결과 3라운드부터 9라운드까지 기존 방법보다 개선된 결과를 도출하였다. 차분 특성의 최대 확률이 7라운드 이후 결과가 없는 것은 7라운드 탐색 시간이 7일 이상 소비되어 탐색을 더 진행하지 않았다. GIFT-64와 마찬가지로 라운드가 증가하면 기존 방법과 동일한 결과를 갖게 될 것이다.

Table 6. Best probability and bound of differential characteristic(GIFT-128)

Round	Best Prob.	Bound (Old)	Bound (New)
1	-1.415	-1.415	-1.415
2	-3.415	-2.83	-2.83
3	-7	-4.245	-5.66
4	-11.415	-7.075	-8.49
5	-17	-9.905	-11.49
6	-22.415	-14.15	-16.98
7	-28.415	-18.395	-22.64
8	-	-24.055	-25.47
9	-	-26.885	-28.3

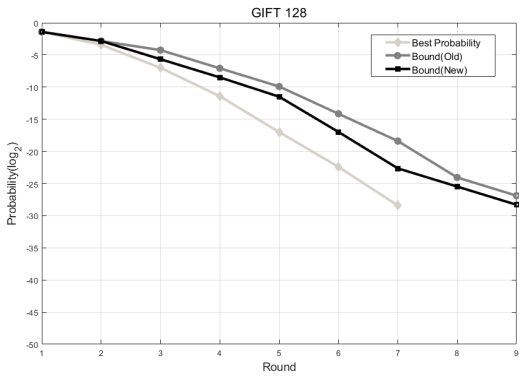

Fig. 5. Graphical representation of the experimental results. (GIFT-128)

5.3 BOGI 기반 암호

본 절에서는 설계자의 관점에서 제안한 방법을 이용하여 BOGI를 이용한 암호의 라운드를 정해본다. GIFT의 경우 [9]에서 S-box를 탐색할 때, 17-unit 이하인 조건을 주었다. 이 조건 때문에 4-uniform을 갖는 S-box를 찾을 수 없어 16-unit 중 6-uniform을 만족하는 S-box를 선택하였다. 이와 같은 기준으로 [14]에 제시된 4비트 S-box 중 16-unit 이하이고, 차분 관점에서 BOGI를 만족하고 6-uniform인 S-box는 Fig. 6.와 같다. 비트 순열은 GIFT의 비트 순열을 사용하면 BOGI를 만족한다.

설계된 64비트 암호의 차분 특성 확률의 상한을 구하여 Table. 8.과 같이 정리하였다. 8라운드까지만 탐색 가능하다고 하면, 2^{-64} 이하의 확률을 갖기 위해 기존 방법으로 구한 상한을 기준으로는 약 4배로 32라운드를 사용해야 하지만 제안한 방법으로는 약 3배로 24라운드를 사용하면 된다. 따라서 기존 방법과 비교하였을 때 8라운드를 줄여서 사용할 수 있다.

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

Fig. 6. Specification of BOGI 4-bit S-box

Table 7. Bound of differential characteristic (BOGI 4-bit)

Round	Bound(Old)	Bound(New)
1	-1.415	-1.415
2	-2.83	-2.83
3	-4.245	-5.66
4	-7.075	-8.49
5	-9.905	-11.49
6	-12.735	-16.98
7	-15.565	-19.81
8	-16.98	-22.64

Table 8. 1-1 bit DDT of BOGI 4-bit S-box

Δx \ Δy	1000	0100	0010	0001
1000	0	0	0	0
0100	2	0	0	0
0010	2	0	0	2
0001	0	0	0	0

VI. 결 론

본 논문은 MILP Solver를 이용하여 BOGI를 고려한 차분 특성 확률의 상한을 구하는 방법에 관해 설명하였다. 제안하는 방법을 BOGI가 적용된 GIFT에 적용하였고, GIFT를 제안한 논문에서 계산한 차분 특성 확률의 상한보다 더 개선된 결과를 얻을 수 있었다. GIFT-64의 7라운드와 GIFT-128의 9라운드에 대해 기존의 방법을 이용하면 차분 특성 확률의 상한이 각각 $2^{-18.395}$와 $2^{-26.885}$이었으나, 제안한 방법을 적용하면 각각 $2^{-19.81}$과 $2^{-28.3}$으로 개선된 결과를 얻었다. 제안하는 방법을 이용하여 BOGI로 설계될 암호들의 차분 분석을 진행할 때, 차분 특성 확률의 상한을 더 정밀하게 계산할 수 있다. 합리적인 시간 내에 차분 특성 확률의 상한을 더 정밀하게 구하는 것은 설계자가 더 적은 라운드 수를 사용함으로써 암호의 성능을 향상시킬 수 있다. 향후 차분 분석의 BOGI처럼 선형 분석에서 비슷한 설계 원리를 갖는 암호분석에도 본 논문에서 제안한 방법을 유용하게 사용할 수 있을 것이다.

References

[1] Biham, Eli, and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems." Journal of CRYPTOLOGY vol. 4, no. 1 pp. 3-72, 1991.

[2] Daemen, Joan, and Vincent Rijmen. "The design of Rijndael: AES-the advanced encryption standard." Springer Science & Business Media, 2013.

[3] Bogdanov, Andrey, et al. "PRESENT: An ultra-lightweight block cipher." International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, pp. 450-466, Sep. 2007.

[4] Shirai, Taizo, et al. "The 128-bit blockcipher CLEFIA." International workshop on fast software encryption. Springer, Berlin, Heidelberg, pp. 181-195, Mar. 2007.

[5] Mouha, Nicky, et al. "Differential and linear cryptanalysis using mixed-integer linear programming." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, pp. 57-76, Nov. 2011.

[6] Sun, Siwei, et al. "Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks." International Conference on Information Security and Cryptology. Springer, Cham, pp. 39-51, Nov. 2013.

[7] 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, pp. 158-178, Dec. 2014.

[8] Sun, Siwei, et al. "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 747 (2014): 2014.

[9] Banik, Subhadeep, et al. "GIFT: a small present." International Conference on Cryptographic Hardware and Embedded Systems. Springer, Cham, pp. 321-345, Sep. 2017.

[10] Bertoni, Guido, et al. "Sponge functions." ECRYPT hash workshop. Vol. 2007. No. 9. 2007.

[11] Nyberg, Kaisa. "Differentially uniform mappings for cryptography." Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, pp. 55-64, May. 1993.

[12] Sasaki, Yu, and Yosuke Todo. "New algorithm for modeling S-box in MILP based differential and division trail search." International Conference for Information Technology and Communications. Springer, Cham, pp. 150-165, Jun. 2017.

[13] Brayton, Robert K., et al. "Logic minimization algorithms for VLSI synthesis." Springer Science & Business Media, Vol. 2. 1984.

[14] Ullrich, Markus, et al. "Finding optimal bitsliced implementations of 4× 4-bit S-boxes." SKEW 2011 Symmetric Key Encryption Workshop, Copenhagen, Denmark. pp. 16-17, Feb. 2011.

〈저 자 소 개〉

이 상 협 (Sanghyeop Lee) 학생회원
2018년 2월: 강남대학교 응용수학과 졸업
2018년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호 알고리즘 설계 및 분석, 대칭키 암호

김 성 겸 (Seonggyeom Kim) 학생회원
2016년 8월: 한양대학교 수학과 졸업
2016년 9월~2018년 8월: 고려대학교 정보보호대학원 석사
2019년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 암호 알고리즘 설계 및 분석, 대칭키 암호, 난수발생기

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 부교수
〈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