Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.1, Feb. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.1.15

SAT 솔버를 이용한 ARX 구조 암호의 디퍼렌셜 확률 근사값 계산*

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

Calculating the Approximate Probability of Differentials for ARX-Based Cipher Using SAT Solver*

HoChang Lee,[1†] Seojin Kim,[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

본 논문에서는 SAT 솔버를 이용하여 디퍼렌셜 확률의 근사값을 구하는 방법에 대해서 설명한다. SAT 솔버를 이용해서 이미 알려진 차분특성을 디퍼렌셜로 구성하여 확률을 높일 수 있다. 본 논문에서 제안하는 방법으로 SPECK32와 SPECK48의 디퍼렌셜을 구성하였다. SPECK32는 $2^{-30.39}$의 확률을 가지는 10-라운드 디퍼렌셜을 찾아냈고, SPECK48은 $2^{-46.8}$의 확률을 가지는 12-라운드 디퍼렌셜을 찾아냈다. 두 특성 모두 가장 길고 높은 확률을 가지는 새로운 디퍼렌셜이다. 본 논문에서 발견한 디퍼렌셜을 이용해서 SPECK32/64의 15 라운드 차분공격, SPECK48/72의 16 라운드 차분공격, SPECK48/96의 17 라운드 차분공격을 하여 이전 공격보다 각각 1 라운드씩 개선하였다.

ABSTRACT

In this paper, we explain a method of appproximating the differentials probability using a SAT solver. It is possible to increase the probability by constructing the differential characteristic which already known to differentials with a SAT solver. We apply our method to SPECK32 and SPECK48. As a result, we introduced a SPECK32's 10-round differentials with a probability of $2^{-30.39}$, and SPECK48's 12-round differentials with probability of $2^{-46.8}$. Both differentials are new and longer round and higher probability than previous ones. Using the differentials presented in this paper, we improved attacks of SPECK32/64 to 15 round, SPECK48/72 to 16 round, SPECK48/96 to 17 round, which attack 1 more rounds of previous works.

Keywords: Differential Cryptanalysis, Differentials SAT, SPECK

Received(09. 26. 2017), Modified(12. 05. 2017), Accepted(12. 08. 2017)
* 본 논문은 2017년 한국정보보호학회 하계학술대회에 발표한 우수논문을 개선 및 확장한 것임
* 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)

I. 서 론

블록암호는 암호시스템에서 가장 많이 사용되는 암호학적 핵심요소이다. 현대 블록암호 설계는 안전성과 더불어 구현 효율성등 여러 목적을 만족하도록 설계되고 있다. 최근 경량 환경에서 적합한 암호가 많이 필요하게 되었다. ARX 구조 암호는 경량 환경에 적합한 암호로, 혼돈(Confusion) 효과를 제공하기 위해서 법덧셈(Addition Modulo 2^n)과 확산(Diffusion) 효과를 제공하기 위해서 비트 순환이동(Rotation), 배타적 논리합(eXclusiveOR, XOR)으로 이루어진 라운드 함수를 반복적으로 사용하는 경량 블록암호이다. 대표적인 ARX 구조 블록암호에는 SPECK[1]과 LEA[2]가 있다.

블록암호의 안전성 분석 방법은 매우 많으며, 이중 차분분석(Differential Cryptanalysis)[3]은 선형분석(Linear Cryptanalysis)[4]과 함께 블록암호를 분석하는 대표적인 방법이다. 차분분석은 블록 암호 E에 대해 어떤 입력 차분(두 평문의 XOR 값)이 랜덤보다 더 높은 확률로 특정 출력 차분(대응되는 암호문의 XOR 값)을 발생시키는 성질(디퍼렌셜, Differentials)을 이용하는 공격법이다. 디퍼렌셜의 정확한 확률을 계산하는 것은 매우 어렵기 때문에, 대부분의 경우 모든 라운드가 독립인 것으로 가정하였을 때 가장 높은 확률을 갖는 차분 경로로 구성되는 차분특성(Differential Characteristic)을 이용한다. 차분특성은 라운드 함수의 입·출력 차분 값과 그 확률을 조사하여 구성할 수 있다. 차분특성 구성은 어렵기 때문에 복잡한 과정을 피하기 위해서 [5,6]등과 같이 자동 탐색 연구가 되었다. 최근에는 임의의 문제를 수학적으로 모델링하여 해를 찾기 위한 도구들이 자동 탐색에서 중요한 역할을 하고 있다. 두 가지 도구가 대표적으로 사용되는데 하나는 Mixed-Integer Linear Programming(MILP) 기반 툴이고, 다른 하나는 SAT/SMT 기반 툴이다.

이진 충족 가능성 문제(SAT, Boolean Satisfiability Problem)는 이진변수들로 이루어진 논리식을 참이 되게 하는 해를 찾는 결정 문제이다. SMT(Satisfiability Modulo Theories)[10]는 SAT의 확장으로 선형식, 배열, 벡터등의 입력방식을 지원한다. SMT 솔버를 이용해서 Kircanski등[7]은 부메랑 특성을 찾는 방법을 제안하였고, Song등[8]은 같은 입·출력 차분특성을 다수 구하여 확률을 높이는 방법에 대해서 제안하였다. 또한 부채널 공격[9]등 많은 연구가 진행중이다. 최근에는 Rotational-Xor Cryptanalysis의 특성을 자동 탐색하는 연구가 진행 중이다[17,18].

본 논문에서는 SAT 솔버를 이용한 ARX 구조 암호의 디퍼렌셜 확률 근사값 계산 방법을 제안한다. 본 논문의 방법으로 일반적인 방법으로 찾을 수 없었던 SPECK32의 $2^{-30.39}$의 확률을 가지는 10-라운드 디퍼렌셜을 찾아냈고, SPECK48의 $2^{-46.8}$의 확률을 가지는 12-라운드 디퍼렌셜을 찾아냈다. 해당 디퍼렌셜은 기존 SPECK32와 SPECK48의 구별자(Distingusiher)와 비교해 가장 길고 높은 확률을 가지며 이를 이용해 더 긴 라운드의 차분분석이 가능하였다. 차분분석의 요약은 Table 1.과 같다. 2장에서는 SAT 솔버를 이용한 디퍼렌셜 확률의 근사값을 계산하기 위한 사전지식에 대해서 설명한다. 3장에서는 SAT 솔버를 이용한 디퍼렌셜 확률의 근사값 계산 방법에 대해서 설명한다. 4장에서는 본 논문의 방법

Table 1. Comparison of our differentials of SPECK32,48 with previous works.

Variant	Rounds Attacked/ Total Rounds	Time Complexity	Data Complexity	Memory Complexity	Reference
SPECK32/64	14/22	2^{63}	2^{31}	2^{22}	[14]
	14/22	$2^{62.47}$	$2^{30.47}$	2^{22}	[8]
	15/22	$2^{63.39}$	$2^{31.39}$	2^{22}	This paper
SPECK48/72	14/22	2^{65}	2^{41}	2^{22}	[14]
	15/22	$2^{69.31}$	$2^{45.31}$	2^{22}	[8]
	16/22	$2^{71.8}$	$2^{47.8}$	2^{22}	This paper
SPECK48/96	15/23	2^{89}	2^{41}	2^{22}	[14]
	16/23	$2^{93.31}$	$2^{45.31}$	2^{22}	[8]
	17/23	$2^{95.8}$	$2^{47.8}$	2^{22}	This paper

으로 SPECK의 디퍼렌셜을 찾는 방법을 설명하고 5장에서 결론을 맺는다.

II. 사전 지식

2.1 표기법

$\oplus$: 배타적 논리합(eXclusive OR, XOR)
$\neg$: 보수(complement)
$\boxplus$: 법덧셈(modular addition)
$\wedge$: 논리곱(AND, logical conjuction)
$\vee$: 논리합(OR, logical disjunction)
$\ggg a(\lll a)$: 오른쪽(왼쪽)으로 a번 비트 순환이동
$a \| b$: a와 b의 연접
$x[i]$: x의 i번째 최하위 비트

2.2 이진 충족 가능성 문제(SAT)

이진 충족 가능성 문제(Boolean Satisfiability Problem, SAT)는 이진변수로 이루어진 논리식이 주어졌을 때, 논리식이 참이 되게 하는 해가 존재하는지 찾는 결정 문제이다. SAT는 이진변수의 개수, 이진변수의 논리합(절, Clause)의 개수, 절의 논리곱으로 표현된 논리식(CNF, Conjunctive Normal Form)을 입력 받는다. 논리식이 참이 되게 하는 이진변수 값이 존재할 경우 SAT(Satisfiable)를 반환하고, 존재하지 않을 경우 UNSAT(Unsatisfiable)를 반환한다.

SAT를 이용해 문제의 해를 구하기 위해서는 문제를 이진변수를 이용해서 절의 논리곱으로 표현해야 한다. 하지만 문제를 이진변수로 이루어진 절의 논리곱으로 표현하기 어렵기 때문에 SMT가 존재한다. SMT는 SAT의 확장으로 다양한 변수의 1차 표현 방식을 입력으로 허용한다.

본 논문에서는 SAT 솔버를 직접 이용하여 ARX 구조 암호의 디퍼렌셜 확률 근사값을 계산하는 방법을 설명한다. 사용한 SAT 솔버는 CryptoMiniSat 5.0.1[16]을 Ubuntu 16.04를 운영체제로 하여 Intel i7-6700 @ 3.40GHz 프로세서, 16GB RAM의 데스크탑에서 실행하였다.

2.3 배경지식 및 정리

ARX 구조 암호의 법덧셈에서의 차분의 확률을 계산하기 위해서는 Lipmaa와 Moriai[11]가 제안한 두 가지 정리를 이용한다.

정리 1. 법덧셈 차분의 존재성 [11]

$(\alpha,\beta\rightarrow\gamma)$을 입력 차분이 α,β이고 출력 차분이 γ인 법덧셈 차분이라 할 때, 법덧셈 차분 $(\alpha,\beta\rightarrow\gamma)$이 0이 아닌 확률을 가질 필요 충분 조건은 아래의 조건이다.

$eq(\alpha \ll 1,\beta \ll 1,\gamma \ll 1) \wedge (\alpha \oplus \beta \oplus \gamma \oplus (\alpha \ll 1)) = 0$

이때 $eq(x,y,z) = (\neg x \oplus y) \wedge (\neg x \oplus z)$이다.

정리 2. 법덧셈 차분의 확률 [11]

법덧셈 차분 $(\alpha,\beta\rightarrow\gamma)$이 0이 아닌 확률을 가질 경우, 법덧셈 차분 $(\alpha,\beta\rightarrow\gamma)$의 확률은 $2^{-\sum_{i=0}^{n-2} all(\alpha[i],\beta[i],\gamma[i])}$이다.

이 때 $all(a,b,c) = \begin{cases} 1 & (a=b=c) \\ 0 & others \end{cases}$ 이다.

정리1로 법덧셈 차분 $(\alpha,\beta\rightarrow\gamma)$이 가능한 차분만을 표현해 차분특성을 구성할 수 있다. 따라서 정리1을 이용해 가능한 차분성질로 차분특성을 구성하고, 그 확률을 정리2를 이용해 계산할 수 있다.

III. SAT 솔버를 이용한 디퍼렌셜 확률의 근사값 계산

많은 라운드의 차분 공격을 위해서는 높은 확률을 가진 긴 라운드의 차분특성이 필요하다. 차분특성의 길이는 한계가 있기 때문에 확률을 높이기 위해서는 같은 입·출력 차분 값을 가지는 디퍼렌셜의 확률을 계산할 필요가 있다. 하지만 디퍼렌셜의 확률을 계산하는 것은 어렵기 때문에 같은 입·출력 차분값을 가지는 차분특성의 확률을 더하는 것으로 디퍼렌셜 확률의 근사값을 계산할 수 있다. 본 논문에서는 입·출력 차분값이 같은 차분특성을 찾는 것은 어렵기 때문에, 같은 입·출력 차분값을 가지는 차분특성을 찾기 위해서 SAT 솔버를 이용하였다. 본 절에서는 ARX 구조 암호의 입·출력 차분값이 같은 차분특성을 찾기 위해서 ARX 구조 암호의 성질을 논리곱 형태 표현하는 방법, 확률 제약식 표현하는 방법, SAT 솔버를 이용하여 ARX 구조 암호의 디퍼렌셜 확률 근사값을 계산하는 방법에 대해서 설명한다.

3.1 ARX 구조 암호의 절의 논리곱 표현 방법

ARX 구조 암호의 특정 차분특성을 SAT 솔버를 이용하여 찾기 위해서는 ARX 구조 암호의 XOR 연산에 대한 차분성질을 절의 논리곱 형태로 표현해야 한다. ARX 구조 암호의 XOR연산에 대한 차분성질을 절의 논리곱으로 표현하기 위해서는 법덧셈, 비트 순환이동, XOR 연산의 차분성질을 표현하면 된다.

법덧셈: 법덧셈 차분 $(\alpha,\beta\rightarrow\gamma)$의 차분성질을 표현하기 위해서는 Lipmaa와 Moriai의 정리1을 이용하여 가능한 경우만을 남기면 된다. 정리1을 비트별 조건식으로 표현하면 아래와 같다.

$$(\neg\alpha[i-1]\oplus\beta[i-1])\wedge(\neg\alpha[i-1]\oplus\gamma[i-1])\\ \rightarrow\neg(\alpha[i]\oplus\beta[i]\oplus\gamma[i]\oplus\alpha[i-1])\ \ i\in[0,n-2]$$

위 조건식을 SAGE의 convert_cnf_table()함수를 이용해서 아래와 같은 8개의 절의 논리곱 형태로 표현된다.

$$(\alpha[i]\vee\beta[i]\vee\gamma[i]\vee\alpha[i+1]\vee\beta[i+1]\vee\neg\gamma[i+1])\wedge\\ (\alpha[i]\vee\beta[i]\vee\gamma[i]\vee\alpha[i+1]\vee\neg\beta[i+1]\vee\gamma[i+1])\wedge\\ (\alpha[i]\vee\beta[i]\vee\gamma[i]\vee\neg\alpha[i+1]\vee\beta[i+1]\vee\gamma[i+1])\wedge\\ (\neg\alpha[i]\vee\neg\beta[i]\vee\neg\gamma[i]\vee\alpha[i+1]\vee\beta[i+1]\vee\gamma[i+1])\wedge\\ (\alpha[i]\vee\beta[i]\vee\gamma[i]\vee\neg\alpha[i+1]\vee\neg\beta[i+1]\vee\neg\gamma[i+1])\wedge\\ (\neg\alpha[i]\vee\neg\beta[i]\vee\neg\gamma[i]\vee\neg\alpha[i+1]\vee\neg\beta[i+1]\vee\gamma[i+1])\wedge\\ (\neg\alpha[i]\vee\neg\beta[i]\vee\neg\gamma[i]\vee\neg\alpha[i+1]\vee\beta[i+1]\vee\neg\gamma[i+1])\wedge\\ (\neg\alpha[i]\vee\neg\beta[i]\vee\neg\gamma[i]\vee\alpha[i+1]\vee\neg\beta[i+1]\vee\neg\gamma[i+1])$$

n비트 법덧셈을 표현하기 위해서 $8(n-1)$개의 절로 법덧셈의 차분성질을 표현할 수 있다.

비트 순환이동: 비트 순환이동은 XOR 연산에 대해 선형연산이므로 원래 비트 순환이동 연산으로 진행된다. 따라서 다른 연산을 표현하는 과정에서 비트를 참조할 때, 비트 순환이동에 맞게 비트를 참조하는 것으로 비트 순환이동의 차분성질을 표현할 수 있다.

XOR: XOR 차분에 대해서 XOR 연산은 선형이므로 원래 XOR 연산으로 진행된다. $a\oplus b\oplus c=0$을 절의 논리곱으로 표현하기 위해서는 2^2개의 절이 필요하다. 하지만 본 논문에서 사용한 SAT 솔버인 CryptoMiniSAT에서는 자주 사용되는 XOR 연산을 XOR 절(XOR Clause)로 입력을 허용한다. 따라서 $a\oplus b\oplus c=0$을 표현하기 위해서 1개의 XOR 절로 표현이 가능하다.

ARX 구조 암호의 각 연산을 위와 같이 표현하는 것으로 한 라운드 차분성질을 표현한 절의 논리곱 형태로 표현이 가능하다. 따라서 라운드를 반복 구성하고 특정 입·출력 차분값에 대한 표현도 추가해서 원하는 입·출력 차분값의 차분특성을 찾는 모델을 만들 수 있다. 만든 모델을 이용해 SAT 솔버로 차분특성을 찾을 수 있다.

3.2 확률 제약식 표현 방법

3.1절의 방법으로 만든 모델을 이용해 SAT 솔버로 ARX 구조 암호의 특정 입·출력 차분값을 가지는 차분특성을 찾을 수 있다. 디퍼렌셜의 근사값 확률을 구하기 위해서는 높은 확률을 가지는 차분특성이 필요하지만 3.1절의 방법은 특정 입·출력 차분값의 차분특성만 찾을 뿐 높은 확률의 차분특성을 보장하지 못한다. 따라서 높은 확률을 가지는 차분특성을 위해 3.1절의 방법으로 만든 모델에 확률 제약식을 절의 논리곱 표현으로 추가하는 방법에 대해서 설명한다.

ARX 구조 암호는 정리2에 의해서 법덧셈의 최상위 비트를 제외한 비트에서만 차분특성의 확률이 발생한다. 또한 비트당 발생하는 확률은 $2^0(=1)$과 2^{-1}로 $-\log_2$ 값을 취했을 경우 0과 1뿐이다. 이러한 사실을 바탕으로 원하는 확률의 차분특성을 찾기 위한 확률 카운터를 설정할 수 있다. 아래는 ARX 구조 암호의 6비트 확률 카운터 예이다. pc_i와 p_0는 피연산자이고 s_i는 결과값이고, c_{i-1}는 pc_i에서 발생하는 자리올림수이다.

$$\begin{array}{cccccccc} & & \curvearrowleft c_0 & \curvearrowleft c_1 & \curvearrowleft c_2 & \curvearrowleft c_3 & \curvearrowleft c_4 & \\ & pc_0 & pc_1 & pc_2 & pc_3 & pc_4 & pc_5 & \\ + & & & & & & p_0 & \\ \hline & s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & \end{array}$$

위와 같은 확률 카운터의 논리 표현은 아래와 같다.

$$s_5 = pc_5 \oplus p_0 \qquad c_4 = pc_5 \wedge p_0$$
$$s_4 = pc_4 \oplus c_4 \qquad c_3 = pc_4 \wedge c_4$$
$$s_3 = pc_3 \oplus c_3 \qquad c_2 = pc_3 \wedge c_3$$
$$s_2 = pc_2 \oplus c_2 \qquad c_1 = pc_2 \wedge c_2$$
$$s_1 = pc_1 \oplus c_1 \qquad c_0 = pc_1 \wedge c_1$$
$$s_0 = pc_0 \oplus c_0$$

p_0는 정리2를 이용해 $p_0 = \neg all(\alpha[i], \beta[i], \gamma[i])$ $(i \in \{0,1,\cdots,n-2\})$와 같이 설정할 수 있다. 따라서 SAGE의 convert_cnf_table() 함수를 이용해서 s_0는 10개의 절, 그 이외 s_i는 1개의 XOR 절로 표현이 가능하다. c_0는 6개의 절, 그 이외의 c_i는 $a = b \wedge c$의 경우로 $(a \vee \neg b \vee \neg c)$, $(\neg a \vee b)$, $(\neg a \vee c)$로 3개의 절로 표현이 가능하다. 따라서 n-비트 확률 카운터 1개를 절의 논리곱 형태로 표현하기 위해서는 $3n-1(=n+n+(n-1))$개의 변수와 $n-1$개의 XOR 절, $3n+10(=10+6+3(n-2))$개의 절로 총 $4n+9$개의 절이 추가로 필요하다.

3.3 디퍼렌셜 확률의 근사값 계산 방법

3.1절과 3.2절의 방법을 이용해서 ARX 구조 암호의 특정 입·출력 차분값의 특정 확률을 가지는 차분특성을 찾기 위한 모델로 표현이 가능하다. 해당 모델을 이용해 디퍼렌셜 확률의 근사값 계산이 가능하며 구하는 과정은 Algorithm1과 같다.

Algorithm1은 특정 입·출력 차분값(Δ_{in}, Δ_{out})과 라운드(r), 확률(p), 찾는 횟수(N)을 입력받아 2^{-p}의 확률을 가지는 r-라운드 차분특성부터 $2^{-(p+N)}$의 확률을 가지는 r-라운드 차분특성을 구해서 각 차분특성의 확률을 모두 더하여 디퍼렌셜 확률의 근사값을 구한다. SAT 솔버의 입력되는 모델 $\mathcal{M}$은 $2^{-(p+i)}$ 확률($i \in \{0,1,\cdots,N\}$)의 특정 입·출력 차분값인 r-라운드 차분특성을 표현한 모델이다. SAT 솔버로 얻은 해는 p 확률을 가지는 해당 입·출력 차분값의 r-라운드 차분특성을 의미한다. 따라서 SAT 솔버로 구한 각 해는 $2^{-(p+i)}$의 확률을 가지는 $(\Delta_{in} \xrightarrow{r} \Delta_{out})$ 차분특성을 의미한다. 구한 차분특성의 확률을 더하는 것으로 해당 입·출력 차분값을 가지는 r-라운드 디퍼렌셜 확률의 근사값을 구할 수 있다.

IV. SPECK에 대한 적용

본 장에서는 3장의 방법을 이용해서 ARX 구조 블

Algorithm 1. Calculation for Differentials of Differential Characteristic

Input:
- Δ_{in}: Input difference
- Δ_{out}: Output difference
- r: The number of rounds you want to find
- p: $-\log_2$(Input Probability)
- N: The number of times for differentials construction

Output:
- Approximation probability of r-round differentials with $(\Delta_{in}, \Delta_{out})$

```
1. // Calculate the approximate probability of differentials
2. procedure calculate_differentials(Δx, Δy, r, p) do
3.   diffs_prob = 0
4.   for i∈{0,1,⋯,N} do
5.     create a model M for r-round differential characteristic.  // Sec 3.1
6.     add a constraints to model M with probability 2^-(p+i).    // Sec 3.2
7.     Solve the model using SAT solver
8.     diffs_prob = diffs_prob + ∑(probability of all solutions)
9.   return diffs_prob
```

록암호인 SPECK family의 디퍼렌셜과 공격에 대해서 설명한다.

4.1 SPECK Family

SPECK[1]은 NSA가 2013년에 공개한 ARX 구조 경량 블록암호이다. 파라미터 분류에 따라 10가지 종류를 가진다. 블록 크기 $2n(n\in\{16,24,32,48,64\})$이고 키 크기가 $mn(m\in\{2,3,4\},\ n$에 따라 결정)인 종류를 SPECK$2n/mn$으로 표기한다. i번째 라운드의 입·출력값을 각각 x_i, y_i라 하고 라운드 키를 k_i라고 할 때, i번째 라운드 함수는 아래와 같다.

$$F(x_i,y_i)=\{(x_i \gg \alpha)\boxplus y_i\}\oplus k_i \| (y_i \ll \beta)\oplus x_i$$

비트 순환이동 상수인 α와 β는 파라미터는 워드의 크기가 32 비트 일 경우 $\alpha=7,\ \beta=2$이고, 나머지는 $\alpha=8,\ \beta=3$이다.

4.2 SPECK의 디퍼렌셜

SPECK의 디퍼렌셜 확률의 근사값을 구하기 위해서는 3장에서 설명한 방법을 이용해 SPECK을 절의 논리곱 형태의 모델로 표현해야 한다. 3.1절의 방법을 이용해 SPECK$2n/mn$의 한 라운드를 절의 논리곱 형태로 표현하기 위해서는 입·출력 변수 $4n(=2n\times 2)$개와 법덧셈의 $8(n-1)+1$개의 절과 XOR의 n개의 절이 필요하다. 또한 3.2절의 방법을 이용해 한 라운드 8-비트 확률 카운터를 절의 논리곱 형태로 표현하기 위해서는 $23(n-1)+8$개의 변수와 $49n$개의 절이 추가로 필요하다. 따라서 8-비트 확률 카운터의 r-라운드를 표현하기 위해서는 총 $2nr+2n+23(n-1)r+8$개의 변수와 $58nr-7r$개의 절이 필요하다. 변수에 대한 정보는 Fig 1.과 같다.

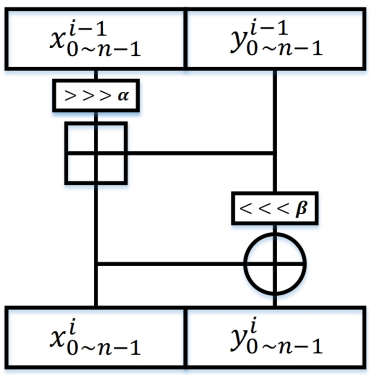


Fig. 1. ith round variables of SPECK$2n/mn$

[12]에서 제안하는 비트단위 차분특성 탐색 알고리즘을 이용해서 SPECK32의 2^{-34}의 확률을 가지는 10-라운드 차분특성 1개와 2^{-35}의 확률을 가지는 10-라운드 차분특성 33개를 찾아냈으며 SPECK48의 2^{-49}의 확률을 가지는 12-라운드 차분특성 6개를 찾아냈다. SPECK32의 34개의 차분특성과 SPECK48의 6개의 차분특성은 모두 구별자로서 의미는 없다. 하지만 각 차분특성을 Algorithm1을 이용해서 각각의 디퍼렌셜 확률의 근사값을 구하였고, 구별자로 의미있는 차분특성을 찾을 수 있다. SPECK32는 2^{-35}의 확률을 가지는 10-라운드 차분특성 (2800,0010)

Table 2. SPECK32's 10-round Differentials with (2800,0010)→(0004,0014)

$poss$	# of solutions	$\sum poss$	Time
2^{-35}	1	2^{-35}	5s
2^{-36}	1	2^{-36}	8s
2^{-37}	4	2^{-35}	8s
2^{-38}	23	$2^{-33.48}$	8s
2^{-39}	45	$2^{-33.51}$	10s
2^{-40}	115	$2^{-33.15}$	11s
2^{-41}	188	$2^{-33.45}$	11s
2^{-42}	268	$2^{-33.93}$	18s
2^{-43}	465	$2^{-34.14}$	20s
2^{-44}	624	$2^{-34.71}$	26s
2^{-45}	909	$2^{-35.17}$	43s
2^{-46}	1,383	$2^{-35.57}$	58s
2^{-47}	2,113	$2^{-35.95}$	1m 1s
2^{-48}	3,371	$2^{-36.28}$	2m 2s
2^{-49}	5,639	$2^{-36.54}$	3m 27s
2^{-50}	10,106	$2^{-36.7}$	7m 38s
2^{-51}	19,172	$2^{-36.77}$	15m 28s
2^{-52}	37,210	$2^{-36.82}$	33m 43s
2^{-53}	71,562	$2^{-36.87}$	1h 19m
2^{-54}	136,937	$2^{-36.94}$	3h 24m
2^{-55}	258,883	$2^{-37.02}$	13h 40m
$\sum\sum poss$		$2^{-30.39}$	

Table 3. SPECK48's 12-round Differentials with (080048,080800)→(840084,A00080)

$poss$	# of solutions	$\sum poss$	Time
2^{-49}	3	$2^{-47.42}$	38s
2^{-50}	0	0	51s
2^{-51}	1	2^{-51}	38s
2^{-52}	0	0	51s
2^{-53}	6	$2^{-50.42}$	1m 22s
2^{-54}	13	$2^{-50.3}$	55s
2^{-55}	6	$2^{-52.42}$	1m 36s
2^{-56}	18	$2^{-51.83}$	2m 4s
2^{-57}	16	2^{-53}	2m 22s
2^{-58}	46	$2^{-52.48}$	2m 39s
2^{-59}	71	$2^{-52.85}$	4m 45s
2^{-60}	89	$2^{-53.52}$	6m 36s
2^{-61}	163	$2^{-53.65}$	9m 18s
2^{-62}	181	$2^{-54.5}$	14m 52s
$\sum\sum poss$		$2^{-46.8}$	

$\xrightarrow{10}$(0004,0014)은 디퍼렌셜 확률의 근사값이 $2^{-30.39}$으로 구별자로서 의미를 갖는다. SPECK32의 디퍼렌셜의 각 해는 Table 2.와 같다.

SPECK48은 2^{-49}의 확률을 가지는 12-라운드 차분특성 (080048,080800)$\xrightarrow{12}$(840084,A00080)과 (0800C8,080800)$\xrightarrow{12}$(840084,A00080)은 디퍼렌셜 확률의 근사값이 $2^{-46.8}$으로 구별자로서 의미를 갖는다. SPECK48의 한 디퍼렌셜의 각해는 Table 3.과 같다.

SPECK32의 2^{-35}의 확률을 가지는 10-라운드 차분특성 (2800,0010)$\xrightarrow{10}$(0004,0014)의 디퍼렌셜의 확률을 5,000개의 랜덤한 키와 2^{32}개의 평문을 이용해서 실험적으로 측정해 보았다. 차분특성의 확률이 2^{-35}이므로 차분특성을 만족하는 평문의 기댓값은 0개라 할 수 있다. 하지만 실험결과 차분특성을 만족하는 평문의 기댓값은 약 4로 나왔으며, 본 논문의 방법으로 측정한 디퍼렌셜 확률의 근사값인 $2^{-30.39}$으로 차분특성이 구별자로서 이용될 수 있다는 것을 알 수 있다. 따라서 본 논문에서 찾는 SPECK32와 SPECK48의 디퍼렌셜은 [13]과 비교해서 각각 더 높은 확률, 더 긴 라운드를 가지는 새로운 구별자 역할을 한다. 해당 결과는 Table 4.와 같다.

4.3 SPECK32와 SPECK48의 차분공격

[14]에서 Dinur는 guess-and-determine 방법을 이용해서 SPECK의 일반적인 키 복구 공격에 대해서 제안하였다. SPECK$2n/mn$의 $p(\geq 2^{-2n+1})$의 확률을 가지는 r-라운드 차분특성이 존재할 경우, 공격자는 해당 차분특성 앞에 한 라운드와 뒤쪽에 m-라운드를 붙여 $(r+m+1)$-라운드를 공격할 수 있다. 해당 공격은 $2p^{-1}$의 선택 평문이 필요하고, $2p^{-1}2^{(m-2)n}$의 시간 복잡도, 2^{22}의 메모리 복잡도가 필요하다.

Table 4. SPECK's DIfferential Characteristic(D) and Differentials(Ds)

Variant	Rounds	Probability	D/Ds	Input/Output Differentials	Reference
SPECK32	9	2^{-30}	D	0211 0A04 / 1001 5001	[13]
	9	2^{-31}	D	0A60 4205 / 81A8 D30B	[15]
	10	$2^{-31.99}$	Ds	2040 0040 / A840 0800	[8]
	10	$2^{-30.39}$	Ds	2800 0010 / 0004 0014	This paper
SPECK48	10	2^{-41}	D	480B01 094009 / 808524 84A805	[15]
	11	2^{-45}	D	001202 020002 / 210020 200021	[13]
	11	$2^{-44.31}$	Ds	504200 004240 / 202001 202000	[8]
	12	$2^{-46.8}$	Ds	080048 080800 / 840084 A00080	This paper

본 논문에서 제안하는 방법으로 찾은 SPECK32와 SPECK48의 디퍼렌셜을 이용해서 각각의 더 긴 라운드 공격을 할 수 있었다. 각 공격의 요약은 Table 1.과 같다.

V. 결 론

본 논문은 SAT 솔버를 이용해서 ARX 구조 암호의 디퍼렌셜 확률 근사값 계산 방법에 대해서 설명하였다. 제안하는 방법을 이용해서 SPECK에 적용하였고, SPECK32의 10-라운드 확률 $2^{-30.39}$의 확률을 가지는 디퍼렌셜과 SPECK43의 12-라운드 확률 $2^{-46.8}$의 확률을 가지는 디퍼렌셜을 찾아냈다. 논문에서 찾은 디퍼렌셜을 이용하여 SPECK32/64, SPECK48/72, SPECK48/96의 차분공격을 각각 1 라운드씩 개선하였다. 제안하는 방법을 이용해 다른 ARX 구조 암호의 디퍼렌셜을 찾아내 안전성 분석이 가능하다.

References

[1] Ray Beaulieu, Douglas Shors, Jason Smith. "The SIMON and SPECK lightweight block ciphers." Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE. IEEE, pp. 1-6, 2015.

[2] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, Dong-Geon Lee. "LEA: A 128-bit block cipher for fast encryption on common preocessors.", WISA 13: 14th vol. 8267 pp. 3-27, Aug, 2014.

[3] Eli Biham and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems." CRYPTO'90, Lecture Notes in Computer Science, vol. 537, pp. 2-21, August, 1991.

[4] Mitsuru Matsui. "Linear cryptoanalysis method for DES cipher.", EUROCRYPT'93, Lecture Notes in Computer Science, vol. 765, pp. 386 - 397, May, 1994.

[5] Alex Biryukov and Vesselin Velichkov. "Automatic search for differential trails in ARX ciphers." CTRSA 2014, Lecture Notes in Computer Science, vol. 8366, pp. 227-250, Feb, 2014.

[6] Alex Biryukov and Ivica Nikolic. "Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia,", EUROCRYPT 2010, Lecture Notes in Computer Science, vol. 6110, pp. 322-344, May, 2010.

[7] Aleksandar Kircanski. "Analysis of boomerang differential trails via a SATbased constraint solver URSA.", ACNS 15: 13th International Conference on Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 9092, pp. 331-349, June, 2015.

[8] Ling Song, Zhangjie Huang, and Qianqian Yang. "Automatic differential analysis of ARX block ciphers with application to SPECK and LEA.", Cryptology ePrint Archive, Report 2016/209, 2016. http://eprint.iacr.org/2016/ 209.

[9] Mohamed, Mohamed Saied Emam, et al. "Improved algebraic side-channel attack on AES." Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on. IEEE, pp. 146-151, Jun, 2012.

[10] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisability modulo theories. Handbook of satisability, vol. 185, pp. 825-885, 2009.

[11] Helger Lipmaa and Shiho Moriai. "Efficient algorithms for computing differential properties of addition.", FSE 2001, Lecture Notes in Computer Science, vol 2355, pp. 336 - 350, April, 2002.

[12] Seojin Kim et al. "Efficient Differential

Trail Searching Algorithm for ARX Block Ciphers." Jouranl of The Korea Institute of Information Security & Cryptology 26(6), pp. 1421-1430, Dec, 2016

〔13〕 Alex Biryukov, Vesselin Velichkov, and Yann Le Corre. "Automatic search for the best trails in ARX: Application to block cipher speck.", FSE 2016, Lecture Notes in Computer Science, vol. 9783, pp. 289 - 310, March, 2016.

〔14〕 Itai Dinur. "Improved differential cryptanalysis of round-reduced Speck.", SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography, Lecture Notes in Computer Science, vol. 8781, pp. 147 - 164, August, 2014.

〔15〕 Abed, Farzaneh, et al. "Differential cryptanalysis of round-reduced simon and speck." International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, pp. 525-545, 2014.

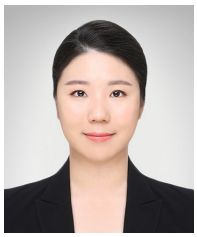
〔16〕 https://www.msoos.org/cryptominisat4

〔17〕 Y. Liu, G. D. Witte, A. Ranea and T. Ashur. "Differential cryptanalysis of round-reduced simon and speck.", IACR Trans. Symm. Cryptol., 2017(3): pp. 24-36, 2017.

〔18〕 G. D. Witte, T. Ashur and Y. Liu, "An Automated Tool for Rotational-XOR Cryptanalysis of ARX-based Primitives", 38th Symp. on Info. Theo. in the Benelux, pp 59-66, 2017.

〈저자소개〉

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



김 서 진 (Seojin Kim) 학생회원
2016년 2월: 고려대학교 수학과 졸업
2016년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 대칭키, 공개키 암호, 해쉬 함수 설계 및 분석

강 형 철 (HyngChul Kang) 학생회원
2010년 2월: 고려대학교 산업시스템정보공학과 학사 졸업
2010년 3월~현재: 고려대학교 정보보호대학원 석박사통합과정
〈관심분야〉 대칭키 암호 설계, 해쉬 함수 분석, 블록 암호 기반 해쉬 모드 분석

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