Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.4, Aug. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.4.569

차분의 상쇄를 이용한 15-라운드 IIoTBC 블록암호에 대한 차분공격*

송 원 우,[1†] 서 재 원,[1] 전 용 진,[2] 김 종 성[3‡]
[1,2,3]국민대학교 (학생, 대학원생, 교수)

Differential Cryptanalysis on 15-Round IIoTBC Block Cipher Utilizing Cancellation of Differences*

Wonwoo Song,[1†] Jaewon Seo,[1] Yongjin Jeon,[2] Jongsung Kim[3‡]
[1,2,3]Kookmin University (Student, Graduate student, Professor)

요 약

64비트 블록암호 IIoTBC는 산업용 IoT 기기의 보안을 목적으로 설계된 암호 알고리즘으로써 128비트의 비밀키를 사용한다. IIoTBC는 IoT에 사용되는 MCU 크기가 8비트인지 16비트인지에 따라 암호화 방식이 달라진다. 본 논문에서는 MCU의 크기가 8비트인 경우 IIoTBC에 대한 차분공격을 다룬다. IIoTBC의 14-라운드의 차분특성을 이용하여 전체 32-라운드 중 15-라운드를 공격한다. 이때 필요한 선택평문과 암호화 연산은 각각 2^{57}과 $2^{122.4}$이다. 본 논문에서 제시한 차분특성은 기존 13-라운드 불능차분 특성보다 긴 라운드를 가지며, 이를 이용한 공격은 IIoTBC에 대한 첫 키복구 공격 결과이다.

ABSTRACT

The 64-bit block cipher IIoTBC is an encryption algorithm designed for the security of industrial IoT devices and uses an 128-bit secret key. The IIoTBC's encryption algorithm varies depending on whether the MCU size used in IoT is 8-bit or 16-bit. This paper deals with a differential attack on IIoTBC when the MCU size is 8-bit. It attacks 15-round out of the entire 32-round using IIoTBC's 14-round differential characteristic. At this time, the number of required plaintexts and encryption are $2^{57}$and $2^{122.4}$, respectively. The differential characteristic presented in this paper has a longer round than the existing 13-round impossible differential characteristic, and the attack using this is the result of the first key recovery attack on IIoTBC.

Keywords: Cryptanalysis, Differential attack, IoT, Lightweight cryptography

I. 서 론

IoT 통신 기술이 발달하면서 고사양 기기뿐만 아니라 저사양 기기를 사용하는 통신도 함께 증가하고 있다. 안전한 통신을 위해서는 메시지 암호화를 위한 블록암호가 필요하다. 그러나 고사양 기기에 적합하게 제작된 블록암호를 통신과 계산 기능이 제약된 저사양 기기에 그대로 사용하기에는 어려움이 따른다[1].

Received(03. 18. 2024), Modified(06. 13. 2024), Accepted(06. 13. 2024)
* 본 논문은 2023년 동계 학술대회에 발표한 우수논문을 개선 및 확장한 것임
* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원 사업의 연구결과로 수행되었음(IITP-2024-RS-2022-00164800)
† 주저자, sww1129@kookmin.ac.kr
‡ 교신저자, jskim@kookmin.ac.kr(Corresponding author)

따라서 다양한 경량 암호기술이 등장하게 되었는데 블록암호 IIoTBC[2] 역시 이에 속한다. IIoTBC는 산업용 IoT 기기의 보안을 목적으로 설계된 암호 알고리즘으로 128비트 키를 사용하는 64비트 블록암호이다. IIoTBC는 효율적인 ASIC구현을 위해 4×4 S-box 논리 게이트 식의 전처리 방법과 불능차분, 부채널, 대수적 관점에서의 안전성 평가를 제공한다. 하지만 해당 논문에서는 일반적인 차분공격에 대한 결과를 제시하지 않으므로 이에 대한 분석이 필요하다.

1990년대에 Biham과 Shamir에 의하여 고안된 차분공격(differential cryptanalysis)[3]은 블록암호의 입출력차분을 사용하여 공격하는 방법이다. 차분공격은 특정한 입출력차분을 갖는 차분특성을 분석하여 해당하는 차분경로를 만족하는 평문쌍을 이용하는 공격이다.

본 논문에서는 14-라운드 차분특성을 이용하여 15-라운드 차분공격을 진행한다. 본 논문에서 제시한 14-라운드 차분 특성은 [2]에서 제시한 13-라운드 불능차분 특성보다 더 많은 라운드를 갖는다. 이러한 차분특성을 사용하여 15번째 라운드의 키 일부를 찾고 최종적으로 마스터키를 찾는 과정을 제시한다. 이때 필요한 선택평문과 암호화 연산은 각각 2^{57}과 $2^{122.4}$이고 본 논문은 IIoTBC에 대한 최초의 키복구 공격을 제안한다.

II. 표기 및 기호

비트는 0 또는 1의 값을 갖는 이진 값이며 n비트는 0부터 2^n-1의 값을 갖는 이진 값이다. 본 논문에서 1워드는 8비트를 의미하고 논문에서 사용하는 표기 및 기호는 Table 1.와 같다.

Table 1. Notations

Notation	Description
?	Unknown $1-bit$ difference
reg_i	The i-th round key
key_i^j	The j-th subkey of the i-th round
H	The upper $4-bit$ of a word
L	The lower $4-bit$ of a word
$\lll n$	$n-bit$ left rotation
$\ggg n$	$n-bit$ right rotation
$c_i[j]$	The j-th word of the i-round ciphertext divided into 8 parts $j\in\{0,1,...7\}$

III. IIoTBC 블록암호

IIoTBC 블록암호는 총 32-라운드의 GFN 구조로 블록 크기는 64비트, 키 크기는 128비트이다. IIoTBC 블록암호는 64비트 평문이 8개의 워드로 나뉘어 암호화가 진행된다. 각 워드는 8개의 브랜치에 입력으로 들어가며 홀수번째 브랜치의 입력은 f_1 함수를 거친 후 짝수번째 브랜치의 입력과 XOR된다. 연산된 결과는 홀수, 짝수 라운드에 따라 $PA1$ 또는$PA2$ 함수로 permutation되고 Fig. 1.는 $PA1, PA2$ 함수와 이를 통한 2-라운드 IIoTBC 암호화 과정을 묘사한다.

암호화가 시작되면, M_{state}는 평문으로 초기화가 된다. M_{state}는 암호화가 진행됨에 따라 값이 바뀌며, 최종적으로 암호문이 된다. 아래는 i번째 라운드에서 M_{state}가 암호화되는 과정이다. M_{state}는 64비트로 8개의 워드$(M_0, M_1, \dots, M_6, M_7)$로 나뉘어 암호화가 진행된다.

$$M_0' = M_0;\ M_1' = f_1(M_0, key_i^1) \oplus M_1;$$
$$M_2' = M_2;\ M_3' = f_1(M_2, key_i^2) \oplus M_3;$$
$$M_4' = M_4;\ M_5' = f_1(M_4, key_i^3) \oplus M_5;$$
$$M_6' = M_6;\ M_7' = f_1(M_6, key_i^4) \oplus M_7;$$

$$\text{if}\,(i \bmod 2 \neq 0)$$
$$M_{state} = PA1(M_0', M_1', \dots, M_6', M_7');$$

$$\text{if}\,(i \bmod 2 = 0)$$
$$M_{state} = PA2(M_0', M_1', \dots, M_6', M_7').$$

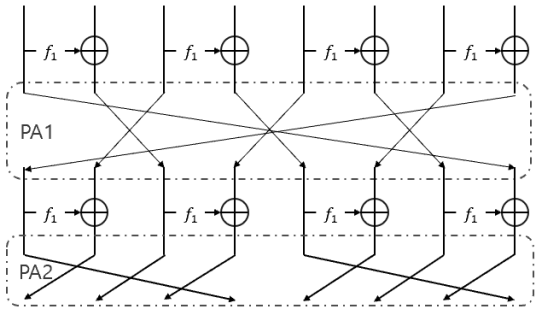

Fig. 1. 2-round encryption of IIoTBC

f_1함수는 AddRoundKey, S-box permutation, 1비트 로테이션으로 구성되며, Fig. 2.에 묘사되어있다. AddRoundKey는 8비트 부분 라운드키와의 XOR 연산과정이며 S-box permutation에서 사용하는 S-box는 Fig. 3.과 같다.

IIoTBC는 128비트의 마스터키를 4등분하여 각각 32비트의 reg_1,reg_2,reg_3,reg_4를 생성 후 이를 통해 나머지 라운드키를 생성한다.

$$reg_1 = k_0 \| k_1 \ldots k_{30} \| k_{31}, \quad (1)$$

$$reg_2 = k_{32} \| k_{33} \ldots k_{62} \| k_{63}, \quad (2)$$

$$reg_3 = k_{64} \| k_{65} \ldots k_{94} \| k_{95}, \quad (3)$$

$$reg_4 = k_{96} \| k_{97} \ldots k_{126} \| k_{127}. \quad (4)$$

$reg_{i(=5,\ldots,32)}$는 reg_{i-4}와 reg_i를 통해 생성되며 이때 f_3함수와 XOR연산이 사용된다. 이를 수식으로 나타내면 다음과 같다.

$$reg_i = f_3(reg_{i-4}) \oplus reg_{i-1}, 5 \le i \le 32. \quad (5)$$

f_3 함수는 32비트의 라운드키를 비트이동 연산 후 k_0,k_2,k_4,k_6과 k_1,k_3,k_5,k_7 각각 4비트를 S-box로 치환하는 함수이다. 32비트 reg_i는 8비트의 서브키 $key_i^{j(=1,2,3,4)}$로 나뉘어 각 라운드 4개의 f_1함수에 적용된다.

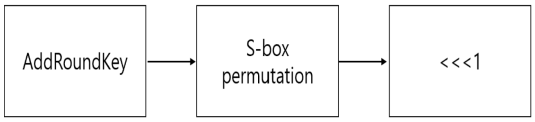

Fig. 2. f_1 function

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

Fig. 3. 4×4 S-box

k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7
k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}	k_{14}	k_{15}
k_{16}	k_{17}	k_{18}	k_{19}	k_{20}	k_{21}	k_{22}	k_{23}
k_{24}	k_{25}	k_{26}	k_{27}	k_{28}	k_{29}	k_{30}	k_{31}

⇩

k'_0	k_8	k_{16}	k_{24}	k'_1	k_9	k_{17}	k_{25}
k'_2	k_{10}	k_{18}	k_{26}	k'_3	k_{11}	k_{19}	k_{27}
k'_4	k_{12}	k_{20}	k_{28}	k'_5	k_{13}	k_{21}	k_{29}
k'_6	k_{14}	k_{22}	k_{30}	k'_7	k_{15}	k_{23}	k_{31}

Fig. 4. f_3 function

IV. IIoTBC에 대한 14-라운드 차분특성 구성

본 절에서는 IIoTBC에 대한 14-라운드 차분특성을 소개한다. 차분특성을 구성할 때 차분이 상쇄되는 성질을 이용하였는데 이는 입력차분이 04일 때 f_1함수를 거친 출력차분이 2^{-3}의 확률로 04로 같은 성질을 말한다. 입력차분 04는 4비트씩 나뉘어지는데 입력차분 0은 1의 확률로 출력차분이 0이고 입력차분 4는 2^{-3}의 확률로 출력차분이 2이다. 출력차분 2는 1비트 왼쪽 로테이션 연산을 거치면 4가 되므로 2^{-3}의 확률로 입력차분과 출력차분이 상쇄된다. Fig. 5.는 차분이 상쇄되는 과정을 묘사한 것이며 Δ가 붙은 것은 차분값을 의미한다.

입력에 0이 아닌 차분이 있는 S-box를 활성 S-box라고 하면 차분이 상쇄되는 성질만을 이용하므로 높은 확률의 차분특성을 구성하기 위해 활성 S-box를 최소화 해야한다.

차분이 상쇄되는 성질은 참/거짓 논리로서 접근할 수 있으므로 이를 1비트로 참/거짓을 표현하여 문제를 접근했다. 각 브랜치에 대하여 차분이 존재하는 경우와 존재하지 않는 2가지 경우가 있으므로 총 256가지 경우에 대하여 14-라운드의 차분특성을 탐색하였다(Fig. 6.) 알고리즘에 의해 나온 14-라운드

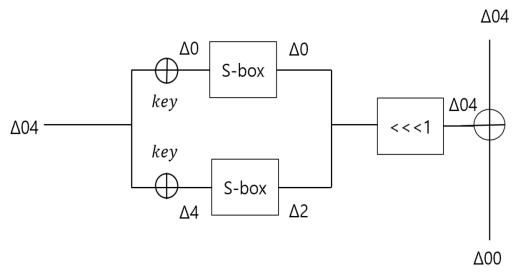

Fig. 5. Property of difference cancellation 04

Algorithm 1: Algorithm to Find the Minimum Number of Active S-boxes Among Differential Characteristics using Differential Property

```
Input: round
Output: Minimun number of active S-boxes among differential paths using Fig.5.
Set active = 0
for 0 ≤ i < 256 do
    (M0,M1,M2,M3,M4,M5,M6,M7) ← i
    for 1 ≤ j ≤ round do
        for 0 ≤ k < 4 do
            if (Mi×2 == 1 and Mi×2+1 == 1)
                Mi×2+1 = 0
                active = active + 1
            else if (Mi×2 == 1 and Mi×2+1 == 0)
                Mi×2+1 = 1
                active = active + 1
        if i % 2 ≠ 0 then
            PA1(i)
        else
            PA2(i)
    printf("input = %d, active = %d\n", j, active);
return active
```

Fig. 6. Algorithm to find the minimum number of active S-boxes among differential characteristics using differential property

의 최소 활성 S-box는 18개이며 2^{-3}의 확률로 차분이 상쇄되므로 $(2^{-3})^{18}=2^{-54}$의 확률로 14-라운드 차분특성을 구성하였다. 이는 Table 2.와 같다.

V. 15-라운드 IIoTBC에 대한 차분공격

5절에서는 15-라운드의 IIoTBC 차분공격을 소개한다. 앞 절에서 소개한 확률 2^{-54}의 14-라운드 차분특성 이후 키복구를 위한 한 라운드를 더 진행하여 최종적으로 15번째 라운드 키를 복구한다.

이후 복구한 키, 나머지 15번째 라운드 키와 12~14번째 라운드키를 이용하여 IIoTBC의 키스케줄링을 통해 최종적으로 마스터키를 복구하는 과정을 소개한다.

5.1 차분 설정

14-라운드 차분특성에 한 라운드를 더 진행한 15번째 라운드 암호문의 출력차분은 (α,04,00,00,β,04,04,00)이다. Fig. 7.는 이를 표현한 그림이다.

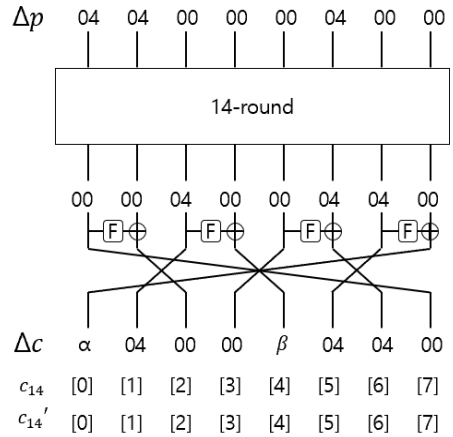

Fig. 7. 15-round differential path

Table 2. 14-round differential characteristic

	ΔM_0	ΔM_1	ΔM_2	ΔM_3	ΔM_4	ΔM_5	ΔM_6	ΔM_7	P
Δr_0	(0,4)	(0,4)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	2^{-3}
Δr_1	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,4)	2^{-3}
Δr_2	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	1
Δr_3	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	2^{-3}
Δr_4	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,4)	(0,0)	2^{-3}
Δr_5	(0,4)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	2^{-6}
Δr_6	(0,4)	(0,0)	(0,0)	(0,4)	(0,4)	(0,4)	(0,4)	(0,0)	2^{-9}
Δr_7	(0,4)	(0,0)	(0,4)	(0,4)	(0,4)	(0,4)	(0,0)	(0,4)	2^{-9}
Δr_8	(0,4)	(0,4)	(0,0)	(0,4)	(0,0)	(0,0)	(0,4)	(0,4)	2^{-6}
Δr_9	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,4)	(0,0)	(0,4)	2^{-3}
Δr_{10}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,4)	2^{-3}
Δr_{11}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	1
Δr_{12}	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	(0,0)	2^{-3}
Δr_{13}	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	(0,4)	(0,0)	2^{-3}
Δr_{14}	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	(0,4)	(0,4)	(0,0)	

5.2 데이터 수집

다음과 같은 과정으로 α와 β의 8비트 값 중 4비트의 값을 알 수 있다. Fig. 8.은 α가 계산되는 과정이며, 키를 제외한 모든 값은 차분이다.

S-box의 입력차분이 0이면 출력차분은 모두 0이므로 α의 8비트 중 출력차분 0이 $\langle\langle\langle 1$ 로테이션 연산된 4비트는 0이다. 따라서 000????0의 형태를 갖는다.

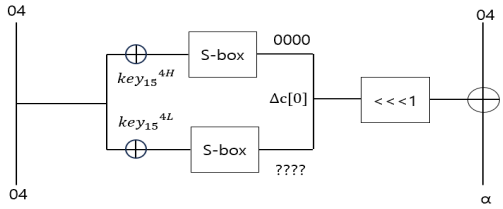

Fig. 8. Calculation process of α

5.3 평문쌍 필터링 단계

Table 2.에 14-라운드 차분특성을 따르지 않아 해당 차분특성과 같은 입력차분을 갖지만 출력차분을 만족하지 않는 쌍이 있을 수 있으므로 본 절에서 이를 제외하는 과정을 다룬다. 이를 필터링 단계라고 한다.

우선 특정한 입력차분을 갖는 평문쌍 2^{56}개 있을 때 14-라운드 차분특성을 만족 할 확률은 2^{-54}이므로 이를 만족하는 $2^{56}\times 2^{-54}=$ 4개의 평문쌍을 찾을 수 있다.

다음은 14-라운드 차분특성 만족하지 않고 차분특성의 출력차분을 만족하는 쌍의 기댓값을 구해본다. 차분특성을 만족하지 않는 암호문쌍의 출력차분은 랜덤하다고 가정하자. α와 β의 각각 4비트와 암호문쌍의 나머지 48비트의 차분으로 암호문 쌍의 56비트 차분값을 알 수 있다.

S-box 특성에 의하여 입력차분이 4일 때의 출력차분인 α와 β는 $2,3,5,6,A,C,E$ 만 가능하므로 α와 β가 차분특성의 출력차분을 만족할 확률이 각각 7/16임을 알 수 있다. 따라서 56비트의 차분값과 α와 β의 각각 나머지 4비트의 확률을 토대로 계산하면 다음과 같다.

$$2^{56}\times(2^{54}-1)/2^{54}\times(1/2)^{56}\times(7/16)^2<1. \quad (6)$$

따라서 이러한 쌍의 기댓값은 1보다 작으므로 고려하지 않는다.

5.4 부분키 복구 단계

이제 필터링된 평문과 암호문쌍으로 키의 일부를 추측한다. Fig. 8.에서 보듯이 α의 8비트 중 0이 아닌 나머지 4비트는 아래 S-box에서 출력된다.

이를 식으로 작성하면 다음과 같다.

$$S(c_{14}[5]^L\oplus key_{15}^{4L})\oplus S(c_{14'}[5]^L\oplus key_{15}^{4L}). \quad (7)$$

$c_{14}[5]\oplus c_{14'}[5]=04$ 이므로 $c_{14'}[5]$는 $c_{14}[5]\oplus 04$로 표현이 가능하다. α 또한 $c_{14}[0]\oplus c_{14'}[0]$로 표현이 가능하므로 $c_{14}[0]\oplus c_{14'}[0]$를 00과 XOR 연산 후 $\rangle\rangle\rangle 1$ 로테이션 연산한 값을 $\Delta c[0]$라고 하자. 상위 4비트인 $\Delta c[0]^H$는 0이므로 고려하지 않고 하위 4비트인 $\Delta c[0]^L$은 다음 수식을 만족한다.

$$S[c_{14}[5]^L\oplus key_{15}^{4L}]\oplus S[c_{14'}[5]^L\oplus key_{15}^{4L}\oplus 4]=\Delta c[0]^L. \quad (8)$$

이를 통해 키를 추측하는 과정은 다음과 같다. 암호문의 차분을 통해 $\Delta c[0]^L$ 값을 알 수 있으므로 입력차분이 $\Delta 4$이고 출력차분이 $\Delta c[0]^L$인 S-box의 출력쌍 (γ_0,γ_1)을 찾고 $S[c_{14}[5]^L\oplus key_{15}^{4L}]=\gamma_0$이거나 $S[c_{14}[5]^L\oplus key_{15}^{4L}]=\gamma_1$를 만족하는 key_{15}^{4L}값을 찾는다.

S-box의 입력차분이 4인 경우 나올 수 있는 출력차분 $2,3,5,6,C,E$는 각각 2/16의 확률로 A는 4/16확률로 나온다. 따라서 출력차분이 A인 경우는 (γ_0,γ_1)쌍이 1쌍이 아닌 2쌍 나오므로 key_{15}^{4L}의 후보가 4개가 나온다.

따라서 평균적으로 나오는 키 key_{15}^{4L} 후보 개수를 계산하면 평균 $2\times 6/7+4\times 1/7\approx 2.3$개의 후보가 나온다. 출력차분이 $\Delta\beta$일 때도 이와 동일하므로 $reg_{15}=(key_{15}^1,key_{15}^2,key_{15}^3,key_{15}^4)$중 $key_{15}^{2L},key_{15}^{4L}$ 즉 8비트를 $2.3^2\approx 2^{2.4}$의 연산으로 찾을 수 있다. 이후 reg_{15}의 나머지 24비트를 전수조사하면 $2^{26.4}$의 연산으로 reg_{15}전체를 구할 수 있다.

5.5 나머지 키복구 단계

IIoTBC의 키스케줄링은 다음과 같다.

$$reg_i = f_3(reg_{i-4}) \oplus reg_{i-1}. \quad (9)$$

5.4에서 reg_{15}를 구했으므로 reg_{14}를 전수조사하면 식 (9)을 이용하여 reg_{11}을 알 수 있다. reg_{13}, reg_{12}도 전수조사를 통해 조사하면 결국 $reg_9, reg_{10}, reg_{11}, reg_{12}, reg_{13}, reg_{14}$ 연속된 6개의 reg를 알 수 있다. IIoTBC의 키스케줄링 구조상 연속된 5개의 reg를 알면 모든 reg를 알 수 있으므로 $2^{26.4} \times 2^{32} \times 2^{32} \times 2^{32} = 2^{122.4}$의 연산으로 마스터키인 $reg_1, reg_2, reg_3, reg_4$를 구할 수 있다.

VI. 결 론

본 논문에서는 14-라운드 차분특성을 이용한 15-라운드 차분공격을 소개했다. 2^{-3}의 확률로 차분이 상쇄되는 성질과 이러한 성질을 갖으며 활성 S-box의 개수가 최소인 차분특성을 구성하였다. 차분공격을 이용하여 2^{57}개의 선택평문과 $2^{122.4}$의 시간 복잡도를 갖고 IIoTBC의 마스터키를 복구하였다. 본 논문에서의 결과는 MILP 등의 자동화 도구를 통한 차분특성의 검색으로 향상될 여지가 존재한다. 본 논문에서 제시한 차분특성은 기존 13-라운드 불능차분특성보다 긴 라운드를 가지며, 이를 이용한 공격은 IIoTBC에 대한 첫 키복구 공격 결과이다. 이러한 결과는 차분공격 관점에서 IIoTBC의 안전성에 대한 평가를 제공한다.

References

[1] S. Moon, M. Kim, T. Kwon, "Trends in Lightweight Crypto Technology for IoT Communication Environments," The Journal of The Korean Institute of Communication Sciences, 33(3), pp. 80-86, Feb. 2016.

[2] J. Kuang, Y. Guo and L. Li, "IIoTBC: A Lightweight Block Cipher for Industrial IoT Security," KSII Transactions on Internet & Information Systems, 17(1), pp. 97-119, Jan. 2023.

[3] E. Biham and A. Shamir, "Differential crypta nalysis of DES-like cryptosystems," Journal of CRYPTOLOGY, vol. 4, no. 1, pp. 3-72, Jan. 1991.

〈저자소개〉

송 원 우 (Wonwoo Song) 학생회원
2020년 3월~현재: 국민대학교 정보보안암호수학과 재학 중
〈관심분야〉 정보보호, 암호 알고리즘

서 재 원 (Jaewon Seo) 학생회원
2022년 3월~현재: 국민대학교 정보보안암호수학과 재학 중
〈관심분야〉 정보보호, 암호 알고리즘

전 용 진 (Yongjin Jeon) 학생회원
2018년 8월: 국민대학교 정보보안암호수학과 졸업
2020년 8월: 국민대학교 금융정보보안학과 석사
2020년 9월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 암호 알고리즘

김 종 성 (Jongsung Kim) 종신회원
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 정보보호 공학박사
2007년 2월: 고려대학교 정보보호대학원 공학박사
2009년 9월~2013년 2월: 경남대학교 e-비즈니스학과 교수
2013년 9월~2017년 2월: 국민대학교 수학과 교수
2017년 3월~현재: 국민대학교 정보보안암호수학과/금융정보보안학과 교수
〈관심분야〉 정보보호, 암호 알고리즘, 디지털 포렌식