Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.4, Aug. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.4.607

GIFT-128에 대한 SITM 공격: NIST 경량암호 최종 후보 GIFT-COFB 적용 방안 연구*

박 종 현,[1†] 김 한 기,[1] 김 종 성[2‡]
[1,2]국민대학교 (대학원생, 교수)

SITM Attacks on GIFT-128: Application to NIST Lightweight Cryptography Finalist GIFT-COFB*

Jonghyun Park,[1†] Hangi Kim,[1] Jongsung Kim[2‡]
[1,2]Kookmin University (Graduate student, Professor)

요 약

SITM (See-In-The-Middle) 공격은 부채널 정보를 활용한 차분 분석 기법 중 하나로, CHES 2020에서 제안되었다. 이 기법은 부분적으로 부채널 마스킹이 적용된 블록암호에서 부채널 마스킹이 적용되지 않은 중간 라운드의 전력 파형을 이용해 차분 분석을 진행한다. 블록암호 GIFT는 CHES 2017에 제안된 경량암호로, 블록암호 PRESENT에서 발견된 취약점을 보완하고 더욱 효율적인 구현이 가능하도록 설계되었다. 본 논문에서는 부분 마스킹이 적용된 GIFT-128에 대한 SITM 공격을 제안한다. 이 공격은 4-라운드와 6-라운드 부분 마스킹이 적용된 GIFT-128을 공격대상으로 하며, 공격에 필요한 시간/데이터 복잡도는 각각 $2^{14.01}/2^{14.01}$, $2^{16}/2^{16}$ 이다. 본 논문에서는 SITM 공격에서 사용 가능한 마스터키 복구 논리를 비교하여, 상황에 따라 더욱 효율적인 논리를 선택하는 기준을 성립한다. 마지막으로, NIST 표준 경량암호 공모사업 최종 후보 중 하나인 GIFT-COFB에 해당 공격을 적용하는 방안을 제시한다.

ABSTRACT

The SITM (See-In-The-Middle) proposed in CHES 2020 is a methodology for side-channel assisted differential cryptanalysis. This technique analyzes the power traces of unmasked middle rounds in partial masked SPN block cipher implementation, and performs differential analysis with the side channel information. Blockcipher GIFT is a lightweight block cipher proposed in CHES 2017, designed to correct the well-known weaknesses of block cipher PRESENT and provide the efficient implementation. In this paper, we propose SITM attacks on partial masked implementation of GIFT-128. This attack targets 4-round and 6-round masked implementation of GIFT-128 and time/data complexity is $2^{14.01}/2^{14.01}$, $2^{16}/2^{16}$. In this paper, we compare the masterkey recovery logic available in SITM attacks, establishing a criterion for selecting more efficient logic depending on the situation. Finally, We introduce how to apply the this attack to GIFT-COFB, one of the finalist candidates in NIST lightweight cryptography standardization process.

Keywords: Differential Cryptanalysis, Side-Channel Analysis, SITM, GIFT, GIFT-COFB

Received(04. 25. 2022), Modified(06. 22. 2022), Accepted(06. 22. 2022)

* 본 연구는 고려대 암호기술 특화연구센터(UD210027XD)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.

† 주저자, mmo330@kookmin.ac.kr
‡ 교신저자, jskim@kookmin.ac.kr(Corresponding author)

I. 서 론

과학기술의 발전으로 인해 IoT 기기의 수요는 점차 증가하고 있다. IoT 기기는 사용자의 편리성을 고려하여 소형화되고 있고 이는 기기에 사용 가능한 공간이 한정되게 된다. 경량 암호는 이처럼 한정된 자원을 요구하는 IoT 환경에 사용되기 적합하다.

CHES 2017에 제안된 블록암호 GIFT[1]는 경량 암호로, 많은 안전성과 구현 효율성 분석 연구 결과들이 제안되었다[2, 3, 4]. 미국 국립표준기술연구소 NIST에서는 경량 암호를 표준화하기 위한 공모사업이 진행되고 있다. 제출된 후보 중에는 GIFT가 핵심 함수로 사용되는 SUNDAE-GIFT[5], HyENA[6], GIFT-COFB[7]가 제안되었고 최종 후보에는 GIFT-COFB가 있다.

차분 분석(Differential Cryptanalysis, DC)은 암호분석 기법의 일종으로, 암호의 안전성을 평가하거나 비밀키를 획득하기 위해 사용되는 기법이다[8]. 이 기법은 특정한 비트에 반전 관계를 갖는 평문 쌍을 암호화하여 생성된 암호문 쌍을 분석해 비밀키를 획득한다.

부채널 분석(Side-Channel Analysis, SCA)은 물리적인 분석 기법으로, 암호 알고리즘이 탑재된 기기에서 발생하는 전력 소모량, 전자파, 연산 소요시간 등의 부가적인 정보들을 이용하여 비밀키를 구한다[9]. 부채널 분석의 대응법으로는 마스킹 기법[10]이 있다.

SCADPA (Side-Channel Assisted Differential Plaintext Attack)는 부채널 정보를 활용한 차분 분석 기법으로, bit-permutation 기반의 블록암호를 공격대상으로 한다[11]. CHES 2020에는 공격대상이 SPN 구조의 블록암호로 확장된 SITM (See-In-The-Middle)이 제안되었다[12]. 이 공격은 SNR (Signal-to-Noise Ratio)이 낮은 혹독한 환경에서도 공격이 가능한 분석으로 제안되었다. 제안된 공격은 부분 마스킹 구현된 블록암호를 대상으로 마스킹 되지 않은 중간 라운드의 전력 파형을 통해 차분 유무를 판단하고 이 정보를 이용하여 차분 공격을 한다. 이를 이용하여, 부분 1차 또는 고차 마스킹 기법을 적용하기 위한 타당한 마스킹 라운드 수를 측정할 수 있고 이는 마스킹 구현으로 발생하는 비용을 줄일 수 있다.

본 논문에서는 블록암호 GIFT-128의 차분 경로를 제시하고 이를 통해 LUT (Look Up Table)로 구현된 GIFT-128에 대한 SITM 공격을 제시한다. 이 공격은 평문을 수집하는 알고리즘과 수집한 평문을 사용해 키 복구를 하는 알고리즘으로 구성되며, 실험적으로 테스트한 결과를 제시한다. 실험 결과로 필요한 평균 데이터 개수와 공격 성공률을 알 수 있다. 최종적으로는 제안하는 공격을 기반으로 NIST 표준 경량암호 공모사업 최종 후보 중 하나인 GIFT-COFB에 적용하는 방안을 소개한다.

본 논문의 구성은 다음과 같다. 2장은 GIFT-128, GIFT-COFB의 알고리즘과 SITM의 개요를 설명한다. 3장에서는 본 논문의 공격에서 활용할 GIFT-128의 차분 경로를 소개한다. 4장은 차분 경로를 이용한 SITM 공격을 설명하고 이를 GIFT-COFB에 적용해 키 복구 공격을 하는 방안을 제시한다. 마지막으로 5장은 결론 및 향후의 계획으로 마무리한다.

II. 배경지식

2.1 GIFT 알고리즘

GIFT는 PRESENT[13]의 알려진 취약점[14, 15]들을 개선하면서 속도와 용량까지 더욱 효율적으로 설계된 블록암호이다. 사용되는 키 크기는 128-비트이고 블록 크기 64-비트를 사용하는 GIFT-64와 128-비트를 사용하는 GIFT-128 두 가지가 있다. GIFT-128의 전체 라운드는 40-라운드이며, 라운드 함수는 다음과 같다.

- SubCells: 128-비트 입력값을 4-비트 단위로 S-box 연산한 128-비트 값을 출력, S-box: $\{0,1\}^4 \rightarrow \{0,1\}^4$
- PermBits: 128-비트 입력값을 비트 단위 위치 이동한 128-비트 값 출력
- AddRoundKey: 128-비트 입력값을 키스케줄을 통해 생성된 라운드키 64-비트와 정해진 상수 8-비트로 XOR 연산한 128-비트 값을 출력

Fig. 1.은 GIFT-128의 라운드 함수를 도식화한 것이다. GIFT 알고리즘에는 whitening key가 사용되지 않는다. 본 논문에서는 GIFT-128의 i번째 라운드를 iR로 표기하고 SubCells에서 S-box의 순서는 Fig. 1.과 같다.

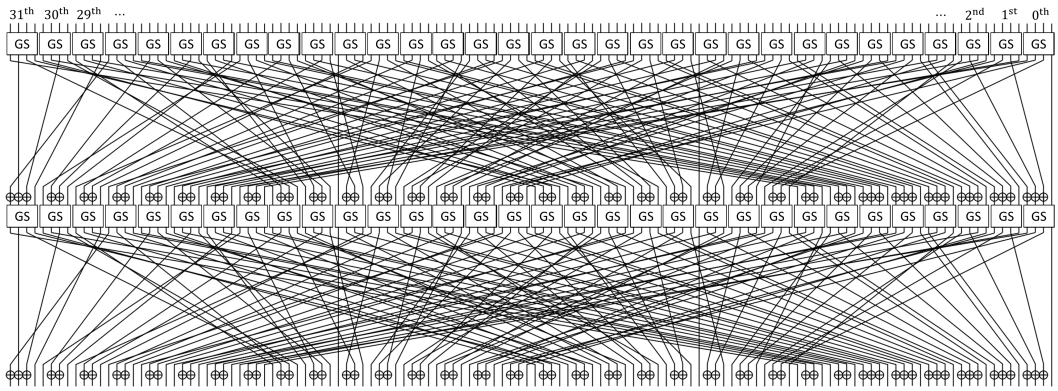

Fig. 1. 2-rounds of GIFT-128(GS is S-box).

128-비트 마스터키 $K = k_7 \| k_6 \| ... \| k_0$라고 하자($k_i$의 크기는 16-비트). GIFT-128의 키스케줄은 다음과 같다.

1) 라운드키 = $k_5 \| k_4 \| k_1 \| k_0$.

2) $k_7 \| ... \| k_1 \| k_0 \leftarrow (k_1 \gg 2) \| (k_0 \gg 12) \| k_7 \| ... \| k_2$,

$a \| b$는 a와 b의 연접을, $X \gg c$는 X를 오른쪽으로 c-비트만큼 회전시킨 것을 의미.

2.2 GIFT-COFB 알고리즘

GIFT-COFB는 NIST 표준 경량암호 공모사업 최종 후보 중 하나로, GIFT-128 기반으로 설계된 AEAD (Authenticated Encryption with Associated Data) 암호이다. GIFT-COFB 암호화 알고리즘의 입력 변수는 다음과 같다.

- K: GIFT-128에 사용하는 128-비트 마스터키
- N: 128-비트 nonce
- A, M: 관련 데이터, 메시지

GIFT-COFB의 출력 변수는 다음과 같다.

- C: 암호문
- T: 태그

GIFT-COFB에 사용되는 함수들은 다음과 같다. 함수의 128-비트 입력값을 X, 왼쪽으로 비트를 회전시키는 것을 $\lll$, $|a|$는 a의 비트 크기로 표기한다. 0^i는 0이 i개만큼 채워진 비트를 의미한다.

- $E_K(X)$: K를 사용하는 GIFT-128로 X를 암호화하여 암호문 출력
- $Trunc_t(X)$: X의 하위 t-비트를 출력
- $G(X)$: $X = X[1] \| X[2]$일 때, $X[2] \| (X[1] \lll 1)$을 연산하여 출력($|X[1]| = |X[2]| = 64$)
- $pad(X)$: 만약 X가 공백이 아니고 128-비트 크기라면, 그대로 X를 출력하고 그 외에는 $X \| 1 \| 0^{128-(|X| mod 128)-1}$을 출력

A(M 또는 C)$[a]$는 대상을 128-비트 블록 크기로 나누었을 때, a번째 블록임을 의미한다. Fig. 2.는 GIFT-COFB의 암호화 알고리즘 예시를 도식화한 것이다. 이 예시에서는 256-비트 A와 M을 암호화하며, 출력하는 태그의 길이는 128-비트이다. 상세한 알고리즘 설명은 [7]을 통해 알 수 있다.

2.3 SITM 개요

SITM 공격은 부채널 정보를 활용한 차분 분석

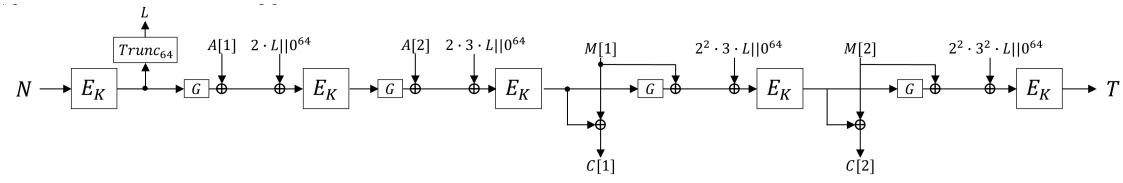

Fig. 2. Example of GIFT-COFB encryption($|M| = |A| = 256$, $|T| = 128$).

기법의 일종으로, SPN 구조의 블록암호를 공격대상으로 한다. 차분 분석은 두 입력값을 XOR 연산한 차분이 라운드 함수를 거치면서 전파되는 차분 특성을 이용하는 암호분석 기법이다. SITM 설명에서는 아래와 같은 용어들이 사용된다.

- 전력 파형: 알고리즘이 탑재된 기기에서 발생하는 전력의 파형
- 차분 파형: 두 전력 파형 T_1, T_2의 차이, $T_1 - T_2$
- 차분 경로: 두 입력값을 XOR 연산한 차분이 암호화 과정을 통해 전파되는 예상 경로
- Active S-box: 차분 경로 내 입력·출력 차분이 0이 아닌 S-box
- Inactive S-box: 차분 경로 내 입력·출력 차분이 0인 S-box
- *Depth*: 공격대상의 전력 파형 관찰을 하는 라운드 위치, ex) *Depth* = 4이면, 암·복호화를 모두 고려하여 공격대상은 6-라운드 마스킹 기법이 적용된다.

해당 공격은 S-box 연산 과정에서 발생하는 전력 파형의 차이를 이용한다. 서로 다른 평문 쌍의 암호화 시 발생한 전력 파형을 수집하였다고 가정하자. 이때, 같은 입력값으로 S-box가 연산 되었다면 발생한 두 전력 파형은 유사할 것이고 서로 다른 입력값이 연산 되었다면 두 전력 파형은 상이할 것으로 예측할 수 있다. 이 논리를 이용해 공격자가 예측한 차분 경로로 암호화가 진행되었는지를 판단할 수 있다. 만약 이를 만족하는 평문 쌍이 수집되었다면, 유효한 차분 특성을 사용해 틀린 키의 후보를 줄일 수 있다. 상세한 키 복구 알고리즘은 4장에서 다룬다.

III. GIFT-128 차분 경로

3.1 3-라운드 GIFT-128 차분 경로

GIFT-128의 PermBits는 확률적으로 active S-box를 4개에서 1개로 감소시킬 수 있다. 3장에서는 이 특성을 사용한 8개 차분 경로 T1~T8을 Fig. 3.에서 소개한다. 소개하는 차분 경로는 2R부터 표기된다. GIFT는 whitening key를 XOR 연산하는 과정이 없으므로 공격자가 원하는 2R의 입력 차분을 확률 1로 구성할 수 있다. 경로마다 2R의 4개 active S-box 위치가 서로 다르며, 입력 차분 (A, B, C, D)는 (13, 12, 14, 3)과 (15, 6, 10, 5)를 이용한다. 각 차분 경로의 확률은 2^{-9}이다.

각각의 GIFT-128 차분 경로는 다음과 같은 조건을 만족한다.

- 3R: 31, 30, …, 25, 24번째 S-box 1개씩 active S-box이고 나머지는 inactive S-box

2R의 active S-box 입력 차분 (A, B, C, D)와 출력 차분을 통해, 실제 S-box 입력값 후보들을 고려할 수 있고 이는 키 복구 알고리즘에 사용된다. 특정 입력 차분과 출력 차분이 성립되게 하는 입력값 쌍은 DDT (Differential Distribution Table) 연산 과정을 통해 알 수 있다[8].

3.2 4-라운드 GIFT-128 차분 경로

GIFT S-box의 입력 차분이 15일 때, 가장 높은 확률로 가능한 출력 차분이 4이다. 이때 확률은 2^{-2}이며, 이 차분 특성을 T1~T8 차분 경로에 적용하면 한 라운드 차분 경로를 연장할 수 있다.

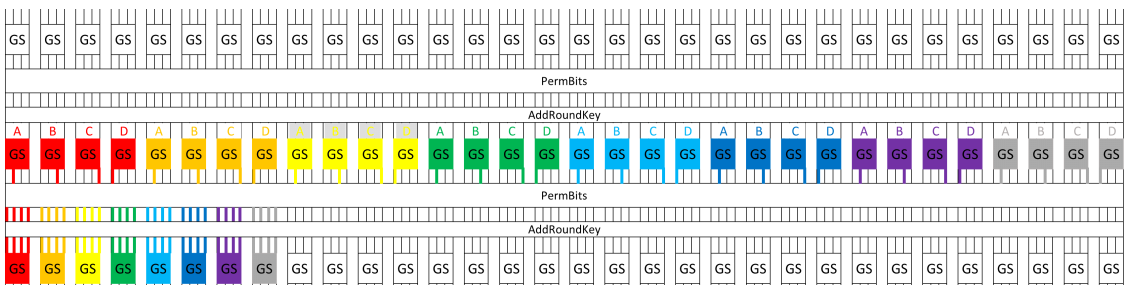

Fig. 3. 8 differential trails of GIFT-128. Paths of different colors are different differential trail. A, B, C and D represent the difference. From left to right, it is a T1 to T8 differential trail.

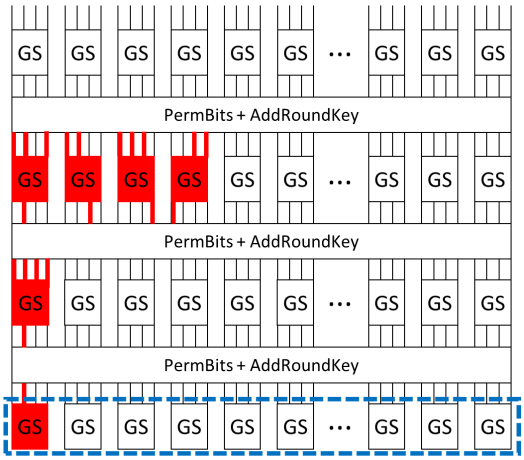

Fig. 4. Differential trail for using $depth = 4$ SITM attack

Fig. 4.는 T1의 차분 경로를 한 라운드 연장한 차분 경로이다. 이와 비슷한 방법으로 T2~T8의 차분 경로를 연장할 수 있다. 각 차분 경로의 확률은 2^{-11}이다.

IV. GIFT-128 SITM 공격

SITM 제안 논문[12]에서는 블록암호 PRESENT를 공격함으로써, bit-permutation 기반의 암호인 RECTANGLE과 GIFT에도 SITM 공격을 적용할 수 있음을 언급하였다. 하지만 구체적인 적용 방법, 차분 경로 그리고 키 복구 내용은 설명하지 않았기 때문에, 본 장에서는 GIFT-128에 SITM 공격을 적용하는 방법을 구체적으로 서술한다. 그리고 이를 통해 GIFT-COFB에 대한 키 복구 공격으로 확장한다.

소개한 GIFT-128 차분 경로를 이용한 SITM 공격 과정은 평문 수집 알고리즘과 키 복구 알고리즘이 있다. 공격 과정에서 각 차분 경로는 독립적으로 사용되며, 모든 경로를 사용하면 1R 라운드키 후보를 약 2^8개로 줄인다. 4장에서는 차분 경로 T1을 사용해 공격 방법을 설명하며, 나머지 경로들도 유사하게 적용할 수 있다.

4.1 평문 수집 단계

차분 경로를 만족하는 **평문 수집 알고리즘**은 아래와 같다.

1) 1R PermBits 출력값 P_1을 랜덤하게 생성한다. 차분 경로의 2R 입력 차분과 P_1을 XOR 연산한 P_2를 생성한다.

2) P_1, P_2를 역 PermBits와 역 SubCells 연산하여 평문 쌍을 구한다.

3) 평문 쌍을 암호화하여 $(depth)$R SubCells에서 발생하는 전력 파형 T_1, T_2를 수집한다.

4) T_1, T_2의 차분 파형을 통해 $(depth)$R에서 31번째만 active S-box임을 판단한다. 만약 이를 만족한다면 P_1을 수집한다. 차분 경로의 확률 특성상 2^9번 시도하면 1번 만족할 것으로 기대할 수 있다.

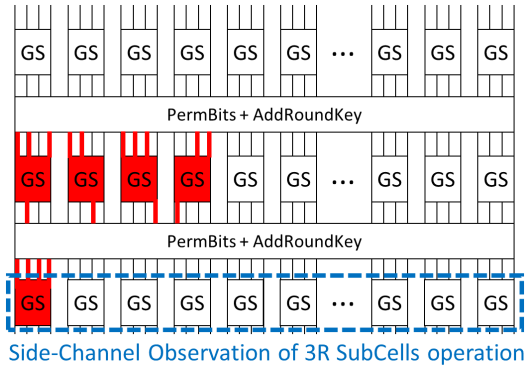

Fig. 5. Step 4 of the algorithm to collect plaintext.

4.2 키 복구 알고리즘

P_1을 수집했다면 키 복구 과정을 진행한다. 키 복구 알고리즘은 다음과 같다.

1) 차분 경로 2R SubCells에서 31, 30, 29, 28번째 S-box의 입력, 출력 차분이 성립하는 입력값 후보들을 사전 연산한다(Inactive S-box 제외).

2) GIFT의 AddRoundKey는 라운드키가 XOR 되지 않는 비트와 고정된 상수가 XOR 연산되는 비트 위치가 존재한다. 이 비트 위치를 고려하

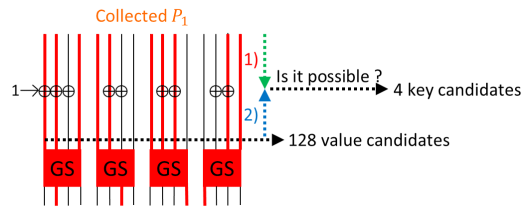

Fig. 6. Step 2 of key recovery algorithm.

면 P_1과 가능한 입력값 후보들을 줄일 수 있다. 이를 만족하는 P_1의 31, 30, 29, 28번째 nibble 값과 입력값 후보를 XOR 연산하여 1-바이트 1R 라운드키 후보로 저장한다.

Table 1.은 평문 수집 알고리즘에서 사용된 평문 쌍의 평균 개수와 키 복구 알고리즘으로 얻은 키 후보 집합에 올바른 키가 있을 확률을 제시한다. 본 실험에서는 평균 개수와 성공률을 구하기 위해 1,000번의 테스트를 진행하였다.

Table 1. For each (A, B, C, D), the experimental results of the algorithm for finding plaintext and the key recovery algorithm.

Depth	(A, B, C, D)	(13, 12, 14, 3)	(15, 6, 10, 5)
3	Data	$2^{11.98}$	$2^{12.02}$
	Prob.	1.0	1.0
4	Data	$2^{14.01}$	$2^{14.03}$
	Prob.	1.0	1.0

4.3 키 후보 집합을 이용한 마스터키 복구 논리 비교

본 절에서는 (A, B, C, D)의 모든 후보를 고려하여 제안하는 공격을 실험한 결과를 소개한다. (A, B, C, D)=(13, 12, 14, 3)일 때, 한 개 차분 경로를 이용한 공격으로 1-바이트 1R 라운드키 후보를 가장 적은 4개로 줄일 수 있었다. (A, B, C, D)=(15, 6, 10, 5)일 때에는 8개로 줄일 수 있었다. 본 절에서는 SITM 제안논문[12]과 향상된 PRESENT에 대한 SITM 공격 논문[16]에서 사용한 마스터키 복구 논리를 제안하는 공격에 각각 적용하여 특징점을 확인하고 비교한다.

4.3.1 단일 키 후보 집합을 이용한 마스터키 복구

SITM 제안논문[12]의 키 복구 논리를 적용한 GIFT-128의 키 복구 과정은 다음과 같다.

1) (A, B, C, D)=(13, 12, 14, 3)으로 설정하고 한 개 차분 경로를 이용한 공격으로 1-바이트 1R 라운드키 후보 4개를 구한다.

2) 2R SubCells 출력값 P_1을 랜덤하게 생성한다. 차분 경로의 2R SubCells 출력 차분과 P_1을 XOR 연산한 P_2를 생성한다.

3) P_1과 P_2를 1)에서 얻은 1-바이트 1R 라운드키 후보 4개를 모두 사용하여 GIFT-128의 입력 평문 위치까지 역연산한다. 이때, 역연산을 위한 나머지 바이트 1R 라운드키는 0으로 설정한다.

4) 연산을 통해 얻은 평문 쌍 4개에 대해서 암호화하여 $(depth)$R SubCells에서 발생하는 전력 파형 T_1, T_2를 수집한다.

5) T_1, T_2의 차분 파형을 통해 $(depth)$R에서 원하는 위치의 Active S-box가 발생했는지를 확인한다. 만약 일치한다면 이때 사용된 1-바이트 1R 라운드키 후보를 올바른 키로 저장한다. 이때, 우연히 잘못된 키가 저장될 확률은 2^{-16}이다.

GIFT-128의 마스터키는 키스케줄 특성상 1, 2R 라운드키를 통해 복구할 수 있다. 2R 라운드키 복구 방법은 다음과 같다. 첫 번째 SITM 공격으로 얻은 1R 라운드키를 사용하면 1R에 대한 부분 암호화를 사전에 진행할 수 있고 이를 통해 depth를 한 라운드 높인 SITM 공격으로 2R 라운드키 후보를 얻을 수 있다.

Depth=3, 4의 GIFT-128 공격 복잡도는 다음과 같다. SITM 공격으로 1R 라운드키를 구하는 $8\times\{2\times(2^9+4)\}\approx 2^{13.01}$의 시간, 데이터 복잡도가 필요하며, 2R 라운드키를 구하는 시간, 데이터 복잡도도 이와 같다. 결과적으로 얻은 1개의 1, 2R 라운드키로 올바른 마스터키를 복구할 수 있다. 공간 복잡도는 128-비트 1, 2R 라운드 키를 저장하는데 필요한 2^5-바이트가 필요하다.

Depth=4, 5의 GIFT-128 공격 복잡도는 다음과 같다. SITM 공격으로 1R 라운드키 후보를 구하는 $8\times\{2\times(2^{11}+4)\}\approx 2^{15}$의 시간, 데이터 복잡도가 필요하고 2R 라운드키 후보 또한 같다. 이후는 depth=3, 4와 같다.

4.3.2 다중 키 후보 집합의 교집합을 이용한 마스터키 복구

향상된 PRESENT에 대한 SITM 공격 논문[16]에서 사용한 키 복구 논리를 적용한 GIFT-128의 키 복구 과정은 다음과 같다.

(A, B, C, D)를 (13, 12, 14, 3)과 (15, 6, 10, 5)를 모두 사용하여 2가지 키 후보들에서 중복되는 키를 선별하면 1-바이트 1R 라운드키 후보를 2개로 줄일 수 있다. 2R 라운드키 복구 방법은 다음과 같다.

첫 번째 SITM 공격으로 얻은 1R 라운드키 후보 2^8개를 사용하면 1R에 대한 부분 암호화를 사전에 진행할 수 있고 이를 통해 depth를 한 라운드 증가시킨 SITM 공격으로 2R 라운드키 후보를 얻을 수 있다. 결과적으로 1, 2R 라운드키 후보를 2^{16}개 얻을 수 있다.

Depth=3, 4의 GIFT-128 공격 복잡도는 다음과 같다. 1R 라운드키 후보를 구하는 첫 번째 SITM 공격으로 $2\times 2^{13}=2^{14}$의 시간, 데이터 복잡도가 필요하고 2R 라운드키 후보를 구하는 두 번째 SITM 공격에서는 1R 라운드키 후보 2^8개를 고려하여 $2\times 2^{13}\times 2^8=2^{22}$의 시간, 데이터 복잡도가 필요하다. 결과적으로 얻은 2^{16}개의 1, 2R 라운드키 후보로 마스터키 후보를 생성할 수 있고 2^{16}번의 전수조사를 통해 올바른 마스터키를 복구할 수 있다. 공간 복잡도는 128-비트 마스터키 후보 2^{16}개를 저장하는데 필요한 2^{20}-바이트가 필요하다.

Depth=4, 5의 GIFT-128 공격 복잡도는 다음과 같다. 1R 라운드키 후보를 구하는 첫 번째 SITM 공격으로 $2\times 2^{15}=2^{16}$의 시간, 데이터 복잡도가 필요하고 2R 라운드키 후보를 구하는 두 번째 SITM 공격에서는 1R 라운드키 후보 2^8개를 고려하여 $2\times 2^{15}\times 2^8=2^{24}$의 시간, 데이터 복잡도가 필요하다. 이후는 depth=3, 4와 같다.

4.3.3 공격 복잡도 비교 및 분석

Table 2.는 [12]과 [16]의 마스터키 복구 논리를 적용했을 때 발생하는 공격 복잡도를 각각 정리하고 있다. 이를 통해 제안하는 GIFT-128의 SITM 공격은 [12]의 마스터키 복구 논리를 사용하는 것이 더욱 효율적임을 알 수 있다. 하지만 라운드키 후보의 개수가 공격 복잡도에 영향을 줄 만큼 크면 [16]의 마스터키 복구 논리를 사용하는 것이 더 효율적일 수 있다. 해당 비교 분석을 통해 다른 차분 경로를 사용하거나 다른 암호를 공격대상으로 하는 SITM 공격이 더욱 효율적인 마스터키 복구 논리를 선택하는 기준을 성립할 수 있다.

Table 2. Our results of the SITM attack complexity on GIFT-128.

Section	*Depth*	Time	Data	Memory
4.3.1	3, 4	$2^{14.01}$	$2^{14.01}$	2^5
	4, 5	2^{16}	2^{16}	2^5
4.3.2	3, 4	$2^{22.03}$	$2^{22.03}$	2^{20}
	4, 5	$2^{24.01}$	$2^{24.01}$	2^{20}

4.4 GIFT-COFB SITM 공격 적용 방안

제안하는 GIFT-COFB의 SITM 공격 가정은 다음과 같다.

- 중복되지 않는 선택 nonce 사용
- LUT로 구현된 GIFT-COFB

위 공격 가정을 만족하는 GIFT-COFB는 소개한 GIFT-128 SITM 공격 적용이 가능하다. 공격자는 nonce 쌍을 선택하여 GIFT-COFB 암호화 알고리즘의 첫 번째 GIFT-128 암호화에서 SITM 공격을 적용해 마스터키를 복구하는 것이 가능하다.

[12]의 마스터키 복구 논리를 적용한 GIFT-COFB의 SITM 공격 복잡도를 정리하면 다음과 같다.

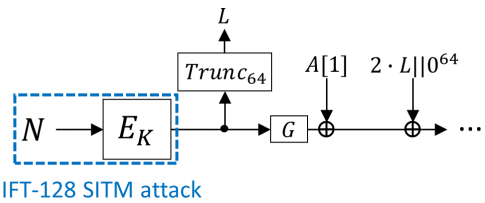

Fig. 7. The area of GIFT-COFB SITM attack.

Table 3. Our results of the SITM attack complexity on GIFT-COFB.

	Depth	Time	Data	Memory
GIFT-COFB	3, 4	$2^{14.01}$	$2^{14.01}$	2^5
	4, 5	2^{16}	2^{16}	2^5

V. 결 론

SITM 공격은 부채널 분석의 도움을 받아 차분 분석을 하는 기법으로, SPN 구조의 블록암호를 공격대상으로 한다. 본 논문에서는 SITM 공격을 블록암호 GIFT-128에 적용했다. 공격은 4-라운드와 6-라운드 마스킹으로 구현된 GIFT-128을 대상으로 하며, 실현 가능한 복잡도를 가진다. 위 공격에 대응하기 위해서는 GIFT-128의 암·복호화를 고려하여 각각 최소 6-라운드와 8-라운드 마스킹을 적용해야 한다. 또한, SITM 공격에서 사용되는 마스터키 복구 논리들의 특징점을 찾고 더욱 효율적인 논리를 선택하기 위한 기준을 성립하였다. 결과적으로 제안한 공격을 NIST 표준 경량암호 공모사업 최종 후보 중 하나인 GIFT-COFB에 적용하는 방법을 제안했다.

SITM 공격은 비교적 최근에 제안된 공격이기 때문에, 현재까지 발표된 암호학적 알고리즘들에 이 공격이 어느 정도로 효과적으로 적용 가능한지 발표된 바가 적다. 따라서, 블록암호 혹은 NIST 표준 경량암호 후보들[17]에 본 공격법이 효과적으로 적용되는지를 연구하는 것은 흥미로운 주제가 될 수 있다.

References

[1] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim and Y. Todo, "GIFT: a small present," International Conference on Cryptographic Hardware and Embedded Systems, LNCS 10529, pp. 321-345, Sep. 2017.

[2] W. Unger, L. Babinkostova, M. Borowczak, and R. Erbes, "Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers," IEEE Computer Society Annual Symposium on VLSI, IEEE, pp. 236-241, Jul. 2021.

[3] R, Zong, X. Dong, H. Chen, Y. Luo, S. Wang, and Z. Li, "Towards Key-recovery-attack friendly distinguishers: Application to GIFT-128," IACR Transactions on Symmetric Cryptology, vol. 2021, no. 1, pp. 156-184, Mar. 2021.

[4] A. Adomnicai, Z. Najm, T. Peyrin, "Fixslicing: a new GIFT representation: fast constant-time implementations of GIFT and GIFT-COFB on ARM Cortex-M," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2020. no. 3, pp. 402-27, Jun, 2020.

[5] SUNDAE-GIFT, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/SUNDAE-GIFT-spec-round2.pdf

[6] HyENA, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/hyena-spec-round2.pdf

[7] GIFT-COFB, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/gift-cofb-spec-final.pdf

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

[9] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," In: Annual international cryptology conference, LNCS 1666, pp. 388-397, Dec. 1999.

[10] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," International conference on information and communications security, LNCS 4307, no. 1, pp. 529 - 545, Dec. 2006.

[11] J. Breier, D. Jap, and S. Bhasin, "SCADPA: Side-channel assisted differential-plaintext attack on bit permutation based ciphers," 2018 Design, Automation & Test in Europe Conference & Exhibition, IEEE, pp. 1129-1134, Mar. 2018.

[12] S. Bhasin, J. Breier, X. Hou, D. Jap, R. Poussier and S. M. Sim, "Sitm: See-in-the-middle side-channel assisted

middle round differential cryptanalysis on spn block ciphers," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2020, no. 1, pp. 95-122, Nov. 2019.
〔13〕 A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, et al., "Present: An ultra-lightweight block cipher," International Workshop on Cryptographic Hardware and Embedded Systems, LNCS 4727, pp. 450-466, Sep. 2007
〔14〕 C. Blondeau and K. Nyberg, "Links between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities," In: EUROCRYPT 2014, LNCS 8441, pp. 165-182, May 2014.
〔15〕 C. Jooyeon, "Linear cryptanalysis of reduced-round PRESENT," In: CT-RSA 2010, LNCS, vol. 5985, pp. 302-317, 2010.
〔16〕 J.H Park, H.G Kim, and J.S Kim, "Improved SITM Attack on the PRESENT Blockcipher," Journal of the Korea Institute of Information Security & Cryptology, 32(2), pp. 155-162, Apr. 2022.
〔17〕 NIST Lightweight Cryptography Standardization: Finalists Announced, https://csrc.nist.gov/News/2021/lightweight-crypto-finalists-announced

〈저자소개〉

박 종 현 (Jonghyun Park) 학생회원
2021년 2월: 국민대학교 정보보안암호수학과 졸업
2021년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 정보보호, 암호 알고리즘

김 한 기 (Hangi Kim) 학생회원
2016년 2월: 국민대학교 수학과 졸업
2018년 2월: 국민대학교 금융정보보안학과 석사
2018년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 암호 알고리즘

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