Journal of the Korea Academia-Industrial cooperation Society
Vol. 23, No. 2 pp. 161-169, 2022

https://doi.org/10.5762/KAIS.2022.23.2.161
ISSN 1975-4701 / eISSN 2288-4688

경량 블록 암호 알고리즘 PIPO에 대한 테이블 기반 마스킹 적용 및 분석

배대현[1], 이재욱[2], 이현로[2], 하재철[2*]
[1]호서대학교 정보보호학과, [2]호서대학교 컴퓨터공학부

Application and Analysis of Table-based Masking Countermeasure for Lightweight Block Cipher PIPO

Daehyeon Bae[1], Jaewook Lee[2], Hyeonro Lee[2], Jaecheol Ha[2*]
[1]Department of Information Security, Hoseo University
[2]Division of Computer Engineering, Hoseo University

요 약 최근 제안된 경량 블록 암호 알고리즘 PIPO는 부채널 분석 공격 대응책 중 하나인 마스킹을 효과적으로 적용하기 위해 비선형 연산의 개수를 최소화하여 설계되었다. PIPO 암호 알고리즘에서는 부채널 공격 취약성이 있는 S-Layer의 비선형 연산에 직접 마스킹을 적용하거나 사전에 계산된 마스킹 테이블을 사용하여 부채널 공격에 대응할 수 있다. 그러나 본 논문에서는 마스킹 기법을 적용해도 커플링 취약점 등으로 인해 의도치 않은 부채널 신호 누출이 발생해 1차 상관 분석 공격에서도 비밀 키가 복구될 수 있음을 확인하였다. 그리고 사전 연산 테이블을 이용한 S-layer 연산에서 2종류의 테이블 기반의 마스킹 방식을 제안하고 이에 대한 안전성 및 효율성을 분석하였다. 1차 상관 전력 분석 공격 내성, 수행 시간 그리고 메모리 측면에서 비교한 결과, 제안하는 방법들은 고전적인 테이블 마스킹 방법보다 많은 클럭 사이클이 소요되지만 기존 방법에 비해 소요되는 메모리가 적고 1차 부채널 분석 공격에 대응하여 비밀 키를 노출시키지 않음을 확인하였다.

Abstract The recently proposed lightweight block cipher algorithm PIPO is designed to minimize the number of nonlinear operations to mask effectively. Moreover, this algorithm is one of the countermeasures against the side-channel analysis. The PIPO cryptographic algorithm can directly apply the masking method to nonlinear operations of vulnerable S-Layer and use a pre-computed masking table to defeat the side-channel analysis. This research confirmed that even if a pre-computed masking table is applied, unintended side-channel leakage occurs due to coupling vulnerabilities and other reasons. The research also confirmed that the secret key can be recovered in the first-order correlation power analysis. In addition, we proposed two types of table-based masking methods of the S-layer operation using a pre-calculation table and analyzed security and efficiency. As a result of comparing resistance and execution time against first-order correlation power analysis, we confirmed that the proposed methods take more clock cycles than the classical masking methods. But, the proposed methods can defeat the first-order side-channel analysis with less memory requirement.

Keywords : Side-Channel Analysis, Power Analysis Attack, Lightweight Block Cipher PIPO, Table Masking, Shuffling Countermeasure

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1074358).
*Corresponding Author : Jaecheol Ha(Hoseo Univ.)
email: jcha@hoseo.edu
Received December 30, 2021 Revised January 25, 2022
Accepted February 4, 2022 Published February 28, 2022

1. 서론

IoT(Internet of Things) 기술이 발달함에 따라 저사양 단말 기기에서도 기밀성을 제공하기 위해 큰 부하없이 사용될 수 있는 암호 알고리즘의 필요성이 대두되고 있다. 이에 따라 수많은 경량 블록 암호 알고리즘이 제안되어 활발히 사용되고 있으며, 대표적으로 SIMON, SPECK[1], HIGHT[2], LEA[3] 등이 있다. 그러나 IoT 기기들은 동작 환경의 특성상 누구나 물리적으로 접근 및 제어가 가능한 경우가 많기에 실행되는 암호 알고리즘에는 부채널 분석 대응책 탑재를 필수적으로 고려해야 한다.

경량 블록 암호 알고리즘 PIPO(Plug-In, Plug-Out)는 2020년에 제안된 대칭 키 방식의 암호 알고리즘으로서 부채널 분석 대응책인 마스킹 기법을 보다 효율적으로 사용할 수 있도록 비선형 연산의 개수를 최소화하는 방향으로 설계되었다[4]. 암・복호화 과정의 모든 연산이 오직 비트별 AND, OR, XOR, NOT 그리고 Rotation으로만 이루어지고, 비선형 계층 연산은 병렬로 처리할 수 있어 기존 경량 블록 암호 알고리즘보다 우수한 성능을 보인다.

그러나 PIPO 암호 알고리즘 역시 부채널 분석(side-channel analysis) 공격에 취약한데 이 공격은 암호 알고리즘이 디바이스에 구현되어 수행되는 동안 발생하는 전력이나 전자기파를 측정하여 내부의 비밀 정보를 복구하는 공격이다. 대부분의 공격은 암호 알고리즘의 비선형 연산 과정에서 누설되는 신호 정보와 비밀 정보와의 상관성을 이용하게 된다. 부채널 분석 공격에 대한 대응책으로 많이 사용되는 것이 마스킹(masking) 기법이다. 마스킹 기법이란 난수를 이용해 암호 알고리즘의 중간 연산 결과값의 노출을 막아 공격자가 모델링할 수 없도록 만드는 대응책이다. 상기한 SIMON, HIGHT, 그리고 LEA와 같은 경량 블록 암호에서도 비선형 연산 부분에서 공격이 이루어지며 대부분 마스킹 기법을 대응책으로 사용하고 있다[5-7]. 그러나 각 알고리즘에서 사용하는 프리미티브에 따라 마스킹을 설계하는 방법은 다르고 구현 효율 역시 크게 차이가 날 수 있다.

경량 블록 암호 알고리즘 PIPO에 대해 부채널 분석 대응책으로 마스킹 대응책을 설계할 경우, 비선형 연산인 AND와 OR 연산시 마스크 값을 적절하게 계산할 수 있도록 하는 것이 중요하다. PIPO를 처음 제안한 논문에서는 마스크 값을 유지하는 방법으로 ISW[8] 방법을 권고하고 있지만, 이 방법은 커플링 취약점 등으로 인해 낮은 차수의 부채널 공격으로도 비밀 키가 쉽게 누출되는 경우가 발생한다[9].

본 논문에서는 이러한 이유로 PIPO의 비선형 계층인 S-layer의 세부 연산에 대해 마스킹 기법을 적용하는 것이 아닌, S-layer 연산 전체에 대한 사전 연산 테이블을 생성하고 여기에 마스킹 대응책을 적용할 수 있는 1차 마스킹 대응책을 제안한다. 그러나 테이블 참조 방식만 사용하면 커플링 취약점 등의 영향으로 비밀 키가 복구될 가능성이 여전히 존재하므로 셔플링(shuffling) 기법 추가한 대응책을 제시하고자 한다.

또한, 테이블 참조 기반 마스킹 대응책의 특성상 많은 양의 메모리를 사용할 수 있기 때문에, 본 논문에서는 메모리 공간과 연산 시간의 상충 관계를 고려하여 전체 테이블 및 소형 테이블 마스킹 기법을 제안한다. 제안한 테이블 참조 기반 대응책들을 8비트 마이크로컨트롤러에서 분석한 결과, 고전적인 테이블 마스킹 방법보다 많은 클럭 사이클이 소요되지만 기존 방법보다 소요되는 메모리가 적고 1차 부채널 분석 공격에 대응할 수 있음을 확인하였다.

2. 경량 블록 암호 알고리즘 PIPO

대칭 키 암호 알고리즘인 PIPO는 저성능, 저용량 기기를 위해 2020년에 제안된 경량 블록 암호 알고리즘이다. 이는 S-box의 비선형 연산 개수를 최소화하여 부채널 분석 대응책인 마스킹 기법을 보다 효율적으로 적용할 수 있도록 설계되었다.

암・복호화 과정은 64비트 단위로 수행되며 키 길이는 128, 256비트 두 종류를 지원한다. 암・복호화는 각각의 키 길이에 따라 13, 17라운드로 수행되고, 각 라운드는 치환 계층인 S-layer와 전치 계층인 P-layer 그리고 라운드키 XOR 과정으로 구성된다. 또한, 연산 과정에서 모든 상태(state)는 8×8크기의 비트 행렬로 처리된다.

PIPO의 S-layer(S-box)는 비트 행렬에서 열(column) 단위로, P-layer는 행(row) 단위로 수행된다. 이때 S-box는 비트별 XOR, AND, OR, NOT 연산으로만 구성된다. 따라서 평문의 각 바이트를 저장하고 있는 레지스터에 대해 비트 연산을 수행한다면 8개의 열에 대한 S-box 연산을 테이블 참조 없이 병렬로 빠르게 수행할 수 있다. 즉, Fig. 1에서 보는 바와 같이 각 레지스터에 저장된 비트는 상호 독립적이므로 ($b_{56}b_{48}b_{40}b_{32}b_{24}b_{16}b_8b_0$),

($b_{57}b_{49}b_{41}b_{33}b_{25}b_{17}b_9b_1$)를 포함한 8개 열에 대해 S-box 연산이 병렬로 수행될 수 있어 고속 연산이 가능하다.

PIPO의 라운드 키는 마스터 키 일부와 라운드 상수가 XOR된 간단한 구조이다. 따라서 사전에 키 확장을 수행하지 않고 암호화 과정에서 마스터 키와 라운드 상수를 XOR하여 사용함으로써 메모리 공간을 절약할 수 있다. 그러나 이러한 키 확장 과정의 특성으로 인해 PIPO에 대한 부채널 공격을 수행할 경우 짝수 혹은 홀수 라운드키 1개씩만 복구하면 마스터 키가 복구될 수 있다.

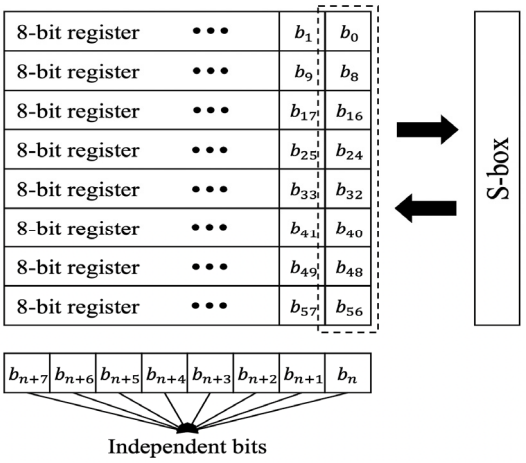

Fig. 1. The units of S-box operation and state matrix stored in 8-bit registers.

PIPO는 마이크로컨트롤러와 같은 저성능 기기에서 사용할 목적으로 설계된 암호 알고리즘이므로 부채널 분석에 대한 대응책이 필수적으로 고려되어야 한다. PIPO는 S-layer의 AND, OR을 제외한 모든 연산이 선형 비트 연산으로 구성되므로 부울 마스크(Boolean mask)가 적용될 수 있다. 그러나 비선형 AND, OR 연산을 수행할 경우에는 마스크 값이 유지되지 않아 추가적인 방법이 사용되어야 한다. 이 과정에서 연산의 많은 오버헤드가 발생하므로 PIPO는 비선형 연산인 AND와 OR의 개수를 최소화해 고차 마스킹을 적용할 경우에도 연산 오버헤드를 최소화하고자 하였다.

마스킹이 적용된 값에 대해 비선형 비트 연산인 AND, OR을 수행할 수 있는 방법으로는 ISW 방법이 있다[8]. 이는 PIPO를 처음 제안한 문헌 [4]에서도 고차 마스킹 적용을 위해 채택한 방법이며 마스킹의 차수(order)와 관계없이 적용할 수 있다. 그러나 모든 쉐어(share), 즉 마스크 값이 저장된 곳에 대한 연산이 수행되어야 하고, 이 과정에서 각 쉐어간 레지스터 중복 사용 등의 커플링 취약점으로 비밀 키가 쉽게 노출되기도 한다[9]. 이러한 현상 때문에 S-layer의 세부 연산마다 마스크 값을 계산하고 유지하는 방법이 아닌, 메모리를 많이 사용하더라도 마스킹 테이블을 미리 만들어 두고 이를 나중에 호출하는 테이블 기반의 마스킹 방법을 선택할 수도 있다.

본 연구에서는 먼저 테이블 기반으로 PIPO의 S-layer를 구현할 경우, 부채널 공격에 대응할 수 있는 마스킹 기법을 적용하더라고 비밀 키가 복구되는 것을 확인한다. 그리고 이 마스킹 기법에 셔플링과 같은 추가적인 대응책을 혼용해야 함을 증명한다. 또한 사용하는 테이블 메모리를 줄이기 위한 방법도 제안하고자 한다. 부채널 공격에 안전한 PIPO 암호 알고리즘의 검증을 위해 실제 마이크로 프로세서에 암호 알고리즘을 구현하고 전력 신호 측정을 통해 구체적인 비밀 키 추출 과정과 대응책의 효율성을 단계적으로 확인한다.

3. PIPO에 대한 테이블 기반 마스킹

마스킹 대응책이란 부채널 분석 공격에 대한 대응책으로서 암호 알고리즘의 중간 연산 값을 2개 이상의 쉐어 공간에 임의로 나누어 저장 및 수행하는 방법이다. 먼저 8비트 단위 입·출력을 갖는 S-box를 가정할 경우 이 테이블을 S_8라 두자. 다른 블록 암호 알고리즘에서는 S-Layer 연산을 수행하기 위한 256바이트의 사전 계산 테이블 S_8과 부채널 분석 공격에 대응하기 위한 256바이트의 마스킹 테이블을 추가로 사용하는 등의 방법을 사용하였다[10,11]. 블록 암호의 구조상 PIPO에서도 기존과 동일한 방법으로 사전 테이블과 마스킹 테이블을 사용할 수 있다. 그러나 단순히 마스킹 테이블만을 적용할 경우에도 부채널 누출이 발생할 수 있으므로, 본 논문에서는 사용되는 메모리를 줄일 수 있는 테이블 기반의 새로운 마스킹 대응책을 제안하고자 한다. 특히, PIPO의 S-layer 8비트 단위의 입·출력을 갖고 있지만 내부적으로는 3비트 단위 S-box인 S_3와 5비트 단위 S-box인 S_5^1, S_5^2로 구성된 unbalanced-bridge 구조를 사용하고 있음을 주목하였다.

본 논문에서는 이러한 3개의 작은 S-box를 구성하고 이에 대해 작은 마스킹 테이블을 이용하자는 것이다. 이

렇게 함으로써 일차적으로 수행 시간이 조금 더 걸리더라도 메모리 사용량을 최소화하자는 것이다. 즉, 전체 테이블 S_8에 대해 마스킹을 적용하는 기존 방법과 소형 테이블인 S_3, S_5^1, S_5^2에 대해 마스킹을 적용하는 두 가지 방법을 구현하여 실험하였다.

3.1 S-Box 테이블 S_8에 대한 마스킹

PIPO의 8비트 단위 S-box인 S_8의 마스킹 테이블을 생성하기 위해서는 각각 8비트인 테이블의 입력 마스크 M_i와 출력 마스크 M_o가 필요하다. 위 두 마스크 값과 Fig. 2의 알고리즘을 사용해 마스킹된 테이블 MS_8가 사전에 생성되어야 한다.

Algorithm 1. Generate a masked table MS_8 using a S_8.

INPUT Input mask M_i,
Output mask M_o,
Table S_8,
Empty masked table MS_8

OUTPUT Masked table MS_8

1. **for** i=0 **to** 255 **do**
2. $MS_8[i \oplus M_i] \leftarrow S_8[i] \oplus M_o$
3. **end for**
4. **return** MS_8

Fig. 2. An algorithm for generating a masked table.

이 과정에서 256바이트 크기의 고정된 S_8가 있어야 하며 입・출력 마스크 값에 따라 생성된 MS_8 역시 256바이트 크기를 갖는다. 마스킹 테이블 MS_8은 M_i 마스크가 씌워진 입력 값이 주어지면 정상 S_8의 출력 값에 새로운 마스크 값인 M_o가 씌워진 값을 반환하게 된다.

일반적으로 사용하는 마스킹 방법에서는 입・출력 마스크가 M_i, M_o로 고정된 1개의 MS_8만을 사용하기 때문에 본 논문에서 평문에 씌워질 마스크 값 역시 Fig. 3과 같이 8비트 M_i를 8번 사용한다. 이후 화이트닝 키가 XOR되어도 마스크 값은 M_i로 유지된다. 그리고 1라운드 M-S-layer 즉, MS_8 테이블을 이용해 S-layer 연산을 수행하며 이는 8번의 테이블 참조만으로 구성된다.

해당 연산을 수행한 이후 상태 행렬의 마스크 값은 M_o이므로, 다음 라운드를 고려해 M_i마스크로 변환한다. 이 과정에서 실제 중간 연산 값이 노출되지 않도록 새로운 마스크 M_i를 먼저 적용한 후, 기존 마스크 M_o를 제거함을 주의할 필요가 있다. 이후 P-layer와 라운드키 XOR 과정을 거쳐도 M_i 마스크 값은 유지되므로 이후 라운드 모두 마스킹된 상태로 연산이 수행될 수 있다. 마지막으로 암호문에 씌워진 M_i 마스크를 제거함으로써 암호화 과정을 종료하게 된다.

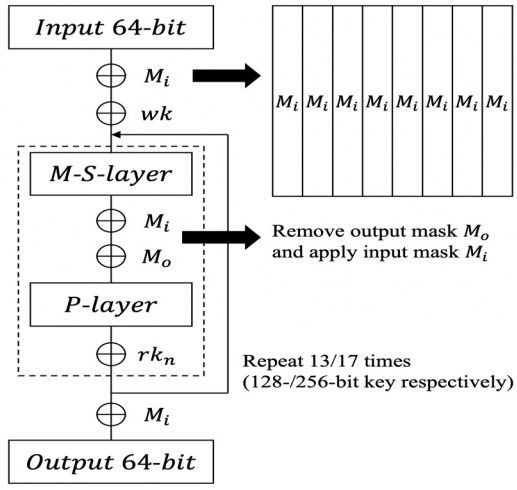

Fig. 3. The overall structure of the classical masking scheme for the full table S_8.

3.2 소형 테이블 S_3, S_5^1, S_5^2 마스킹

전체 테이블 S_8과 마스킹 테이블 MS_8은 각각 256바이트이므로 마스킹 적용을 위해서는 총 512바이트의 많은 메모리가 필요하다. 그러나 소형 테이블 S_3, S_5^1, S_5^2은 각각 8, 32, 32바이트이므로 마스킹 테이블까지 포함하면 144바이트의 메모리만 필요하다. 그러나 S_8을 이용할 경우 한 라운드에 8번의 테이블 참조만이 수행되지만, 소형 테이블을 이용할 경우에는 24번의 테이블 참조가 수행되어야 하므로 수행 시간이 증가하게 된다.

소형 테이블 마스킹을 적용한 PIPO의 암호화 과정은 앞선 MS_8를 사용한 방법에서 M-S-layer 연산 부분과 해당 출력 마스크를 제거하는 부분 외에는 모두 동일하다. M-S-layer의 출력 마스크는 MS_8을 이용할 때 M_o였지만, 소형 테이블을 이용할 때는 소형 S-box의 출력 마스크 값 M_o이고 이를 기반으로 새롭게 생성된 M_0'가 출력 마스크가 된다.

소형 마스킹 테이블이 적용된 M-S-layer 연산을 상세히 나타낸 것이 Fig. 4이다. 그림에서 M_i^5, M_i^3은 M_i

의 일부 비트열을 의미하며, M_o^5, M_o^3 역시 M_o의 일부 비트열을 의미한다. 즉, 소형 마스킹 테이블이 정해지면 S-box 내부의 중간 마스크 값이 다음 수식 (1)과 같이 정해지게 되는데 이것이 S-box의 최종 마스크 값은 아니다.

$$M_0 = M_0^5 \ || \ M_0^3 \quad (1)$$

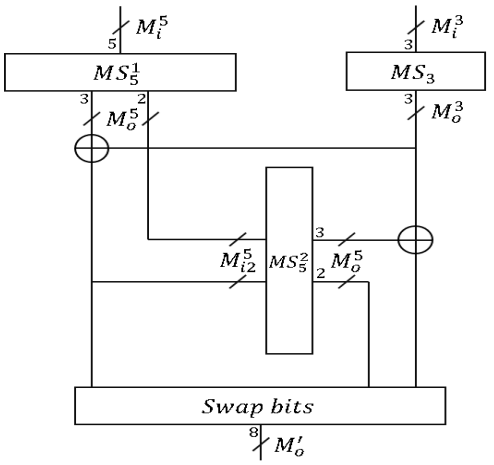

Fig. 4. The detailed structure of S-layer for proposed smaller table masking.

따라서 최종 마스크 값 M_o'은 다음 Fig. 5과 같이 M_o값으로부터 계산할 수 있다. 이 경우 M_o'은 중간 마스크 값 M_o만으로 결정되므로 암호화 이전의 사전 준비 단계에서 한 번만 계산하면 된다. 결국, 소형 마스킹 테이블을 이용하는 경우에는 S-Box 연산을 마치면 M_i 값을 마스킹을 적용한 후 다시 M_0'을 제거해야 한다.

```
Mo' = 0;
Mo' |= ((Mo & 0b1) ^ ((Mo >> 6) & 0b1));
Mo' = Mo' << 1;
Mo' |= (((Mo >> 4) & 0b1) ^ ((Mo >> 7) & 0b1));
Mo' = Mo' << 1;
Mo' |= (((Mo >> 0) & 0b1) ^ ((Mo >> 5) & 0b1));
Mo' = Mo' << 1;
Mo' |= ((Mo >> 1) & 0b1);
Mo' = Mo' << 1;
Mo' |= ((Mo >> 2) & 0b1);
Mo' = Mo' << 1;
Mo' |= (((Mo >> 3) & 0b1) ^ ((Mo >> 7) & 0b1));
Mo' = Mo' << 1;
Mo' |= (((Mo >> 4) & 0b1) ^ ((Mo >> 5) & 0b1));
Mo' = Mo' << 1;
Mo' |= (((Mo >> 3) & 0b1) ^ ((Mo >> 6) & 0b1));
```

Fig. 5. Calculation code of M_o' using M_o

3.3 테이블 참조 및 평문 비트 로드 셔플링

마스킹 테이블 MS_8 또는 MS_3, MS_5^1, MS_5^2을 이용해 연산을 수행하더라도 의도치 않은 미미한 부채널 누출이 발생해 비밀 키가 노출되는 경우가 존재하는 것을 실험적으로 확인하였다. 따라서 제안하는 마스킹 대응책에 셔플링 기법을 추가하여 이를 완화하고자 한다.

각 라운드에서 상태 행렬에 대해 연산을 수행할 경우에는 8개의 열 모두 처리해야 하므로 MS_8 또는 MS_3, MS_5^1, MS_5^2 테이블 각각 8번의 참조가 발생한다. 따라서 처리되는 상태 행렬에서 열의 순서를 매 암호화 과정마다 무작위로 섞는 셔플링 대응 기법을 추가할 수 있다. 이때 연산 순서를 결정하는데 사용되는 셔플링 테이블 8바이트가 추가로 필요하며, 이 과정에서 사용되는 난수는 마스크 값인 M_i와 M_o를 사용할 수 있다.

또한, 마스킹 테이블 참조를 위해 각 레지스터에서 1비트를 읽어오는 과정에서도 미미한 비밀 정보 누출이 발생함을 확인하였다. 따라서 테이블 참조 이전, 이후에 각 레지스터에서 비트를 읽어오거나 저장하는 과정에 셔플링을 적용하여 이에 대한 누출을 완화할 수 있다. 이 경우에도 8바이트의 셔플링 테이블이 필요한데, 이는 앞서 테이블 참조 셔플링에서 사용된 셔플링 테이블을 공유한다. 최종적으로 제안하는 테이블 기반 1차 마스킹 대응책은 위와 같은 2가지 셔플링 대응책을 포함한다.

4. 부채널 전력 분석 공격 및 효율성

4.1 PIPO에 대한 전력 분석 공격

전력 분석 공격은 암호 알고리즘의 중간 연산 결과 값을 모델링하여 이를 처리할 때 소비 전력을 예측하고 실제 측정된 소비 전력과의 연관성을 분석함으로써 수행된다. 본 논문의 모든 공격 및 분석 실험은 8비트 단위의 ATxmega128D4 마이크로컨트롤러를 대상으로 수행하며, 신호 수집은 ChipWhisperer 플랫폼을 사용해 다음 Fig. 6과 같은 환경에서 수행한다[12].

암호 알고리즘의 중간 연산 결과 값을 계산한다는 것은 앞선 모든 라운드의 키를 알고 있어야만 가능하다. 따라서 전력 분석 공격은 첫 라운드 혹은 마지막 라운드부터 순차적으로 수행된다. 본 논문에서의 부채널 전력 분석 공격은 모두 화이트닝 키 wk를 복구하는 것을 목표로 1라운드 S-layer의 결과 값을 예측해 수행한다.

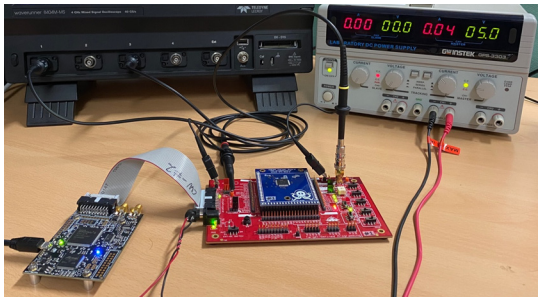

Fig. 6. Experimental setup using ChipWhisperer platform.

제안하는 MS_8 및 MS_3, MS_5^1, MS_5^2 기반의 마스킹 및 셔플링을 적용한 구현체가 동작할 때 1~2 또는 1~3 라운드에서 수집한 10,000개의 파형을 분석하고, 64개의 모든 공격 조합에 대해 올바른 키를 가정했을 때의 상관 전력분석(Correlation Power Analysis)을 수행하였다[13]. 제안 방식에 대한 상관 계수를 분석한 결과를 나타낸 것이 Fig. 7과 Fig. 8이다. 그림에서 보는 바와 같이 올바른 키에서 부채널 누출이 존재하지 않으며, 64개의 비트 조합 모두에서 키가 노출되지 않는 것을 확인할 수 있다.

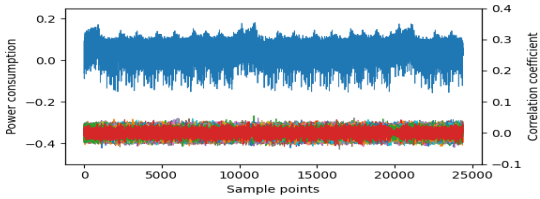

Fig. 7. The result of 64 CPAs for PIPO with proposed full table-based masking scheme. (Target round: 1~2, Compile optimization : -O3)

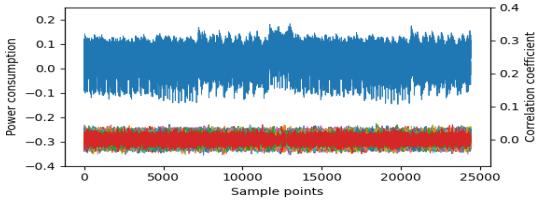

Fig. 8. The result of 64 CPAs for PIPO with proposed smaller table-based masking scheme. (Target round: 1~2, Compile optimization : -O3)

한편 문헌 [4]에서 사용한 ISW 방식을 이용하여 1차 마스킹을 구현하고, 해당 구현체의 1~2라운드에 대해서도 공격을 시도하였다. 총 64개의 비트 조합을 대상으로 공격을 수행한 이후 올바른 키를 가정했을 때의 상관 계수를 나타낸 것이 다음 Fig. 9이다. 그림에서 보는 바와 같이 부채널 누출이 발생하며, 이로 인해 64비트의 화이트닝 키가 모두 복구될 수 있음을 확인하였다. 이는 마스킹 알고리즘의 문제가 아닌 각 쉐어간 동일한 레지스터를 사용하는 등의 문제로 발생하는 커플링 취약점으로 분석된다.

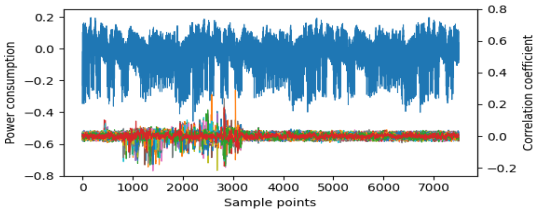

Fig. 9. The result of 64 CPAs and correlation coefficients for PIPO with ISW-based masking scheme. (Target round: 1~3, Compile optimization : -O3)

4.2 마스킹 대응책의 오버헤드 분석

PIPO는 저사양 기기에서 수행되는 경량 블록 암호 알고리즘이므로 부채널 분석 대응책 적용으로 인한 연산의 오버헤드 및 추가 메모리 공간 그리고 필요한 난수의 크기에 대해서도 분석이 필요하다. 따라서 8비트 마이크로컨트롤러 ATxmega128을 기준으로 동작 클럭 사이클과 추가 메모리 공간을 분석 및 정리한 것이 Table 1이다.

부채널 분석 대응책이 적용되지 않은 PIPO의 비트슬라이스 구현체는 최적화 옵션이 0일 경우 약 2만 2천 그리고 1~3인 경우 약 5~6천 사이클로 동작한다. 그리고 ISW 방법을 이용해 S-layer 세부 연산에 대해 직접 1차 마스킹 적용한 경우 최적화 옵션이 0일 때는 약 18만 클럭 사이클이 소요되지만, 3인 경우에는 약 7천 5백 클럭 사이클만으로 수행된다. 그러나 해당 구현체는 커플링 취약점 등으로 인해 1차 부채널 분석으로도 키가 노출되는 경우가 존재한다.

제안하는 방법 중에서 전체 테이블인 S_8에 대해 마스킹 및 셔플링을 적용한 구현체는 최적화 옵션이 0일 때 마스킹 테이블 생성 과정에서 약 1만 6천 클럭 사이클, 암호화 과정에서 약 28만 3천 사이클이 소요된다. 또한, 최적화 옵션이 3일 경우에는 테이블 생성 및 암호화 과정 각각 5천 8백, 13만 2천 클럭 사이클이 소요된다. 또한, S_8, MS_8 그리고 셔플링 테이블 저장을 위한 추가 메

Table 1. Comparison of the clock cycle and additional memory size according to the countermeasure.

Implementation	Random byte	Additional table memory	Complier optimization	Clock cycle		CPA resistence
				Table gen.	Enc.	
Original	0 byte	N/A	-O0	0	22,438	×
			-O1	0	6,350	
			-O2	0	5,770	
			-O3	0	5,128	
ISW[4]	9 byte	N/A	-O0	0	181,235	△
			-O1	0	39,328	
			-O2	0	25,210	
			-O3	0	7,523	
Full table Masking (S_8)	2 byte	512 byte (256+256)	-O0	14,123	153,159	×
			-O1	6,666	79,449	
			-O2	4,878	75,552	
			-O3	4,878	74,999	
Proposed M. + S. (S_8)	2 byte	520 byte (256+256+8)	-O0	16,573	283,142	○
			-O1	8,361	136,227	
			-O2	6,556	136,610	
			-O3	5,839	132,230	
Proposed M. + S. (S_3, S_5^1, S_5^2)	2 byte	152 byte (32+32+8+32+32+8+8)	-O0	6,943	426,806	
			-O1	3,839	193,423	
			-O2	3,271	192,199	
			-O3	2,623	176,085	

모리 공간 520바이트가 필요하며, 9바이트의 난수가 필요한 ISW 방법에 반해 오직 2바이트의 난수만으로 안전하게 구현할 수 있다.

소형 테이블인 S_3, S_5^1, S_5^2에 대해 마스킹 및 셔플링을 적용한 구현체는 모든 최적화 옵션에서 S_8 대상 마스킹 구현체에 비해 암호화 과정에서 약 1.3~1.5배 느린 것으로 분석된다. 그러나 필요로 하는 추가 메모리 공간이 S_8 구현체가 520바이트인 것에 반해 오직 152바이트만이 필요한 것으로 보아 메모리 측면에서는 보다 우수한 것으로 판단된다.

PIPO에 대한 테이블 마스킹 대응책을 적용할 경우 대상 S-box의 종류에 따라 메모리 공간 및 클럭 사이클간 상충 관계가 존재한다. 즉, 많은 메모리를 사용하는 S_8 대상의 마스킹 대응책은 비교적 적은 클럭 사이클 동안에 실행될 수 있지만, 소형 테이블 대상의 마스킹 대응책은 적은 메모리를 사용하는 대신 많은 클럭 사이클이 소요된다. 따라서 이러한 상충 관계를 고려해 적용 대상 환경에 적합한 방법을 선택해야 할 것이다.

5. 결론

최근에 제안된 경량 블록 암호 알고리즘 PIPO는 원래 부채널 공격에 대응하면서 구현을 최적화하기 위해 비선형 연산을 최소화하면서 병렬처리가 가능하도록 설계되었다. PIPO는 S-Layer를 사전 계산 테이블을 이용하여 구현할 수 있는데, 이 경우에는 직접 비선형 연산을 구현하는 방법에 비해 메모리는 많이 필요하지만, 고속으로 수행할 수 있다. 그러나 사전 테이블 이용 방식도 부채널 공격에 취약한 특성이 있어 마스킹 테이블 기법을 사용할 수 있다.

본 논문에서는 기존에 제안되었던 마스킹 테이블 대응책을 마이크로프로세서에 직접 구현한 결과, 비밀 키가 누설됨을 실험을 통해 확인하였다. 그러나 마스킹 기법과 셔플링 기법을 동시에 적용하면 1차 전력 부채널 공격을 충분히 방어할 수 있음도 확인하였다. 또한, 논문에서는 전체 S-box 테이블을 이용하는 것이 아니라 S-Layer 내부의 소형 S-box만 마스킹 테이블을 적용함으로써 전력 분석 공격에 대응하면서 사용 메모리를 감소시킬 수 있는 방안도 제시하였다. 제안하는 부채널 분석 공격 대응 기법들은 더 많은 수행 시간과 메모리가 필요하지만, 보다 안전하게사용할 수 있다.

References

[1] T. R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith and L. Wingers, "The SIMON and SPECK lightweight block ciphers", Proceedings of the 2015 IEEE Design Automation Conference(DAC'15), CA, USA, pp. 1-6, Jun. 2015.
DOI: https://doi.org/10.1145/2744769.2747946

[2] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device", CHES'06, LNCS 4249, pp. 46-59, 2006.
DOI: https://doi.org/10.1007/11894063_4

[3] D. Hong, J. Lee, D. Kim, D. Kwon, K. Ryu, and D. Lee, "LEA, A 128-bit block cipher for fast encryption on common processors," WISA'13, LNCS 8267, pp. 3-27, 2014.
DOI: https://doi.org/10.1007/978-3-319-05149-9_1

[4] H. Kim, Y. Jeon, G. Kim, J. Kim, B. Sim, D. Han, H. Seo, S. Kim, S. Hong, J. Sung, and D. Hong, "PIPO :A Lightweight Block Cipher with Efficient Higher-Order Masking Software Implementations," ICISC'20, LNCS 12593, pp. 99-122, 2020.
DOI: https://doi.org/10.1007/978-3-030-68890-5_6

[5] B. Bhasin, T. Graba, J. Danger, and Z. Najm, "A look into SIMON from a side-channel perspective", *Proceedings of* IEEE International Symposium on Hardware-Oriented Security and Trust(HOST'14), pp. 56-59, May 2014.
DOI: https://doi.org/10.1109/HST.2014.6855568

[6] T. Kim, Y. Won, J. Park, H. An and D. Han, "Side Channel Attacks on HIGHT and Its Countermeasures," Journal of KIISC, Vol. 25, No. 2, pp. 457-466, 2015.
DOI: https://doi.org/10.13089/JKIISC.2015.25.2.457

[7] J. Park, T. Kim, H. An, Y. Won and D. Han, "Side channel Attacks on LEA and Its Countermeasures," Journal of KIISC, Vol. 25, No. 2, pp. 449-456, 2015.
DOI: https://doi.org/10.13089/JKIISC.2015.25.2.449

[8] Y. Ishai, A. Sahai, and D. Wagner, "Private Circuits: Securing Hardware against Probing Attacks," CRYPTO'03, LNCS 2729, pp. 463-481, 2003.
DOI: https://doi.org/10.1007/978-3-540-45146-4_27

[9] K. Papagiannopoulos and N. Veshchikov, "Mind the Gap: Towards Secure 1st-order Masking in Software," COSADE'17, LNCS 10348, pp. 282-297, 2017.
DOI: https://doi.org/10.1007/978-3-319-64647-3_17

[10] S. Chari, C. Jutla, J. Rao and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks", CRYPTO'99, LNCS 1666, pp. 398-412, 1999.
DOI: https://doi.org/10.1007/3-540-48405-1_26

[11] H. Yoo, J. Ha, C. Kim, I, Park, and S. Moon, "A secure ARIA implementation resistant to differential power attack using random masking method", Journal of KIISC, Vol. 16, No. 2, pp. 129-139, 2006.
DOI: https://doi.org/10.13089/JKIISC.2006.16.2.129

[12] C. O'Flynn and Z. Chen, "ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research, COSADE'14, LNCS 8622, pp. 243-260, 2014.
DOI: https://doi.org/10.1007/978-3-319-10175-0_17

[13] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," CHES'04, LNCS 3156. pp. 16-29, 2004
DOI: https://doi.org/10.1007/978-3-540-28632-5_2

배 대 현(Daehyeon Bae) [준회원]

- 2021년 2월 : 호서대학교 컴퓨터정보공학부 (학사)
- 2022년 2월 : 호서대학교 정보보호학과 (석사)

〈관심분야〉
부채널 공격, 암호학, 정보보호

이 재 욱(Jaewook Lee) [준회원]

- 2017년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 학부과정

〈관심분야〉
인공지능 보안, 자연어처리, 부채널 공격

이 현 로(Hyeonro Lee) [준회원]

• 2017년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 학부과정

〈관심분야〉
부채널 공격, 양자내성 암호, 머신러닝

하 재 철(Jaecheol Ha) [종신회원]

• 1989년 2월 : 경북대학교 전자공학과 (학사)
• 1993년 8월 : 경북대학교 전자공학과 (석사)
• 1998년 2월 : 경북대학교 전자공학과 (박사)
• 1998년 3월 ~ 2007년 2월 : 나사렛대학교 정보통신학과 교수
• 2007년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 교수
• 2013년 1월 ~ 현재 : 한국정보보호학회 상임부회장
• 2009년 1월 ~ 현재 : 한국산학기술학회 이사

〈관심분야〉
암호학, 네트워크 보안, 부채널 공격, 머신러닝