2018년 정보보호학술논문발표회

국산 암호 알고리즘 부채널 분석에 대한 고찰

안규황*, 권용빈*, 권혁동*, 서화정*†

*한성대학교 대학원 정보시스템공학과

A Study on Side Channel Analysis of Domestic Cryptographic Algorithm

Kyuhwang An*, Yong-Been Kwon* Hyeokdong Kwon*
Hwajeong Seo*†

*Division of information system engineering, Hansung University.

요 약

암호 알고리즘의 키(Key)를 찾아내는 방법으로 기존의 수학적 알고리즘을 분석하며 접근하는 방법과 다르게 암호 알고리즘이 하드웨어에서 동작할 때 생기는 전력 소모, 시간 등의 누수 정보들을 분석하는 방법으로 부채널 분석(Side-Channel Analysis)이 있다. 현재 AES 등 널리 이용되고 있는 국제 표준 알고리즘들이 부채널 분석으로부터의 취약점이 밝혀지고 있고 이와 동시에 대응 기법이 마련되는 등 연구가 활발하게 진행되고 있어, 국산 암호를 대상으로도 이러한 연구를 통해 취약점을 분석하고 보완할 필요하다. 본 논문에서는 부채널 분석 기법과 그 대응기법들을 소개한다.

I. 부채널 분석이란

부채널 분석이란 암호 알고리즘이 탑재된 전자기기를 작동시킬 경우 발생하는 전자파, 전력 소모량등의 누수정보들을 분석하여 비밀 키를 획득하는 분석법이다. 최초의 부채널 분석은 1996년에 P.Kocher에 의해 발표한 시차분석 공격[1]이며, 해당 논문을 기점으로 본격적인 부채널 관련 연구가 실행되었다.

부채널 분석이 발표되기 이전에는 다양한 암호 알고리즘의 비밀 키를 찾기 위해 다양한 평문을 넣는 방법인 전수조사, 비밀 키를 탈취하는 키 로깅 등이 시행되어 왔지만, 부채널 분석 발표 이후에는 암호 알고리즘이 탑재된 전자기기가 발생시키는 누수정보들을 분석하여 비밀 키를 획득할 수 있게 되었다. 따라서 이전과 달리 암호 알고리즘 또는 하드웨어 설계, 개발 시에 누수정보를 고려하여야 한다.

II. 부채널 분석 기법과 그 대응 기법

본장에서는 부채널 분석 중 연산의 전력소모량의 차이를 이용하여 분석하는 대표적인 기법들인 SPA(Simple Power Analysis), DPA(Differential Power Analysis), CPA(Correlation Power Analysis) 그리고 Guessing Entropy를 소개하고 이에 대한 대응 기법들을 소개한다.

2.1 분석 기법

2.1.1 SPA(Simple Power Analysis)

암호 알고리즘의 내부 함수들은 각기 다르게 구성되어 있어 서로 다른 전력 소모를 보인다. 이에 따라 각 함수별로 패턴화되어 파장으로 나타낼 수 있으며 이를 기반으로 어떤 함수들이 실행되는지를 알 수 있어 중간 값 등 키와 관계된 정보의 추출이 가능하다.

2.1.2 DPA(Differential Power Analysis)

평문을 넣어 암호를 구동하여 파형을 수집하고 수집된 파형에서 분류함수를 통해 정확한 비밀 키와 반응이 일어나는 파형을 얻는다. 두 개로 분류된 데이터를 각각 평균하여 차분을 구한다. 추측된 비밀 키 값이 올바를 경우 반응지점에서의 차이가 크게 발생해 차분 값이 큰 값을 가지고 맞지 않다면 0에 가까운 작은 값을 가진다. 이를 추측된 모든 비밀 키에 반복

하여 올바른 비밀 키 값을 찾을 수 있다.

2.1.3 CPA(Correlation Power Analysis)

평문 또는 비밀 키 값과 평문과 비밀 키를 연산(평문을 P, 비밀 키를 C라고 할 경우 P + C, P ⊕ C 등)하여 나오는 값을 선택한다. 공격자가 원하는 N번만큼의 평문 데이터를 암호 알고리즘에 입력시킨 후 소비되는 전력을 측정한다. 이때 소비되는 전력량을 I라고 할 때 해당 파형의 길이는 N * I가 된다. 예상 가능한 중간 값을 모든 키에 대입하여 계산한다. 계산 된 중간 값에 상응하는 전력 소비량을 계산하여 실제 전력 소비 값과 비교한다. 여기서 만약 임의로 측정한 값이 실제 전력 소비 값과 높은 상관관계를 갖고 있다면, 실제 비밀 키를 찾았다고 볼 수 있다.

2.1.4 Guessing Entropy

Guessing Entropy는 무작위로 추출한 파형 중 실제 비밀 키 값을 도출하기 위하여 공격자가 임의로 정의한 횟수만큼 돌려 실제 키와 상관관계가 높은 순서대로 나타내는 방법을 의미한다. 이때 추출하는 파형이 아무리 많다고 할지라도 상관관계가 가장 높은 비밀 키를 실제 비밀 키라 칭할 수 있다.

2.2 대응 기법

이에 대한 전력 분석의 대응 기법인 무작위성 기법, 마스킹 기법, 블라인딩 기법을 소개한다.

2.2.1 무작위성(Randomization) 기법

무작위성 기법을 적용하지 않은 암호 알고리즘의 경우 소비 전력 및 실행 시간 등 다양한 방법으로 부가적인 정보를 획득할 수 있게 되지만, 무작위성 기법을 적용한 암호 알고리즘의 경우 암호 알고리즘을 실행시키는데 발생하는 모든 값에 대해 난수처리 한다. 이렇게 되면 공격자는 최종 출력 값으로 난수를 획득하게 되고, 해당 난수에 대한 중간 값을 유추할 수 없기 때문에 SPA에 대하여 대응할 수 있다.

2.2.2 마스킹(Masking) 기법

마스킹 기법이란 암호 알고리즘이 연산을 수행할 때 발생하는 중간 값을 공격자가 유추할 수 없게끔 발생하는 전력 소모량을 조작하여 실제 비밀 키와의 상관관계를 제거하는 방법이다. 기본적인 마스킹 연산은 수식 1과 같다.

$$\text{불 연산} \ : x' = x \oplus r \quad (1)$$
$$\text{산술 연산} \ : A = x - r \bmod 2^k$$

2.2.3 블라인딩(Blinding) 기법

블라인딩 기법은 Chaum이 최초로 제안하였다. 블라인딩 기법은 암호 알고리즘에 적용되어 있는 함수를 이용하여 평문을 입력하여 비밀 키가 도출될 때 평문과 비밀 키를 숨기면서 값을 계산해주는 암호 함수를 설계해주는 기법이다.

III. 관련 연구 동향

3.1 SEED[2]

SEED에 대한 부채널 공격을 실험하기 위해 해당 논문에서는 XMEGA 보드를 사용했다. 해당 논문에서는 더미 연산 & 셔플링 및 1차 마스킹이 적용된 SEED에 대한 SOCPA 공격법에 대하여 설명한다. 이를 설명하기에 앞서 해당 논문에서 제안한 방법들에 대한 평가들을 위하여 몇 가지를 정의한다. 첫 번째로 부채널 공격 수행을 위한 비밀 키 추측을 위한 최소 경우의 수이다. 예를 들어 상관 전력 분석으로 S-Box를 추측하기 위해서는 2^8만큼의 공격 복잡도가 생긴다. 비밀 키 추측을 위해 최소 경우의 수만큼 수행해야 되기 때문이다. 두 번째로 셔플링이 적용되지 않은 비밀 키가 추출되는 최소 파형의 수를 ∂고 할 때, 특정 연산에서 셔플링이 $\frac{1}{n}$만큼 적용된다면, 필요한 최소 파형의 수는 $\partial \times \frac{n(n-1)}{2}$이다. SOCPA에 대한 공격지점은 더미 연산 및 셔플링 기법이 적용된 S-Box 출력과 G-함수 출력을 의미한다.

제안하는 공격 기법을 사용할 경우 위에서 언급한 2^8 만큼의 공격 복잡도가 아닌 2^{16}이 된다. 그 이유는 8비트 중 2비트 만 상관이 있기 때문이다. Noise-free 환경에서도 입력 가능한 값에 대한 상관도에 대하여 전수조사 하였을 때, 8비트의 값과 2비트의 상관도는 0.25이다.

따라서 필요파형 수는 상관도의 역수의 제곱에 비례하여 $\left(\frac{1}{0.25}\right)^2$배의 파형 수가 필요하다.

제안한 공격 방법을 수행하기 위해 G-함수 출력 값을 해밍웨이트 값으로 생성하였다. 현재 SEED에 적용되어 있는 셔플링 대응기법을 우회하기 위해 G-함수 출력 값을 로드하는 부분인 G-함수 덧셈 부분에서 파형을 수집하였으며, 기존 공격 방법들에 대한 공격을 수행하기 위해 G-함수 연산 부분을 따로 수집하였다. 그 결과 해당 논문에서 제안한 방법으로 약 8,000개 파형으로 정확히 Guessing Entropy가 1로 수렴함을 알 수 있었으며, 실제 장비에서 100,000개 파형에 대해 일반적인 방법으로 옳은 비밀 키를 추출하는데 실패했지만 해당 논문의 공격 방법에 의해 8,000개의 파형으로 옳은 비밀 키를 추출할 수 있음을 증명하였다.

3.2 HIGHT[3]

3.2.1 평문과 암호문을 이용한 HIGHT 부채널 분석

HIGHT 암호 알고리즘에 존재하는 취약점을 검증하고 이를 증명하기 위해 해당 논문에서는 ATmega128 보드를 사용했다. 해당 논문에서는 2개의 선형 연산을 분석하여 비밀 키를 복구하는 방법으로 취약점을 보여준다. 그림 1를 보면 빨간색 네모와 파란색 동그라미가 있다. 빨간색 네모의 경우 비선형 연산을 분석하는 지점이며 파란색 동그라미는 선형 연산을 분석하는 지점이다. 선형 연산보다 비선형 연산이 부채널 분석에 더 용이하게 사용되지만, WK_1, WK_3은 선형 연산으로 이루어져 있기 때문에 WK_1, WK_3를 구하기 위해선 선형 연산도 수행을 해줘야한다.

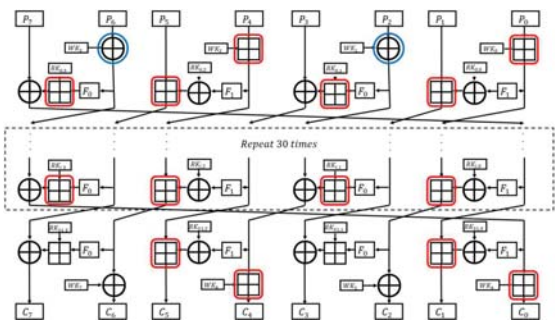

Fig. 1 Attack point of CPA

평문과 암호문 두 가지 모두 사용하는 부채널 분석의 경우 WK_1, WK_3가 k_{13}, k_{15}를 복구하는 것과 같기 때문에 비선형 연산만으로 원하는 암호문을 획득할 수 있다. 평문과 암호문을 모두 사용할 수 있는 환경에서는 라운드 함수 중 마지막 31번째 단계를 이용하는 것이 적절하다. 평문만 사용할 경우엔 선형 연산을 반드시 수행해줘야지만 원하는 암호문을 획득할 수 있다. 평문만 사용할 수 있는 환경이라면 라운드 함수 중 마지막 31번째 단계를 제외한 부분을 사용하는 것이 적절하다.

3.2.2 HIGHT 부채널 분석 실험

해당 논문에서는 ATmega128 보드에 8bits로 구현된 HIGHT 알고리즘을 사용하였으며, 공격 성공률 과 guessing entropy 각각 1,000번씩 분석하였다.

그림 2, 3[1]를 보면 각각 1,000번을 돌린 결과에 대해 확인할 수 있다. 그림 내부에 있는 F_0은 선형 함수만을 이용한 분석이며, F_1은 비선형 함수만을 이용한 분석이다.

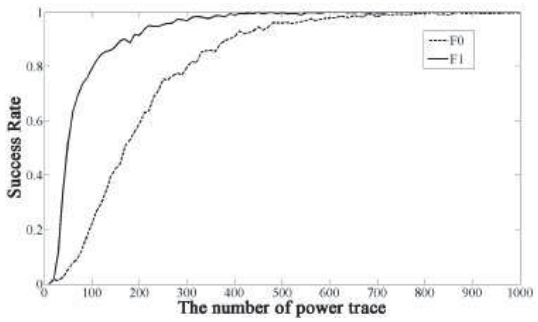

Fig. 2 Success rate of CPA for HIGHT

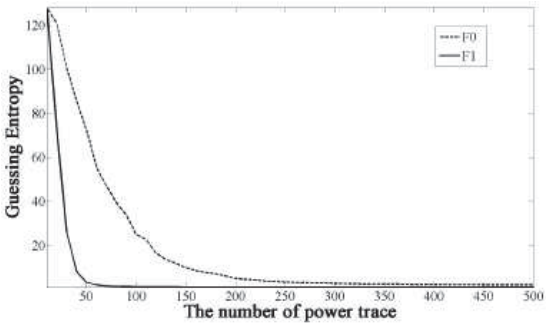

Fig. 3 Guessing entropy of CPA for HIGHT

1) 해당 그림은 원 논문에서 참조하였다.

그림 2는 공격 성공률을 나타내는 그림으로 공격 성공률이 99%이상이 되기 위해서 F_0는 730번 이상 F_1은 400번 이상인 것을 알 수 있다.

그림 3는 guessing entropy로 복잡도가 3이하가 되기 위해서 F_0는 270번 이상 F_1은 50번 이상인 것을 알 수 있다. 따라서 선형과 비선형 함수 모두 부채널 분석에 가능함을 증명하였으며 앞에서도 언급했듯이 부채널 분석에는 비선형 함수가 조금 더 용이한 것을 실험을 통해 다시 한 번 알 수 있었다.

3.3 LEA

2014년 CPA에 의해 LEA의 서브 키 두 쌍이 밝혀지면서[4], LEA 역시 전력 부채널 분석에 안전하지 않음이 알려졌다. 따라서 적어도 이러한 1차 분석에 대한 마스킹 기법을 적용하는 것이 필요하며, 이 과정에 있어 LEA의 설계 목적인 경량화, 고속화를 고려하는 것이 필요하다.

3.3.1 LEA에 대한 마스킹

LEA의 구조에서도 두 부분(XOR, Addition)에서 중간 값이 생성된다. 이 값을 통계적으로 분석하여 비밀 키를 탈취할 수 있다. 따라서 이 지점의 값에 난수를 취하여 비밀 키와의 상관관계를 제거하는 방식으로 마스킹 기법이 적용될 수 있다. 이러한 마스킹 기법의 적용에는 몇 가지 진행 중인 연구가 있다. 먼저, 더해지는 난수 자체의 안전성을 제기한 논문에서는 안전성을 검증한 TRNG(True Random Number Genorator)기반의 마스킹 기법을 제안했다.

다음으로 LEA가 가지는 ARX구조의 특징아래 두 종류의 연산 불 연산과 산술 연산간의 변환 기법이 필요하다. 이 변환 기법을 다른 논문에서는 이론상에서 효율적이라고 여겨졌던 룩업테이블 방식의 변환 기법이 실제 LEA가 목적으로 하는 플랫폼 위에서 비효율적임을 밝히고 마스킹 기법이 적용된 두 숫자를 기법을 해제하지 않고 연산하는 Secure Addition 기법을 적용한 실용적인 방법을 제안하고 구현한다. 이를 통하여 기존 AES 대비 마스킹 기법 적용에 의한 연산량 증가의 비효율성 문제를 완화했다.

IV. 결론

본 논문에서는 국산 암호 알고리즘에 적용된 부채널 분석에 대해 기존에 연구 된 사례들에 대하여 조사하였다. 그 결과로 국산 암호의 각 구조에 맞추어 부채널 분석이 시행될 수 있으며 부채널 분석 대응 기법 또한 국산 암호의 구조적 특징과 목적을 고려하여 효율적으로 적용되어야 함을 알 수 있다. 이후에는 또 다른 국산 암호 LSH, ARIA와 아직 표준화 작업이 진행되진 않았지만, 가장 최근에 나온 국산 블록암호인 CHAM에 대하여 본 논문에서 살펴본 바를 참고하여 부채널 분석을 해보고 그 대응 기법을 적용해 볼 예정이다.

[참고문헌]

[1] P. C. Kocher, "Timing Attacks on Implementation of Diffe-Hellman, RSA, DSS and Other Systems," Available: https://www.paulkocher.com/doc/TimingAttacks.pdf

[2] Y. S. Won, A. S. Park, D. G. Han, "Side Channel Analysis with Low Complexity in the Diffusion Layer of Block Cipher Algorithm SEED," *Journal of the Korea Institute of Information Security & Cryptology* 27(5), pp. 993-1000, Oct. 2017.

[3] T. J. Kim, Y. S. Won, J. H. Park, H. J. An, D. G. Han, "Side Channel Attacks on HIGHT and Its Countermeasures," *Journal of the Korea Institute of Information Security & Cryptology* 25(2), pp. 457-465, Apr. 2015

[4] Y. D. Kim, and H. S. Yoon, "First Experimental Result of Power Analysis Attacks on a FPGA Implementation of LEA," *IACR Cryptology ePrint Archive 2014*: 999. 2014