AES 혼합 구현 기법이 부채널 분석에 미치는 영향

원유승, 박명서, 이예림, 한동국
국민대학교 수학과 정보보안연구소
e-mail: mathwys87@kookmin.ac.kr

Effects of mixed AES implementation techniques against Side Channel Analysis

Yoo-Seung Won, Myung-Seo Park, Ye-Rim Lee, Dong-Guk Han
Dept of Math and CISI, Kook-Min University

요 약

스마트 디바이스 내에 내재된 암호 알고리즘의 이론적인 안전성이 보장되었더라도 연산 수행 시 소모되는 전력소모, 전자기파와 같은 물리적 정보와의 관계를 분석하는 부채널 공격에 대해 취약하다. 이에 대해 부채널 공격의 대표적 대응 기법으로 셔플링 기법과 마스킹 기법이 제안되었다. 특히 셔플링을 활용한 대응 기법은 효율성을 최대한 유지하면서 분석의 난이도를 높이는 기법으로 잘 알려져 왔다. 본 논문에서는 AES 대칭키 암호의 부채널 대응 기법으로 8가지 AES 구현 기법을 셔플링 하여 구현할 경우 부채널 분석에 미치는 영향에 대해 연구하였다. 실험 결과 실제 기대되어지는 분석 난이도는 셔플링을 하지 않은 것에 비해 64배 정도 공격 복잡도가 높아져야 하지만, 실제는 7배정도의 공격 복잡도 증가로 분석이 되었다.

1. 서론

수학적으로 안전하다고 알려져 있는 암호 알고리즘조차도 부가적인 정보 누출에 의해 비밀 키의 값을 알아 낼 수 있는 부채널 공격이 1996년 Paul Kocher에 의해 소개되었다.[1] 부채널 공격은 암호 알고리즘이 수행되는 동안 시간측정, 전력소모측정 또는 하드웨어적 강제에러주입 등의 부가적 정보를 이용하여 공격하는 방법이다. 이러한 부채널 공격에도 안전한 암호 알고리즘을 구현하기 위해 많은 암호학자들에 의해 대응 기법들이 연구되었다. 대표적인 대응 기법으로는 마스킹, 랜덤딜레이, 셔플링 등의 기법이 존재한다.

AES(Advanced Encryption Standard) 대칭키 암호[2]에 대한 셔플링 기법의 적용은 평문공격과 암호문 공격에 대응하기 위해 라운드 함수 중에서 일반적으로 처음 두 라운드와 마지막 두 라운드에 적용시킨다. 또한, 셔플링 기법은 셔플링 복잡도에 따라 부채널 공격의 대응 기법으로 널리 쓰이고 있다. 특히, 2차 이상의 전력분석에 대한 대응 기법을 효율적으로 구성하기 위해서 셔플링 기법이 많이 활용되어진다. 본 논문에서는 8가지의 다른 방법으로 구현된 AES 대칭키 암호를 매번 암호화 과정에서 랜덤하게 실행시키는 셔플링을 구현할 경우, 부채널 분석의 복잡도에 어떤 영향을 미치는지 연구를 하였다. 이론적인 공격 복잡도에 대한 기댓값은 64배 정도 공격 복잡도의 향상이 이루어져야 하지만, 실험 결과 그보다 훨씬 작은 7배 정도의 공격 복잡도 안에서 분석이 됨을 확인하였다. 그리고 실험 결과로부터 이론적 공격 복잡도 증가가 나타나지 않은 원인에 대해서 분석 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 부채널 공격 기법 중 하나인 상관전력분석공격과 일반적인 셔플링 대응 기법을 설명한다. 3장에서는 상관전력분석공격을 통해 AES 혼합 구현 기법의 분석 결과를 소개하고, 마지막으로 4장에서 결론을 맺는다.

2. AES에 대한 상관전력분석공격과 셔플링

2.1 AES 알고리즘 개요

AES는 블록 단위로 메시지를 처리하는 대칭키 블록암호알고리즘으로서 널리 사용되고 있는 표준 암호 알고리즘 중 하나이다. AES의 한 라운드는 SubBytes, ShiftRow, Mixcolumn, AddRoundKey 함수로 구성되어 있다. SubBytes 함수는 비선형성을 갖는 S-Box로 구성되어 바이트 단위로 치환을 수행하는 함수이고, ShiftRow 함수는 행 단위로 순환 시프트(Cyclic Shift)를 수행하는 단순 순열 함수이다. MixColumn 함수는 유한 체 $GF(2^8)$ 산술식을 활용한 치환으로 확산성을 제공하기 위해 열 단위로 혼합하는 함수이다. AddRoundKey 함수는 라운드

키와 XOR 연산을 수행하는 함수이다.

2.2 AES의 S-Box에 대한 상관전력분석공격

2.2.1 소비전력 모델링

각각의 디바이스에서 데이터를 처리할 때 소모되는 전력의 특징에 따라 나뉘는 전력 모델 중에 해밍 웨이트 모델(Hamming weight model)과 해밍 디스턴스 모델(Hamming distance model)이 통상적으로 쓰이고 있다.[3] 해밍 웨이트 모델은 데이터 처리하는 과정에서 전력을 소비한다는 가정에서 나온 모델로 '1'이 '0'보다 많은 전력을 소비한다는 사실에 기반을 두고 있고 해밍 디스턴스 모델은 데이터의 입력으로부터 출력이 되는 데이터의 상태천이에서 소비전력에 영향을 미치는 사실에 기반을 두고 있다.

2.2.2 S-Box 출력에 대한 상관전력분석공격

전력분석은 실제 전력 소모량이 옳은 키가 연산될 때 비례하지만 틀린 키가 연산될 때는 비례하지 않는 성질을 이용하여 분석한다. 여기서 상관전력분석공격(CPA, Correlation Power Analysis)[3]은 해밍 웨이트와 데이터의 전력파형 간의 상관계수를 구하여 가장 높은 상관계수를 가지는 것을 옳은 키로 추측하는 공격 방법을 말한다. 본 논문에서는 AES의 16개 바이트를 한 바이트 씩 나눠 키를 분석한다. 한 바이트 당 256개의 추측 키에 대한 S-Box 출력을 중간 값으로 하여 해밍 웨이트를 구한 후 미리 수집한 전력 파형간의 상관계수를 구하여 가장 높은 값을 가지는 추측 키를 옳은 키로 간주한다.

2.2.3 AES 셔플링 기법

일반적인 셔플링 기법의 적용은 암호 알고리즘 내의 함수 연산을 출력 값에 영향을 주지 않게 섞어준다. AES 대칭키 암호를 일반적인 셔플링 기법으로 적용을 하면 기법이 적용된 라운드의 SubBytes내 S-Box 연산 순서를 바꾸고, ShiftRow, MixColum 함수의 연산 순서를 바꾼다.

3. AES 혼합 구현기법에 대한 상관전력분석공격

본 장에서는 혼합된 AES 구현 기법인 8가지의 다른 방법으로 구현된 AES 대칭키 암호를 상관전력분석공격을 통해 분석을 한다. 상관계수가 높게 나타난 각각의 피크 포인트(Peak Point)에 대한 위치 분석을 하였다. 또한, 혼합된 AES 구현 기법과 단일 AES와 비교 분석을 통하여 셔플링 복잡도 이론의 값이 맞는지에 대해 실제 측정값을 통하여 확인하고, 이에 대한 실험 결과를 해석한다.

3.1 AES 혼합 구현 기법

AES 혼합 구현 기법은 AES를 다양한 방법으로 구현하여 랜덤하게 적용시키는 셔플링 기법을 사용하였다. AES 구현 방법은 8가지이고 <표 1>과 같이 구성되었다.

<표 1> AES 혼합 구현 방법 (총 8가지)

AES 구현 방법 이름	특징
Original1	S-Box 테이블을 256×1 로 참조
Original2	S-Box 테이블을 16×16 로 참조
Bertoni[4]	Bertoni제안한 방법으로 구현
T-Table[5]	AES를 T-Table로 구현
Original1 Macro	AES Original1의 함수를 모두 매크로로 구현
Original2 Macro	AES Original2의 함수를 모두 매크로로 구현
Bertoni Macro	AES Bertoni의 함수를 모두 매크로로 구현
T-Table Macro	AES T-Table의 함수를 모두 매크로로 구현

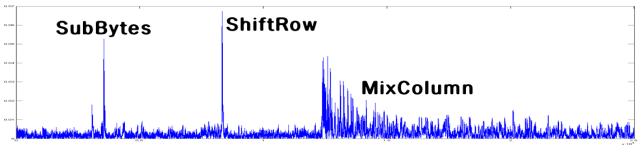
본 논문에서 분석할 혼합된 AES는 서로 다르게 구현된 8가지의 AES 대칭키 암호를 랜덤하게 적용시키기 위해 rand()함수를 사용하였다.

3.2 AES 혼합 구현기법에 대한 상관전력분석공격 결과 분석

AES 혼합 구현 기법을 분석하기 위해 혼합된 AES 코드와 8가지 각각의 단일 AES 코드는 다음과 같은 실험 환경에서 파형을 수집하여 분석을 하였다.

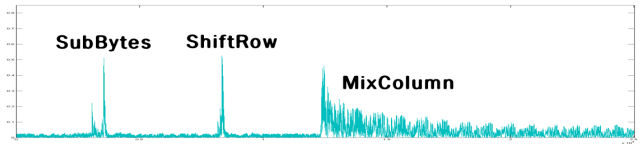
분석 타켓은 S-Box 출력 값이므로, 각각의 AES의 1라운드를 단순전력분석공격(SPA, Simple Power Analysis)[4]을 통해 파형을 수집하였다. 혼합된 AES는 100,000개의 파형을 수집하였고, 8가지의 단일 AES는 셔플링 복잡도에 따른 결과 분석을 하기 위해 각각 12,500개의 파형을 따로 수집하고, 상관전력분석공격을 적용하였다.

아래의 (그림 1~9)의 x축은 수집된 전력파형의 포인트 수를 의미하고, y축은 상관전력분석공격 결과인 상관계수를 의미한다. (그림 1)은 혼합된 AES 100,000개의 파형을 상관전력분석공격을 통하여 분석한 총 16바이트 중 피크 포인트의 분포가 뚜렷한 6번째 바이트에 대한 그림이다. (그림 2~9)는 각각의 구현 기법에 따라 12,500개의 파형으로 6번째 바이트에 대한 상관전력분석공격 결과를 나타내고 있다.

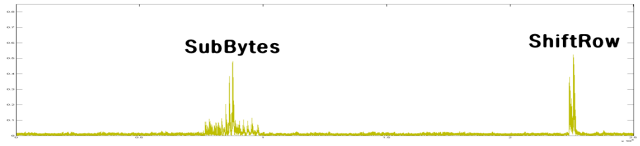


(그림 1) 혼합된 AES CPA 분석 결과

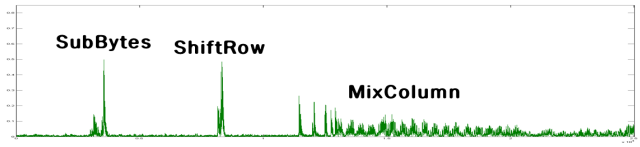
(그림 1)에서 보이는 상관계수에 대한 3개의 피크 포인트의 위치를 정확하게 분석하기 위해서 나머지 8가지의 AES를 상관전력분석공격을 적용하여 결과를 분석하였다. 아래 그림 8개는 각각의 AES에 대한 상관계수에 대한 피크 포인트이다.



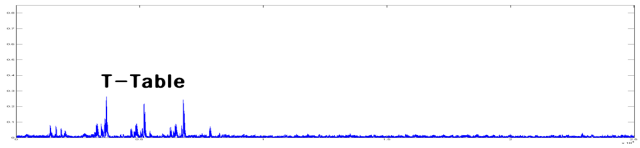
(그림 2) AES Original1 CPA 분석 결과



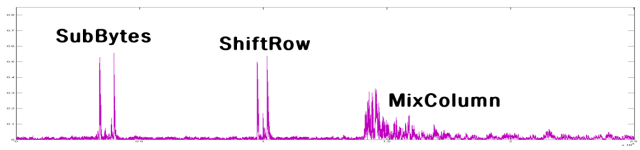
(그림 3) AES Original2 CPA 분석 결과



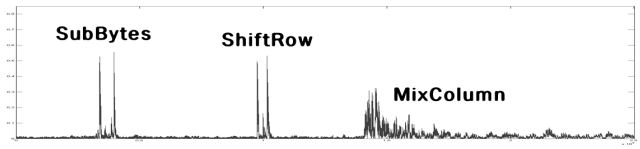
(그림 4) AES Bertoni CPA 분석 결과



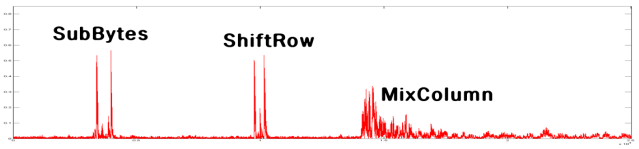
(그림 5) AES T-Table CPA 분석 결과



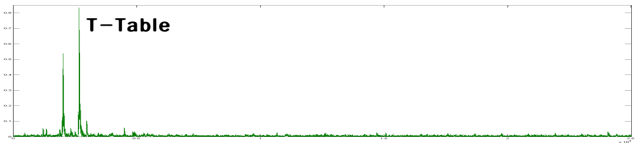
(그림 6) AES Original1 Macro CPA 분석 결과



(그림 7) AES Original2 Macro CPA 분석 결과



(그림 8) AES Bertoni Macro CPA 분석 결과



(그림 9) AES T-Table Macro CPA 분석 결과

(그림 1)에서 총 3가지 피크부분으로 분석 된 것은 각각 SubBytes, ShiftRow, MixColumn이다. 이에 대해 3가지로 나누어 설명한다.

(1) SubBytes 피크 부분

SubBytes부분을 살펴보면 (그림 2, 4, 6, 7, 8)에서 SubBytes 분석 지점이 동일 시간 위치에서 나타나는 것을 볼 수 있다. 이는 서로 다른 코드임에도 불구하고 동일한 시간 위치에 SubBytes가 수행이 되어 상관전력분석공격분석 시 중간 값이 정확히 분석되었다는 것을 알 수 있다.

(2) ShiftRow 피크 부분

ShiftRow부분은 (그림 2, 3, 4)에서 보이듯이 동일 시간 위치에서 ShiftRow가 수행되었다는 것을 알 수 있지만, 특이한 점은 (그림 3)에서 SubBytes부분이 수행된 부분이 다른 (그림 2, 4)와 동일한 시간 위치에서 수행되어 상관전력분석공격 분석이 더 용이하였다. (그림 6, 7, 8)의 ShiftRow도 동일 시간 위치에서 수행되었지만, 실제 혼합된 AES가 분석이 되었을 때, 소음(noise)으로 작용하여 분석이 되지 않았다.

(3) MixColumn 피크 부분

MixColumn부분은 Original2, T-Table, T-Table Macro를 제외하고는 모두 긴 꼬리를 무는 것처럼 분석이 되었다. 이는 MixColumn 구현 시 xtime을 연산하고 Mixcolumn이 AES 1라운드 수행 시 뒤쪽 부분에 위치하여 정렬이 맞지 않아 전체적으로 꼬리를 무는 것과 같이 분석이 되었다. 따라서 (그림 2, 4, 6, 7, 8)의 MixColumn 부분에 의해 혼합된 AES도 또한 피크가 나타났다.

3.3 AES 혼합 구현기법에 대한 성능 분석

기존의 상관전력분석공격으로 혼합된 AES를 분석하면 셔플링을 하지 않았을 때 분석 파형 개수가 m개이고 셔

플링 복잡도가 $\frac{1}{n}$이라 할 때, AES에 셔플링 기법이 적용된 파형을 분석하기 위해서는 이론적으로 $m \times n^2$의 파형이 필요하다.[5]

본 실험에서 적용한 셔플링 기법이 실질적으로 적용이 되었는지 알아보기 위하여 6번째 바이트에서 Ratio[1] 1.2 기준으로 혼합된 AES가 분석이 되는 최소파형 개수와 단일 AES 각각에 대한 최소파형 개수를 나타낸 표이다.

<표 3> <표 1>의 AES에 대한 상관전력분석공격분석 최소파형 개수

AES 구현방법 이름	6번째 바이트(단위 100개)
Original1	2
Original2	2
Bertoni	2
T-Table	43
Original1 Macro	1
Original2 Macro	1
Bertoni Macro	1
T-Table Macro	1
8가지 단일 AES 최소 파형 개수 평균	6.625
이론적 분석 파형 개수	424
혼합된 AES의 분석 최소 파형 개수	63

앞서 소개한 방법으로 나머지 8가지 AES 최소파형 개수의 평균에 대하여 이론적 분석 파형 개수가 계산되어진다. 본 논문에서의 셔플링 복잡도는 $\frac{1}{8}$이므로, 이론적 파형 개수는 최소파형 개수 평균의 64배가 되어야 한다. <표 3>를 참고하면 혼합된 AES의 분석 최소 파형 개수는 이론적 분석 파형 개수 42,400개의 $\frac{1}{7}$ 수준에서 분석된다는 것을 볼 수 있다. 즉, 8가지 구현 기법을 랜덤하게 적용할 경우, 공격 복잡도는 이론적인 64배의 증가를 시키지 못하고, 그보다 훨씬 적은 7배 정도 공격 복잡도로도 분석이 가능하다는 것이다.

이론적 분석 파형 개수보다 더 적은 파형으로 혼합된 AES가 분석된 원인은 다음과 같다. 8가지 구현기법을 통해 이론적인 64배 공격 복잡도를 얻기 위해서는 분석위치에 해당하는 S-Box 출력 전력 정보가 8가지의 구현에서 모두 다른 시간 위치에 존재해야 한다는 것이다. 하지만, 앞에서 살펴본 (그림 1~9)와 같이 S-Box 출력의 정보를 활용하는 SubBytes, ShiftRow, MixColumn 부분에 대한 전력 파형이 일부 같은 시간 위치에 존재하였다. 따라서 이론적인 64배 보다 훨씬 적은 공격 복잡도로 분석이 성공할 수 있었던 것이다.

1) Ratio는 $\frac{(옳은\ 키\ 상관계수)}{MAX(틀린\ 키\ 상관계수)}$를 의미하며, Ratio 1.0을 넘으면 옳은 키를 찾았다고는 할 수 있으나 틀린 키와 명확히 구분 짓기 위하여 Ratio 1.2를 택하였다.

4. 결론

본 논문에서는 다양한 구현 기법을 활용한 셔플링 기법이 부채널 분석에 미치는 영향에 대해 연구하였다. AES 대칭키 암호의 8가지 구현 기법을 적용하여 셔플링을 하였을 경우, 이론적인 64배 공격 복잡도가 아닌 7배 정도의 공격 복잡도 향상이 나타남을 실험을 통해 검증하였다. 따라서 혼합된 AES 구현 기법은 부채널 공격 관점에서 볼 때 기존 셔플링 기법보다 효율성이 떨어지므로 현실적인 대응 기법으로는 적합하지 않다.

Acknowledgments

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012-007285)

참고문헌

[1] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” CRYPTO, LNCS 1666, 1999, pp. 388-397.

[2] National Institute Standards and Technology: Advanced Encryption Standard (AES). Publication 197 (2001)

[3] E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In M. Joye and J.-J. Quisquater, editors, Cryptographic Hardware and Embedded Systems – CHES 2004, pages 16 - 29.

[4] Guido Bertoni, Luca Breveglieri, Pasqualina Fragneto, Macro Macchetti, Stefano Marchesin, Efficient Software Implementation of AES on 32-Bit Platforms - CHES 2002, pages 159-171, 2002

[5]Joan Daemen, Vincent Rijmen, The Rijndael Block Cipher. Version 2, September (1999)

[6] S. Mangard, E. Oswald, and T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, 2010.

[7] Roger S. Pressman ″Software Engineering A Practltiners′ Approach″ 3rd Ed. McGraw Hill