情報保護學會論文誌
第19卷 第2號, 2009. 4

중간 일치 분석법에 기반한 AES에 대한 부채널 공격*

김 종 성,† 홍 석 희,‡ 이 상 진
고려대학교 정보보호연구원

Side-Channel Attacks on AES Based on Meet-in-the-Middle Technique*

Jongsung Kim,† Seokhie Hong,‡ Sangjin Lee
CIST, Korea University

요 약

본 논문에서는 블록암호 분석기법인 중간 일치 분석법을 이용한 새로운 부채널 공격 방법을 소개한다. 이 공격 기법을 이용하여, 축소 마스킹을 사용하는 미연방 표준 블록암호 AES에 대한 부채널 공격을 소개한다. 즉, 10개 라운드에 축소 마스킹을 사용하는 AES는 기 제안된 4-라운드 중간 일치 함수에 기반한 부채널 공격에 취약함을 보인다. 이는 전체 12-라운드 192-비트 키 AES가 부채널 공격에 안전하기 위해서는 전체 12개의 라운드에 마스킹을 이용해야함을 나타낸다. 본 논문의 결과는 10개 라운드에 축소 마스킹을 사용하는 AES에 대한 첫 분석 결과이다.

ABSTRACT

In this paper we introduce a new side-channel attack using block cipher cryptanalysis named meet-in-the middle attack. Using our new side-channel technique we introduce side-channel attacks on AES with reduced masked rounds. That is, we show that AES with reduced 10 masked rounds is vulnerable to side channel attacks based on an existing 4-round function. This shows that one has to mask the entire rounds of the 12-round 192-bit key AES to prevent our attacks. Our results are the first ones to analyze AES with reduced 10 masked rounds.

Keywords: Side-channel attacks, Meet-in-the-middle attacks, AES

I. 서 론

부채널 공격[1]에 안전한 암호 알고리즘의 구현으로 각 라운드별 마스킹을 덧붙이는 방법이 가장 널리 사용되고 있다. 하지만, [2]는 매 라운드마다 마스킹을 덧붙이는 방법은 암호 알고리즘에 대한 구현 비용을 많이 증감 시킬 수 있다는 문제점을 지적하고, 매 라운드가 아닌 처음과 마지막 몇 라운드에만 마스킹을 사용하는 방법을 소개하였다. 그러한 예로서 [2,3]는 DES에 대한 처음과 마지막 3개 또는 4개의 라운드에만 마스킹(총 6개 또는 8개의 라운드에 대한 마스킹)을 덧붙이는 방법을 제시하였지만, [3,4]에 의하면, 6개 또는 8개의 라운드에 대한 마스킹을 이용하는 DES는 부채널 공격에 여전히 취약하다. 또한, [5]는 처음과 마지막 3개 또는 4개의 라운드에 마스킹(총 6개 또는 8개의 라운드에 대한 마스킹)을 사용하는 AES 또한 부채널 공격에 취약함을 보이고, 부채널 공격에 안전하기 위해서는 최소 10개의 라운드에 마스킹을 사용하여 AES를 구현할 것을 권장하였다.

본 논문에서는 블록암호 분석기법인 중간 일치 분석법을 이용한 새로운 부채널 공격 방법을 소개한다. 이 공격 기법을 이용하여 최초로 10개의 라운드에 마스킹을 사용하는 AES에 대한 부채널 공격을 고안하는데 성공하였다. 본 논문의 공격 결과는 다음과 같다: 처음과 마지막 각각 5개 라운드에 마스킹(총 10개의 라운드에 대한 마스킹)을 덧붙인 AES는 2^{32+m}

접수일(2008년 6월 24일), 수정일(2008년 11월 7일), 게재확정일(2009년 3월 23일)
* 이 연구에 참여한 연구자는 '2단계 BK21사업'의 지원비를 받았음.
† 주저자, joshep@cist.korea.ac.kr
‡ 교신저자, hsh@cist.korea.ac.kr

선택 평문의 데이터 복잡도, 2^{203-n} AES 128-bit 블록 메모리 복잡도와 $2^{206.4-n}+2^{44.4+n}$ AES 암호화 과정의 시간 복잡도로 공격된다 (만약 $0 \le n \le 22$이면, $m=0$이고, 그렇지 않으면, $m=\lceil n-22 \rceil$임). 만약 $n=82$이면, 2^{92}의 선택 평문의 데이터 복잡도, 2^{121} AES 128-bit 블록 메모리 복잡도와 $2^{126.4}$ AES 암호화 과정의 시간 복잡도를 요구한다. 이는 전체 12-라운드 192-비트 키 AES와 14 라운드 256-비트 키 AES가 최소 12개의 라운드에 마스킹을 덧붙여 구현해야만 본 논문의 공격을 피할 수 있음을 나타낸다. 또한, 본 논문의 부채널 공격 기법이 8개 라운드에 마스킹을 사용하는 AES에 더 작은 공격 복잡도로 적용됨을 보인다.

본 논문의 구성은 다음과 같다. 2, 3장에서 AES 알고리즘 및 중간 일치 분석법을 이용한 새로운 부채널 공격 기법을 소개한다. 4장에서는 10개 라운드와 8개 라운드에 마스킹을 사용하는 AES 알고리즘에 대한 부채널 공격을 소개한다. 끝으로, 5장에서 결론을 맺는다.

II. AES 알고리즘 소개

AES[6]는 가장 널리 사용되고 있는 표준 암호 알고리즘 중 하나로 키 길이에 따라 AES-128, AES-192, AES-256으로 나뉘어진다. AES-128, AES-192, AES-256은 각각 128, 192, 256 비트 키를 사용하며, 10, 12, 14 라운드로 구성된 128 비트 블록 암호이다. AES의 암호화 과정 상의 128 비트 상태 값은 [그림 1]과 같이 16개 바이트 Xij로 이루어진 4X4 행렬로 나타낼 수 있다. AES의 한 라운드는 SubByte(SB), ShiftRow(SR), MixColumn(MC), AddRoundKey(ARK) 함수를 차례대로 사용하며, 첫 번째 라운드 전에 ARK 함수를 적용하고 (이 단계에서 사용되는 키는 화이트닝 키라 부름 - K^0), 마지막 라운드에서 MC 함수를 생략한다. 각 함수는 다음과 같이 동작한다.

- SB 함수는 각각의 바이트에 동일한 비선형 S-박스를 적용한다.
- SR 함수는 행 0, 1, 2, 3을 왼쪽으로 각각 0, 1, 2, 3 바이트 순환 이동시킨다.
- MC 함수는 선형 변환으로 4 바이트로 구성된 각 열을 변환시키는 4X4 행렬로 $GF(2^8)$ 위에서 연산한다.
- ARK 함수는 키와 상태값의 비트별 합 연산을 수행한다.

X_{11}	X_{12}	X_{13}	X_{14}
X_{21}	X_{22}	X_{23}	X_{24}
X_{31}	X_{32}	X_{33}	X_{34}
X_{41}	X_{42}	X_{43}	X_{44}

[그림 1] AES의 128 비트 상태 값

본 논문에서 다루는 부채널 공격은 AES의 키 스케줄의 동작 과정과는 무관하게 적용되므로, AES에 대한 키 스케줄의 묘사는 생략한다. 본 논문의 공격 대상은 AES-192와 AES-256이다.

III. 중간 일치 공격을 이용한 새로운 부채널 공격

본 장에서는 블록암호 분석기법 중 하나인 중간 일치 공격[7]이 헤밍 웨이트 측정값을 이용하는 부채널 공격으로 확장되는 방법을 최초로 소개한다. 먼저, [7]에 소개된 중간 일치 공격에 대한 r 라운드[1] (축소) 블록암호에 대한 일반적인 공격 방법은 다음과 같다 (r은 전체 라운드의 축소 라운드의 수일 수도 있음; 예를들어, 전체 12 라운드 AES-192 중 r=7 라운드 공격[7], 즉, 7 라운드 공격의 가정은 공격자가 임의의 평문에 대한 7 라운드 후의 암호화된 값을 획득할 수 있음).

1. 먼저 블록암호의 마지막 (r-1) 라운드에 대한 처음 입력의 한 바이트 a와 마지막 출력의 한 바이트 b의 대응 함수를 찾는다 (의존하는 상수 비트 수에 따라 함수의 개수가 결정됨). 예를 들어, 바이트 a와b가 고정된 바이트 상수값 c1, c_2에 대해 $b=S(a+c_1)+S(a+c_2)$의 식으로 표현이 되면, a, b, c_1, c_2 모두 각각 8 비트이고, 각 (c_1, c_2)의 값에 a에서 b로 대응하는 하나의 함수가 결정되므로, 총 가능한 함수의 개수는 2^{16}이다 (여기서, S는 S 박스를 나타내며, c_1, c_2의 값은 키의 정보를 포함할 수 있으며, a와는 독립적인 값이다).

1) r 라운드 또는 r-라운드는 r 개의 라운드를 의미하며, 라운드 r은 r번째 라운드를 의미한다.

2. 그 후 모든 가능한 대응 함수 각각에 대한 출력값을 함수별로 저장한다 (함수별 입력값 0-255에 대응하는 출력값을 순서대로 저장; 각 함수는 한 바이트 위에서 정의된 함수이므로 0-255까지의 입력값을 갖음).
3. 대응 함수의 입력값 0-255가 순서대로 나오도록 첫번째 라운드의 해당되는 키를 추측하여 평문 집합을 구성한다 (평문 집합을 구성하는 방법은 두번째 라운드의 입력 바이트 a는 0-255의 값을 나머지 바이트들은 상수값을 선택하여, 두번째 라운드의 입력 바이트 a와 관계되는 첫번째 라운드의 키를 추측하여 한 라운드 복호화하여 평문 집합을 구성함).
4. 구성한 평문 집합에 대한 r 라운드 후의 암호문 집합을 획득한다. 대응 함수의 출력값에 해당하는 바이트에 대한 획득한 암호문 집합의 바이트의 값들을 축출한다.
5. 축출한 바이트 값의 리스트가 단계 2에서 저장한 출력값 리스트와 일치하는 대응 함수가 존재한다면, 추측한 키를 올바른 키로 출력한다. 그렇지 않으면, 단계 3으로 돌아가 또 다른 키를 추측하여 단계 3, 4, 5를 수행한다.

위의 단계 1과 2는 선 계산 작업(pre-computation process)으로 이루어지며, 단계 3, 4, 5는 암호문 집합 획득(online-computation process)과 암호문 집합에 대한 해당 바이트의 분류, 비교 작업으로 이루어진다. 대응 함수의 총 입력값의 개수가 256개이고, 각 출력값이 일치할 확률은 2^{-8}이므로 단계 3에서 추측한 키가 틀린 키임에도 불구하고 단계 5에서 올바른 키로 출력할 확률은 $(2^{-8})^{256}\times t$의 값이 된다 (t는 단계 2에서 저장한 함수의 총 개수임). 하지만, 올바른 키를 추측한 경우 위 알고리즘은 반드시 추측한 키를 출력한다. 이 방법은 [7]에 소개된바 있다.

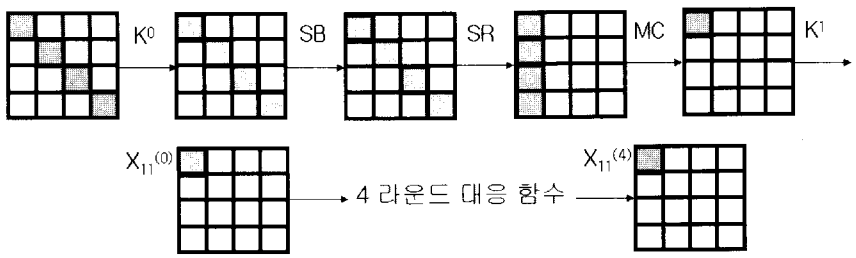
이제, 위의 중간 일치 공격이 부채널 공격으로 확장될 수 있음을 보인다. 처음 r 라운드에 축소 마스킹을 사용하는 블록암호에 대한 중간 일치 공격을 이용한 부채널 공격은 다음과 같다 (블록암호의 전체 라운드 수는 r 보다 큼).

1. 위의 단계 1과 동일함 (대응 함수는 축소 마스킹이 없는 오리지널 블록암호의 (r-1) 라운드 함수임).
2. 그 후 모든 가능한 대응 함수 각각에 대한 출력값의 비트별 헤밍 웨이트를 함수별로 저장한다 [8,9] (저장시, 함수별 입력값 0-255에 대응하는 출력값의 헤밍 웨이트를 순서대로 저장함; 헤밍 웨이트는 비트별 헤밍웨이트를 의미함, 즉, 1 바이트에 대한 헤밍웨이트는 0-8까지의 값을 가질 수 있음).
3. 위의 단계 3과 동일함 (평문 집합 구성시, 마스킹이 없는 오리지널 블록암호에서 수행함).
4. 구성한 평문 집합에 대한 r 라운드 후의 해당 바이트의 암호화된 평문 집합의 헤밍 웨이트를 측정한다 여기서, 해당 바이트는 단계 1, 2의 대응 함수에 대한 출력 바이트를 의미한다 (부채널 분석으로 해당 바이트의 헤밍 웨이트를 측정할 수 있음).
5. 측정한 헤밍 웨이트 리스트가 단계 2에서 저장한 헤밍 웨이트 리스트와 일치하는 대응 함수가 존재한다면, 추측한 키를 올바른 키로 출력한다. 그렇지 않으면, 단계 3으로 돌아가 또 다른 키를 추측하여 단계 3, 4, 5를 수행한다.

대응 함수의 총 입력값의 개수가 256개이고, 각 출력값의 해밍 웨이트가 0부터 8 사이이므로, 잘못 추측된 키임에도 불구하고 위 알고리즘이 추측한 키를 출력할 확률은 $(2^{-3})^{256}\times t$의 값 보다 작다 (t는 단계 2에서 저장한 함수의 총 개수임). 하지만, 올바른 키를 추측한 경우 위 알고리즘은 반드시 추측한 키를 올바른 키로 출력한다. 이유는 블록암호의 처음 r 라운드에 마스킹을 사용하였기 때문이다. 이는 r 라운드 이후의 암호화된 평문의 값은 오리지널 블록 암호의 처음 r 라운드 이후의 암호화된 평문의 값과 동일하다는 것을 나타낸다. 자세한 확률 계산과 공격 복잡도는 다음 장의 AES에 대한 부채널 공격에서 다룬다.

IV. 10 라운드 마스킹을 사용하는 AES에 대한 부채널 공격

본 장에서는 처음과 마지막 5 라운드 각각에 마스킹을 사용하는 AES-192와 AES-256에 대한 부채널 공격을 소개한다 (만약 AES-128에 처음과 마지막 5 라운드 각각에 마스킹을 사용한다면, 이는 AES-128 전체 라운드에 마스킹을 사용하는 것을 의미함). 공격은 FSE 2008[7]에 소개된 4 라운드 대응 함수를 이용한다.



〔그림 2〕 처음 5 라운드에 마스킹을 사용하는 AES에 대한 부채널 공격

4.1 AES에 대한 4 라운드 대응 함수 및 선 계산 작업

Property 17〔7〕. 첫번째 바이트 $X_{11}(0)$이 active 바이트이고 나머진 모두 passive 바이트인 256개의 입력값을 고려하자.[2] 이 입력값들에 대한 4 라운드 후의 출력값들을 얻는다 (4 라운드 후의 첫번째 바이트를 $X_{11}(4)$로 표기함). 그러면, $X_{11}(0)$에서 $X_{11}(4)$로 가는 대응 함수는 25개의 고정된 바이트의 값에 의해 결정된다 (25개의 고정된 바이트 값은 $X_{11}(0)$ 값과는 독립적임).

Property 1의 4 라운드 함수는 앞장의 단계 1에서의 예제 함수와 비슷한 형태를 가진다 (자세한 함수의 형태는 〔7〕를 참조). Property 1에 의하면, 25개의 고정된 바이트 각각의 값에 대해 $X_{11}(0)$에서 $X_{11}(4)$로의 대응 함수를 선계산하여 테이블에 저장하면, 첫번째 바이트 $X_{11}(0)$이 active 바이트이고 나머진 모두 passive 바이트인 256개의 입력값을 갖는 입력 집합에 대한 4 라운드 대응 함수는 반드시 테이블에 저장된 함수 중 하나와 동일하다.

〔7〕에서는 앞 장에서 묘사한 중간 일치 공격과 같이 모든 가능한 대응 함수에 대한 출력값들을 함수별로 저장하였다. 하지만, 본 논문의 공격에서는 모든 가능한 대응 함수의 출력값에 대한 헤밍 웨이트를 함수별로 테이블에 저장한 후 (각 함수별로 active 입력 바이트의 값은 0부터 255 순서대로 취하여 대응되는 출력 헤밍 웨이트 값을 저장함), 잘 구성된 선택 평문 집합에 대한 저장 테이블 함수 출력값에 대응하는 바이트의 헤밍 웨이트를 측정하여, 저장된 테이블의 함수별 헤밍 웨이트와 비교함으로써 공격이 이루어진다. 함수에 대한 헤밍 웨이트 테이블을 형성하기 위해서는 25개의 고정된 바이트 각각의 값에 대해 2^8개의 출력값을 계산해야함으로 $2^8\times(2^{25})^8=2^{208}$ 4 라운드 암호화 선 계산량이 필요하며 ($2^{206.4}$ AES 암호화 과정), $2^{-5}\times2^{208}=2^{203}$ AES 128-bit 블록 메모리양이 필요하다 (하나의 출력값에 대응하는 헤밍 웨이트 값이 0부터 8이므로 각 출력값에 대한 4 비트 메모리, 즉 2^{-5} AES 128-비트 블록 메모리가 요구됨).

4.2 처음과 마지막 5 라운드 각각에 마스킹을 사용하는 AES에 대한 부채널 공격 알고리즘

위의 4 라운드 함수의 성질을 처음과 마지막 5 라운드 각각에 마스킹을 사용하는 AES-192 또는 AES-256의 라운드 2-5에 적용한다면 (라운드 수는 1부터 시작함), 처음 5 라운드 마스킹을 사용할 때나 마스킹을 사용하지 않는 오리지널 AES는 동일한 평문에 대해 동일한 5 라운드 출력값을 가지므로, 다음과 같은 과정으로 부채널 공격을 할 수 있다 (그림 2 참조; 아래의 공격은 마지막 5 라운드의 마스킹에 영향을 받지 않음).

1. X_{11}, X_{22}, X_{33}, X_{44} 바이트는 모든 값을 취하고 나머지 바이트는 고정된 상수값을 취하는 2^{32}개의 평문으로 이루어진 평문 집합을 구성한다.
2. X_{11}, X_{22}, X_{33}, X_{44} 바이트에 해당하는 32-비트 화이트닝 키 K^0_{11}, K^0_{22}, K^0_{33}, K^0_{44}와 X_{11}에 해당하는 8-비트 첫번째 라운드 키 K^1_{11}를 추측하여 (그림 2의 어두운 부분의 K^0와 K^1), 라운드 1의 출력값의 형태가 첫번째 바이트 $X_{11}^{(0)}$이 active 바이트이고 (그림 2의 어두운 부분 $X_{11}^{(0)}$) 나머진 모두 passive 바이트가 되는 2^8개의 평문을 평문 집합으로부터 축출한다 (2^8개의 평문 축출시, 마스킹이 없는 오리지널 AES에서 수행).[3]
3. 축출한 2^8개의 평문에 대해 5 라운드 후의

2) Active 바이트는 0-255 값 모두를 한번씩 취한다. Passive 바이트는 active 바이트가 0-255 값 모두를 한번씩 취하는 동안 동일한 상수값을 취한다.

〔표 1〕 AES-192에 대한 기존의 중간 일치 공격과 본 논문의 부채널 공격 비교

	기존의 중간 일치 공격[7]	본 논문의 부채널 공격
대응 함수의 라운드 수	4 라운드	4 라운드
이용하는 대응 함수의 정보	출력값	출력값의 해밍웨이트
공격 라운드 수	7 라운드	전체 12 라운드 (처음과 마지막 5 라운드 각각에 마스킹 사용)
데이터 복잡도	2^{92} 선택 평문	2^{92} 선택 평문
시간 복잡도	2^{150} 암호화 과정	$2^{126.4}$ 암호화 과정
메모리 복잡도	2^{148} AES 128-비트 블록	2^{121} AES 128-비트 블록

$X_{11}^{(4)}$에 해당하는 바이트의 출력값에 대한 헤밍 웨이트를 측정한다 (그림 2의 어두운 부분 $X_{11}^{(4)}$).

4. 만약 측정한 헤밍 웨이트 리스트와 일치하는 함수가 저장 테이블에 존재한다면, 추측한 키를 올바른 키로 출력하고 종료한다. 그렇지 않다면, 단계 2로 돌아가 또 다른 키를 추측하여 단계 2, 3, 4를 수행한다.

4.3 공격 복잡도 계산

이 공격은 232의 선택 평문과 약 $2^{40}\times2^{8}\times2^{-5.6}=2^{42.4}$ AES 암호화 과정을 요구한다 (위 공격 알고리즘의 시간 복잡도는 단계 2에서 가장 큰 비중을 차지함, 단 $2^{-5.6}\approx\frac{4(Bytes)}{16(Bytes)}\times\frac{1(Ro.)}{12(Ro.)}$, Ro.=Rounds). 만약 올바른 키가 단계 2에서 추측이 되었다면, Property 1에 의해서 이 공격은 올바른 키를 출력한다. 그렇지 않은 경우, 추측한 키를 출력할 확률은 $(2^{-3})^{256}\times(2^{8})^{25}\times(2^{8})^{5}=2^{-528}$이다 ($(2^{-3})^{256}$은 저장한 하나의 대응 함수의 출력값에 대한 헤밍 웨이트 리스트가 측정한 헤밍 웨이트 리스트와 일치할 확률이고, $(2^{8})25$은 저장한 대응 함수의 개수이고, $(2^{8})5$은 단계 2에서 추측하는 키의 개수를 나타냄). 이는 99% 이상의 확률로 이 공격은 올바른 키를 출력하고 종료함을 의미한다. 따라서, 테이블 형성 과정의 선 계산량을 포함하면, 이 공격은 2^{32}의 선택 평문의 데이터 복잡도, 2^{203} AES 128-bit 블록 메모리 복잡도와 $2^{206.4}$ AES 암호화 과정의 선 계산량과 $2^{42.4}$ AES 암호화 과정의 시간 복잡도를 요구한다 (선 계산량은 4.1절 참조).

4.4 시간-메모리 보완 분석기법에 기반한 AES에 대한 부채널 공격

위 공격은 AES-256에 대한 전수조사 공격 보다 빠르지만, 높은 선 계산량으로 인하여 AES-192의 전수조사 공격보다 빠르지 못하다. 하지만, 시간-메모리 보완 공격을 이용하여 좀 더 많은 데이터량으로 AES-192의 전수조사 공격보다 빠른 공격법을 고안할 수 있다. 즉, 테이블에 저장하는 함수의 개수를 줄이는 대신에 위 공격의 평문 집합을 여러개 테스트 함으로써 선 계산량을 줄일 수 있다. [7]에 의하면, 저장하는 함수의 개수를 n_1 비율로 줄이고 각 추측한 키에 대해 테스트하는 2^{8}개의 평문을 갖는 테스트 평문 집합의 개수를 n2 비율로 늘렸을 때, $n_2=4n_1$이면 98%의 성공확률로 위 공격을 수행할 수 있다 (주의: 위의 공격은 2^{32}개의 평문을 갖는 평문 집합을 이용하지만, 각 테스트하는 평문 집합은 2^{8}개의 평문을 갖는 테스트 평문 집합을 이용함). 또한, 단계 1에서 구성한 2^{32}개의 평문을 갖는 평문 집합은 2^{8}개의 평문을 갖는 테스트 평문 집합을 2^{24}개 구성할 수 있다. 이는 하나의 테스트 평문 집합이 24-비트 ($X_{10}^{(0)}$, $X_{20}^{(0)}$, $X_{30}^{(0)}$)의 하나의 값에 의해 만들어지기 때문이다 (($X_{10}^{(0)}$, $X_{20}^{(0)}$, $X_{30}^{(0)}$)는 라운드 2의 입력 바이트들임). 따라서, 저장하는 함수의 개수를 2n 비율로 줄이고, 2^{8}개의 평문을 갖는 테스트 평문 집합의 개수를 2^{n+2} 비율로 늘리면, 시간-메모리 보완 공격에 의해 본 공격은 98%의 성공확률로 2^{32+m}의 선택 평문의

3) 이 과정은 앞 장의 단계 3에서 설명한 것과 같이, 라운드 1의 출력값을 첫번째 바이트 X11(0)이 active 바이트이고(0부터 255 순서대로 취함) 나머진 모두 passive 바이트가 되도록 먼저 선택한 후에 추측한 키에 대해 선택한 값을 한 라운드 복호화하여 평문 집합으로부터 28 평문을 축출한다.

데이터 복잡도, 2^{203-n} AES 128-bit 블록 메모리 복잡도와 $2^{206.4-n}$ AES 암호화 과정의 선 계산량과 $2^{44.4+n}$ AES 암호화 과정의 시간 복잡도를 요구한다. (만약 $0 \le n \le 22$이면, $m=0$이고, 그렇지 않으면, $m = \lceil n-22 \rceil$임). 만약 $n=82$이면, 2^{92}의 선택 평문의 데이터 복잡도, 2^{121} AES 128-bit 블록 메모리 복잡도와 $2^{126.4}$ AES 암호화 과정의 시간 복잡도를 요구한다. [표 1]은 AES-192에 대한 기존의 중간 일치 공격과 본 논문의 부채널 공격을 비교한 표이다. [표 1]에 의하면, 공격 복잡도와 공격 라운드 수 관점에서 본 논문의 부채널 공격이 기존의 중간 일치 공격에 비해 월등하다.

4.5 처음과 마지막 4 라운드 각각에 마스킹을 사용하는 AES에 대한 부채널 공격

처음과 마지막 4 라운드 각각에 마스킹을 사용하는 AES도 비슷한 방법에 의해 공격이 된다.

Property 2[10]. 첫번째 바이트 $X_{11}^{(0)}$이 active 바이트이고 나머진 모두 passive 바이트인2^{56}개의 입력값을 고려하자. 이 입력값들에 대한 3 라운드 후의 출력값을 얻는다 (3 라운드 후의 첫번째 바이트를 $X_{11}^{(3)}$로 표기하자). 그러면, $X_{11}^{(0)}$에서 $X_{11}^{(3)}$로 가는 대응 함수는 9개의 고정된 바이트의 값에 의해 결정된다 (주의: Property 2는 [10]에서 처음 소개되었고, Property 1은 [7]에서 처음 소개되었음).

Property 2를 라운드 2-4에 적용하고 위 공격 알고리즘과 같이 화이트닝 키와 첫번째 라운드 키의 40 비트를 추측한 후 평문 집합에 대한 4 라운드 출력값에 대한 헤밍 웨이트를 측정하여 키를 찾을 수 있다. 시간-메모리 보완 공격까지 적용을 하면, 8 라운드 마스킹을 사용하는 AES에 대한 부채널 공격은 98%의 성공확률로 2^{32+m}의 선택 평문의 데이터 복잡도, 2^{75-n} AES 128-bit 블록 메모리 복잡도와 $2^{78.4-n}$ AES 암호화 과정의 선 계산량과 $2^{44.4+n}$ AES 암호화 과정의 시간 복잡도를 요구한다 (m은 4.4절의 m과 동일함).

주의: 4.2절 공격 알고리즘의 단계 3을 제외한 모든 단계는(선 계산으로 만든 테이블 포함) 컴퓨터의 프로그래밍 작업으로 수행하며, 단계 3은 하드웨어적인 부채널 파형 분석에 의해 수행한다(파형 분석을 통한 헤밍 웨이트 측정).

V. 결 론

본 논문에서는 블록암호의 분석기법인 중간 일치 공격을 이용하여 새로운 부채널 공격을 제안하고, 제안된 부채널 공격법을 적용하여 최초로 10 라운드 마스킹을 사용하는 AES-192와 AES-256이 부채널 공격에 취약함을 보였다. 본 논문의 결과는 AES가 부채널 공격에 안전하기 위해서는 최소 12 라운드 또는 전체 라운드에 마스킹을 사용해야 함을 나타낸다.

본 논문에서 소개한 중간 일치 분석법을 이용한 부채널 공격법은 기존에 제안된 바 없으며, 블록암호 분석기법을 이용한 부채널 공격 범위[4,5,8,11]를 확장시켰다. 또한, 이 방법을 이용하여 10 라운드 마스킹을 덧붙인 AES에 대한 부채널 공격이 가능함을 최초로 보였다. 비록 본 논문의 결과는 축소 마스킹을 사용하는 AES에 대한 이론적인 분석 결과이지만, 본 논문에서 제안한 부채널 공격은 확산 효과가 느린 다른 블록암호에(예: CRYPTON, SQUARE) 축소 마스킹 기법을 사용할 때 유용하게 사용될 수 있으므로, 축소 마스킹 기법을 이용한 블록암호 구현시 본 논문의 부채널 공격에 대한 안전성을 검토할 것을 권장한다.

참 고 문 헌

[1] P.C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO'99, LNCS 1666, pp. 388-397, 1999.

[2] M.L. Akkar and L. Goubin, "A generic protection against high-order differential power analysis," FSE 2003, LNCS 2887, pp. 192-205, 2003.

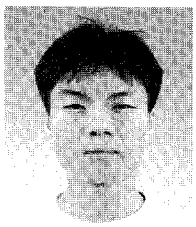
[3] M.L. Akkar, R. Bevan, and L. Goubin, "Two power analysis attacks against one-mask methods," CHES 2004, LNCS 3156, pp. 332-347, 2004.

[4] H. Handschuh and B. Preneel, "Blind differential cryptanalysis for enhanced power attacks," SAC 2006, LNCS 4356, pp. 163-173, 2007.

[5] A. Biryukov and D. Khovratovich, "Two new techniques of side-channel cryptanalysis," CHES 2007, LNCS 477, pp. 195-208, 2007.

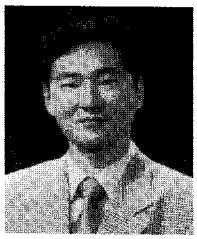
[6] J. Daemen and V. Rijmen, The design of Rijndael: AES - the advanced encryption standard, Springer, Mar. 2002.
[7] H. Demirci and A.A. Selcuk, "A meet-in-the-middle attack on 8-round AES," FSE 2008, LNCS 5086, pp. 116-126, 2008.
[8] K. Schramm and C. Paar, "A collision-attack on AES: combining side channel- and differential-attack," CHES 2004, LNCS 3156, pp. 163-175, 2004.
[9] K. Schramm and C. Paar, "Higher order masking of the AES," CT-RSA 2006, LNCS 3860, pp. 208-225, 2006.
[10] H. Gilbert and M. Minier, "A collision attack on 7 rounds of Rijndael," In the third AES Candidate Conference, pp. 230-241, Apr. 2000.
[11] R.C.-W. Phan and S.M. Yen, "Amplifying side-channel attacks with techniques from blockcipher cryptanalysis," CARDIS 2006, LNCS 3928, pp. 135-150, 2006.

〈著者紹介〉



김 종 성 (Jongsung Kim) 정회원
2000년 8월: 고려대학교 수학과 학사
2002년 8월: 고려대학교 수학과 석사
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 박사
2007년 2월: 고려대학교 정보보호대학원 박사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 연구교수
〈관심분야〉 대칭키 암호의 분석 및 설계, 멀티미디어/유비쿼터스 정보보호, 부채널 공격

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 2월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 2월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보보호대학원 조교수
2008년 9월~현재: 고려대학교 정보경영공학전문대학원 부교수
〈관심분야〉 대칭키 암호의 분석 및 설계, 컴퓨터 포렌식



이 상 진 (Sangjin Lee) 종신회원
1987년 2월: 고려대학교 수학과 학사
1989년 2월: 고려대학교 수학과 석사
1994년 2월: 고려대학교 수학과 박사
1989년 2월~1999년 2월: 한국전자통신연구원 선임 연구원
1999년 2월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~2006년 2월: 고려대학교 정보경영공학전문대학원 부교수
2006년 3월~현재: 고려대학교 정보경영공학전문대학원 교수
〈관심분야〉 대칭키 암호의 분석 및 설계, 정보은닉이론, 컴퓨터 포렌식