情報保護學會論文誌
第 22 卷 第 3 號, 2012. 6

AES 암호 알고리듬에 대한 반복문 뒷 라운드 축소 공격*

최 두 식,[1†] 최 용 제,[2] 최 두 호,[2] 하 재 철[1‡]
[1]호서대학교, [2]한국전자통신연구원

A Late-Round Reduction Attack on the AES Encryption Algorithm Using Fault Injection*

Doo-sik Choi,[1†] Yong-Je Choi,[2] Doo-ho Choi,[2] Jae-cheol Ha[1‡]
[1]Hoseo University, [2]ETRI

요 약

오류 주입 공격은 비밀 키를 내장하여 사용하는 암호 장치에서 연산 수행시 공격자가 오류를 주입하는 방법으로 비밀 키를 찾아낼 수 있어 암호시스템 운영의 심각한 위협이 되고 있다. 논문에서는 AES 암호 연산을 수행하는 동안 라운드 함수를 반복적으로 사용하는 경우, 반복하는 구문에 오류를 넣어 한 라운드를 생략하면 쉽게 비밀 키를 추출할 수 있음을 보이고자 한다. 제안하는 공격 방법을 상용 마이크로프로세서에서 실험한 결과, 두 개의 정상-오류 암호문 쌍만 있으면 128비트 AES 비밀 키가 노출됨을 확인하였다.

ABSTRACT

Since an attacker can extract secret key of cryptographic device by occurring an error during encryption operation, the fault injection attack have become a serious threat in cryptographic system. In this paper, we show that an attacker can retrieve the 128-bits secret key in AES implementation adopted iterative statement for round operations using fault injection attack. To verify the feasibility of our attack, we implement the AES algorithm on ATmega128 microcontroller and try to inject a fault using laser beam. As a result, we can extract 128-bits secret key by obtaining just two pairs of correct and faulty ciphertexts.

Keywords: Fault injection attack, AES, Iterative statement, Cryptographic secure chip

I. 서 론

암호 연산을 수행하는 하드웨어 칩에 오류를 주입하여 비밀 키를 찾아내는 오류 주입 공격은 1997년 Boneh 등이 RSA-CRT(Chinese Remainder Theorem) 서명 알고리즘 과정에서 비밀 정보를 추출할 수 있음을 보임으로서 소개되었다[1]. 그 후 Biham과 Shamir가 대칭 암호에 대해서 오류 공격이 가능함을 제안하였는데[2], 이 경우 두 개 이상의 암호문을 차분하는 방식을 사용하였으므로 이를 차분 오류 분석(Differential Fault Analysis, DFA) 공격이라고 불렀다. 차분 오류 주입 공격은 비밀 키를 내장한 암호용 칩에서 암호 연산을 수행시, 생성된 정상 암호문과 오류를 주입하여 얻은 오류 암호문 쌍을 분석하여 비밀 키를 추출할 수 있는 공격 기법으로서 블록 암호 알고리듬에 대한 위협적인 물리적 공격 방법 중 하나이다[3-5]. 특히, 국제 표준인 AES[6] 알고리즘에 대해서도 DFA에 대한 많은 연구가 있었는데, Piret와 Quisquater는 최소 2개의 정상-오류

접수일(2011년 8월 31일), 수정일(2011년 1월 16일), 게재확정일(2012년 1월 17일)
* 본 연구는 방송통신위원회 및 한국방송통신전파진흥원의 방송통신기술개발사업의 일환인 SCARF 프로젝트로 수행하였음. [부채널 공격 방지 원천기술 및 안전성 검증기술 개발]
† 주저자, pori86@hanmail.net
‡ 교신저자, jcha@hoseo.edu

암호문 쌍을 이용하여 128비트 AES 비밀 키를 찾는 방법을 제안하였다[7]. 또한 Giraud는 연산시 한 비트 오류가 발생하거나 키 스케쥴상의 한 바이트 오류를 주입하면 비밀 키를 찾아낼 수 있음을 보였다[8]. 그 후 Kim과 Quisquater는 AES 키 스케쥴링 상에 오류를 주입하는 방법으로 모두 8개의 정상-오류 암호문 쌍을 이용하여 비밀 키를 찾아내는 방법을 제안하였다[9]. 지금까지 제시된 대부분의 오류 주입 방법들은 암호 연산시나 키 스케쥴링이 수행되는 동안 임시로 저장되는 중간 데이터에 오류를 주입하는 "데이터 오류 주입 공격"이었다.

그 후 새로운 형태의 오류 주입 공격이 제시되었는데 AES 암호 연산시 "for"문과 같은 반복적인 연산을 하는 과정에 오류를 주입하는 "반복문 오류 주입 공격"이다[10-12]. 이 방법들은 AES에서 저장된 데이터가 아닌 프로그램 코드를 공격하는 새로운 공격 형태이다. 2005년 Choukri와 Tunstall는 반복문 오류 주입 공격을 이용하여 AES의 10라운드 수행 과정을 1라운드로 줄인 후 그 결과 값을 차분하여 비밀 키를 공격하는 방법을 제안하였다[10]. 그 후 FIPS 표준에서 권고한 또 다른 AES 구현 방법을 사용해도 오류 주입을 통해 최소 2라운드까지 줄일 수 있으며, 이 구현 방법 역시 128비트 비밀 키가 노출됨을 보였다[11].

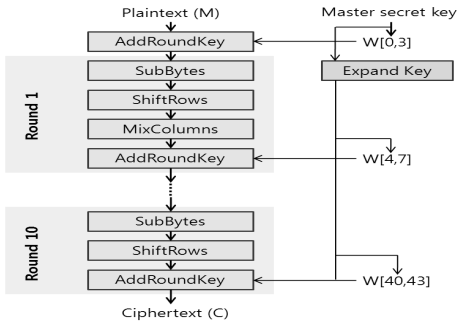
본 논문에서는 AES에 대한 좀 더 효율적인 반복문 오류 주입 공격 방법을 제안하고자 한다. 논문 [10]과 [11]에서는 암호화 과정 중 반복문의 앞단을 공격함으로써 수행하는 라운드 수를 1개~ 2개로 줄인 것에 비해, 본 논문에서는 반복문의 뒷단에서 공격함으로써 수행되는 라운드 함수 수는 9개로 많지만 한 라운드만 생략하면 공격이 더 용이함을 보일 것이다.

본 논문의 2장에서는 AES 암호 알고리듬과 구현 방법에 따른 기존 "for"문에 대한 라운드 축소 오류 주입 공격 방법을 소개한다. 제 3장에서는 논문에서 제시하는 반복문 뒷단을 공격하는 오류 주입 방법을 제시한다. 4장에서는 상용 마이크로프로세서에 AES 알고리듬을 구현할 경우 실제로 비밀 키가 노출됨을 실험을 통해 증명하고자 한다. 마지막으로 5장에서 결론을 맺는다.

II. AES 구현과 라운드 축소 공격

2.1 AES 암호 방식

AES는 미국 NIST에 의해 연방 정보처리 표준으로 지정된 대칭키 암호화 방식으로 128, 192, 256비트의 비밀 키를 사용한다. 입력 블록은 128비트이며 각 라운드의 연산 과정을 거쳐 128비트의 암호문을 출력된다. AES에서는 사용하는 키 길이에 따라 라운드 수가 달라지게 되는데, 예를 들어 128비트 비밀 키를 사용하는 경우에는 10라운드 연산을 거치게 된다. 본 논문에서는 라운드 수 N_r은 10이 되는 128비트 비밀 키를 갖는 AES를 공격 목표로 하며 이를 나타낸 것이 [그림 1]이다.



[그림 1] AES 암호화 과정

실제 암호 연산을 수행할 경우 각 라운드에는 라운드 키를 사용하게 되는데 이것은 128비트 비밀 키를 확장해서 사용한다. 키 확장 과정을 통해 모두 11개의 라운드 키를 만들게 되며 각 라운드 키는 128비트이다. 그림에서 $W[0]$, $W[1]$, $W[2]$, $W[3]$는 라운드 함수가 시작되기 전 평문과 XOR되는 초기 라운드 키이다. 암호과정에서 각 라운드는 SubBytes(SB), ShiftRows(SR), MixColumns (MC) 그리고 AddRoundKey(ARK 혹은 ⊕로 표기) 연산을 수행하는데 마지막 10라운드에는 MixColumns 연산을 수행하지 않는다.

AES 암호를 순차적으로 구현할 때 마지막 10라운드를 어떻게 처리하는가에 따라 오류 주입 공격이 달라지게 되므로 여기에 주목할 필요가 있다. 논문에서 제안하는 공격도 구현 형태에 따라 공격 방법과 복잡도가 달라지게 된다. 또한, AES에서는 한 라운드 키 128비트만 노출되면 역계산에 의해 마스터 비밀 키를 알 수 있는데 논문에서는 공격을 통해 마지막 10라운드 키 $W[40]$, $W[41]$, $W[42]$, $W[43]$를 추출하고자 한다.

2.2 AES 라운드 축소 공격

AES 암호를 구현하는 방식은 여러 가지가 있으나 동일한 라운드를 반복하게 되므로 "for"문과 같은 반복

```
state = M;
ARK(state, W[0]..W[3]);
for( i=1; i<=10 ; i++) { //오류주입 위치
  SB(state);
  SR(state);
  if( i != 10) MC(state);
  ARK(state,W[i*4]..W[i*4+3]);
}
C=state;
```

〔그림 2〕 AES 암호화 의사 코드(TYPE-Ⅰ)

문을 이용하여 구현하게 된다. 2005년, Choukri와 Tunstall는 AES가 수행되는 동안 반복적인 연산을 하는 과정에 오류를 주입하여 라운드 수를 줄여 비밀키를 공격하는 방법을 제안하였다〔10〕. 그들이 논문에서 구현하여 사용한 AES 암호 과정의 의사 코드(pseudo code)는 〔그림 2〕와 같으며 이것을 편의상 TYPE-Ⅰ이라고 부르기로 한다. 이 구현의 특징은 10라운드 과정도 "for"문 안에 두고 구현하였다. 이 공격의 핵심은 "for"문이 수행될 때 오류를 주입하여 반복문을 한 번만 수행하고 벗어나도록 하였다. 그러면 초기 ARK와 1라운드만 수행한 오류 암호문을 출력하게 된다.

공격자는 2개의 평문 M_1과 M_2을 입력한 후 출력된 오류 암호문 C_1'과 C_2'을 이용하여 다음 식을 만족하는 초기 라운드 키 $W^0 = W[0]\|W[1]\|W[2]\|W[3]$를 바이트 단위로 찾게 된다. 여기서 W^i는 128비트인 i번째 라운드 키를 의미한다.

$$SR^{-1}(MC^{-1}(C_1' \oplus C_2')) = SB(M_1 \oplus W^0) \oplus SB(M_2 \oplus W^0) \quad (1)$$

즉, (식) (1)에서 우측 항은 바이트 단위로 연산할 수 있어 초기 라운드 키를 바이트 단위로 키를 계산해 낼 수 있다. 이 공격의 핵심은 1라운드만 수행하고 "for" 문을 벗어나도록 구현 프로그램의 수행 코드에 적절한 오류를 주입하는 일인데 저자들은 이를 실험으로 가능함을 보였다.

그러나 Choukri와 Tunstall이 사용한 TYPE-Ⅰ의 구현은 오류 주입 공격이 쉽게 적용되므로 알고리즘을 구현하는 개발자가 〔그림 3〕과 같이 10라운드 과정을 "for"문 바깥에서 수행하도록 하면 공격은 복잡해진다. 실제로 AES 표준 문서인 FIPS 197에서는 10라운드를 "for"문 바깥에서 수행하도록 예시하고 있다. 자세한 내용은 표준문서 〔6〕에서 참조할 수 있으며 이와 같은 구현을 편의상 TYPE-Ⅱ이라고 부르기로 한다. 이 경우 "for"문에서 오류를 주입하여 한 라운드만 수행하고 빠져 나오게 되므로 초기 ARK와 1라운드 그리고 10라운드를 수행한 오류 암호문을 출력하게 된다. 이와 같은 공격에는 10개의 정상-오류 암호문 쌍과 2^{40}번의 라운드 키 전수조사(exhaustive search)가 필요하다〔11〕.

```
state = M;
ARK(state, W[0]..W[3]);
for(i=1; i<=(Nr-1); i++) //오류주입 위치
{
  SB(state);
  SR(state);
  MC(state);
  ARK(state,W[i*4]..W[i*4+3]);
}
  SB(state);
  SR(state);
  ARK(state,W[Nr*4]..W[Nr*4+3]);
C=state;
```

〔그림 3〕 AES 암호화 의사 코드(TYPE-Ⅱ)

Ⅲ. 제안하는 라운드 축소 오류 주입 공격

본 논문에서는 TYPE-Ⅰ과 TYPE-Ⅱ와 같은 구현에서 새로운 라운드 축소 오류 주입 공격을 제안하고자 한다. 공격의 핵심은 오류를 주입하는 위치를 "for"문이 빠져나오는 반복문의 앞부분이 아니라 뒷부분으로 하여 공격에 필요한 오류 주입 회수 및 연산 복잡도를 낮추고자 하는 것이다. 즉, 지금까지의 AES 공격은 반복문이 수행되자마자 오류를 주입함으로써 한 번의 반복문이 수행되게 하였다. 그러나 논문에서는 반복문이 수행되는 과정에서 마지막 라운드 부분을 수행하기 전에 오류를 넣어 "for"문 내의 마지막 라운드를 생략하자는 것이다.

3.1 TYPE-Ⅰ 구현에 관한 공격 제안 1

먼저 TYPE-Ⅰ과 같이 구현한 경우를 살펴보자. 먼저 공격을 위해서는 하나의 메시지 M에 대한 정상 암호문 C이 필요하다. 그리고 동일한 메시지를 입력으로한 오류 암호문 C'을 얻어야 한다. 오류 암호문은 마지막 10라운드만 수행하지 않게 되므로 9라운드

까지 수행한 결과가 된다. 즉, $C' = M^9$이 된다. 따라서 다음과 같은 등식이 성립하게 된다. 여기서 M^i는 i번째 라운드까지 수행한 중간 값이 된다.

$$\begin{aligned} C &= (SR(SB(M^9))) \oplus W^{10} \\ &= (SR(SB(C'))) \oplus W^{10} \end{aligned} \quad (2)$$

식 (2)에서 보면 10라운드 키 W^{10}을 제외한 모든 값을 알 수 있으므로 아래 수식을 통해 라운드 키를 구할 수 있다. 식 (3)은 모두 바이트 단위로 처리가 가능하므로 쉽게 128비트 10라운드 키를 구할 수 있다. 따라서 이 공격에는 하나의 정상-오류 암호문 쌍이 필요할 뿐이다.

$$W^{10} = (SR(SB(C'))) \oplus C \quad (3)$$

3.2 TYPE-II 구현에 관한 공격 제안 2

AES를 TYPE-II과 같이 구현했을 경우 공격하는 방법은 좀 더 복잡해진다. 이 공격을 위해서는 먼저 하나의 평문 M_1에 대한 정상 암호문 C_1이 있다고 가정하자. 그리고 [그림 3]의 알고리듬이 수행되는 동안 오류를 주입하여 "for"문내의 마지막 라운드를 수행하지 않고 생략한다. 그러면 이 오류 암호문은 9번째 라운드만 제외하고 1라운드에서 8라운드까지 그리고 마지막 10라운드를 수행한 결과와 같다. 따라서 8라운드까지 수행한 결과인 M_1^8이후의 연산은 다음 수식과 같다.

$$C_1 = SR(SB(MC(SR(SB(M_1^8))) \oplus W^9)) \oplus W^{10} \quad (4)$$

$$C_1' = SR(SB(M_1^8)) \oplus W^{10} \quad (5)$$

여기서 식 (5)를 다시 전개하면 다음과 같다.

$$C_1' \oplus W^{10} = SR(SB(M_1^8)) \quad (6)$$

따라서 식 (4)는 중간 메시지 M_1^8을 제거하고 식 (7)과 같이 표현할 수 있다.

$$C_1 = SR(SB(MC(C_1' \oplus W^{10}) \oplus W^9)) \oplus W^{10} \quad (7)$$

식 (7)에서는 구해야 하는 라운드 키가 W^9와 W^{10}이 있어 실질적인 탐색이 어렵다. 따라서 새로운 메시지 M_2에 대한 정상 암호문 C_2와 오류 암호문 C_2'을 구한다. 이 경우 정상 암호문과 오류 암호문 쌍의 관계식은 식 (8)과 같다.

$$C_2 = SR(SB(MC(C_2' \oplus W^{10}) \oplus W^9)) \oplus W^{10} \quad (8)$$

식 (7)과 (8)에서 각각 W^9를 다시 표현하면 다음과 같다.

$$W^9 = (SB^{-1}(SR^{-1}(C_1 \oplus W^{10}))) \oplus (MC(C_1' \oplus W^{10})) \quad (9)$$

$$W^9 = (SB^{-1}(SR^{-1}(C_2 \oplus W^{10}))) \oplus (MC(C_2' \oplus W^{10})) \quad (10)$$

위의 두 식 (9)와 (10)을 차분하면 W^9는 소거되면서 식 (11)과 같은 등식이 성립한다.

$$\begin{aligned} &(SB^{-1}(SR^{-1}(C_1 \oplus W^{10}))) \oplus (MC(C_1')) \\ &\oplus (SB^{-1}(SR^{-1}(C_2 \oplus W^{10}))) \oplus (MC(C_2')) = 0 \end{aligned} \quad (11)$$

식 (9)와 (10)에서 우변항의 오류 암호문과 XOR하는 W^{10}은 MC 연산이 선형성을 만족하므로 두 식을 차분하는 과정에 소거된다. 결국, 공격자는 두 쌍의 정상-오류 암호문 쌍을 가지고 식 (11)을 만족하는 10라운드 키를 찾게 된다.

공격 과정에서 SR^{-1}연산을 고려하여 행(row)을 기준으로 라운드 키를 한 바이트씩 찾게 된다. 이미 공격자는 오류 암호문으로부터 $MC(C_1')$과 $MC(C_2')$을 계산할 수 있고 $R = MC(C_1') \oplus MC(C_2')$이라 두자. SB^{-1}와 SR^{-1}는 바이트 단위로 처리가 가능하다. 예를 들어 10라운드 키는 $W^{10} = W[40] \| W[41] \| W[42] \| W[43]$와 같이 4개의 워드로 구성되어 있고, 첫 번째 워드는 $W[40] = b_{0,0} \| b_{0,1} \| b_{0,2} \| b_{0,3}$와 같이, 마지막 워드는 $W[43] = b_{3,0} \| b_{3,1} \| b_{3,2} \| b_{3,3}$와 같이 4개의 바이트씩 구성되어 있다. 따라서 바이트 단위로 라운드 키를 예측하여 다음 등식이 성립하는 키를 찾을 수 있게 된다.

$$\begin{aligned} &(SB^{-1}(SR^{-1}(C_{1(i,j)} \oplus b_{i,j}))) \\ &\oplus (SB^{-1}(SR^{-1}(C_{2(i,j)} \oplus b_{i,j}))) = R_{i,j} \end{aligned} \quad (12)$$

여기서 $0 \le i, j \le 3$이며 $C_{1(i,j)}$는 암호문 C_1의 i번째 행, j번째 열의 바이트를 의미한다. 또한, $R_{i,j}$는 $R = MC(C_1') \oplus MC(C_2')$의 i번째 행, j번째 열의 바이트를 나타낸다.

따라서 공격자는 두 쌍의 정상-오류 암호문 쌍을 이용하여 식 (12)를 만족하는 10라운드 키를 바이트 단위로 찾게 된다. 이 경우 식 (12)를 만족하는 라운드 키는 각 바이트마다 두 개씩 존재하게 된다. 즉, 두 쌍의 정상-오류 암호문 쌍을 이용하면 식 (12)를 만족하는 각 바이트의 키 후보가 256개에서 2개로 줄어들게 된다. 결국, 10라운드 키 전체를 찾기 위해서는 16개의 키 바이트를 찾아야 하므로 두 개의 정상-오

〔표 1〕 AES 라운드 축소 오류 주입 공격 비교

구 분	Choukri-Tunstall〔10〕	Park 등의 방법〔11〕	제안 방법 1	제안 방법 2
AES 구현 방법	TYPE-Ⅰ	TYPE-Ⅱ	TYPE-Ⅰ	TYPE-Ⅱ
오류 주입 위치	반복 함수 앞단	반복 함수 앞단	반복 함수 뒷단	반복 함수 뒷단
오류 주입시 수행 라운드	1라운드	1, 10라운드	1~9 라운드	1~8, 10 라운드
공격 라운드 키	초기 라운드 키 W^0	초기 라운드 W^0	10 라운드 W^{10}	10 라운드 W^{10}
필요한 입력	2	10	1	2
정상 암호문	0	10	1	2
오류 암호문	2	10	1	2
복잡도(전수조사)	-	2^{40}	-	2^{16}

류 암호문 쌍을 이용하여 2^{16}번의 바이트 단위 전수조사가 필요하다. 그러나 이러한 10라운드 키 후보군에 대한 전수 조사는 수 초(sec)안에 이루어지므로 키를 찾는데 문제가 되지는 않는다.

Ⅳ. 비교 및 실험 고찰

4.1 공격 방법 비교

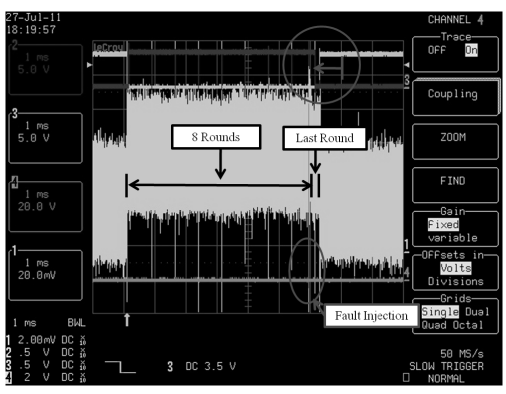
지금까지의 라운드 축소를 이용한 공격 방법을 비교한 것이 〔표 1〕이다. Choukri-Tunstall 방법과 Park 등의 방법은 반복 함수의 앞단을 공격하므로 초기 라운드 키를 추출하였지만 제안 방법에서는 10라운드 키를 추출하고자 하였다. Choukri-Tunstall 방법의 특징은 정상 암호문이 필요 없고 오류 암호문 두 쌍만 필요한 반면, 제안 방법 1에서는 동일한 입력에 대한 정상 암호문과 오류 암호문 쌍이 필요하므로 실제로는 한 번의 오류 주입만으로 공격이 가능하다. 제안 방법 2도 Park 등의 방법에 비해 2개의 정상-오류 암호문 쌍만으로도 비밀 키를 추출할 수 있어 매우 위협적인 공격임을 알 수 있다. 따라서 반복 함수의 앞단을 생략하는 공격 방법보다 뒷단의 라운드 함수를 생략하는 공격 방법이 보다 용이함을 알 수 있다.

4.2 오류 주입 실험

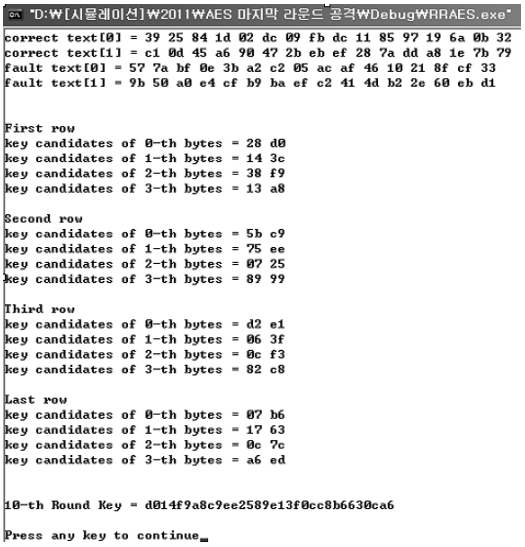
논문에서는 제안하는 라운드 축소 오류 주입 공격이 실제로 가능한지를 실험하였다. 암호를 위한 칩으로 사용할 수 있는 상용 마이크로프로세서 ATmega128〔13〕에 AES를 구현하여 실험하였다. 오류 주입을 위해서는 EzLaze 3 레이저 장비를 사용하였다. 물론, 라운드가 시작하고 마치는 등의 시간 측정을 위해서는 오실로스코프 장비를 사용하였으며 오류 주입을 위해 칩은 디캡핑(decapping)하여 실험하였다.

실험에 사용된 128비트의 AES 알고리즘의 비밀 키는 "0x 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c"이며, 10라운드 키는 "0x d0 14 f9 a8 c9 ee 25 89 e1 3f 0c c8 b6 63 0c a6"이다. 그리고 평문은 "0x 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34"을 사용하였다.

〔그림 4〕는 TYPE-Ⅱ 형태로 구현한 AES 암호 알고리듬에 오류를 주입하여 공격한 파형을 나타낸 것이다. 그림에서 보는 바와 같이 8라운드까지 수행한 후 오류에 의해 9라운드를 생략하고 바로 10라운드만 수행함을 파형을 통해서 확인하였고 또한 출력 값을 통해서도 검증하였다. 그림의 하단에는 오류가 주입된



〔그림 4〕 TYPE-Ⅱ 구현에 대한 라운드 축소 오류 주입



〔그림 5〕 라운드 키에 대한 전수 조사

이후의 출력을 보인 것인데 정상 암호문은 "0x 39 25 84 1d 02 dc 09 fb dc 11 85 97 17 6a 0b 32"이고 오류암호문은 "0x 57 7a bf 0e 3b a2 c2 05 ac af 46 10 21 8f cf 33"이다.

〔그림 5〕는 위의 오류 주입 실험에서 얻은 2개의 정상-오류 암호문 쌍을 이용하여 10라운드 키를 찾는 과정을 나타낸 것이다. 각 바이트마다 키를 추측하는데 2개씩의 후보 키가 추출되었으며 이에 대한 2^{16}번의 전수 조사를 통해 최종적으로 10라운드 키를 찾을 수 있었다.

V. 결 론

그 동안 AES에 대한 오류 주입 공격은 암호 연산을 수행하거나 키 스케쥴 과정에서 중간 값에 오류를 주입하여 비밀 키를 찾는 방법이 대부분이었다. 그러나 최근 중간 데이터에 대한 변형이 아닌 프로그램 코드에 오류를 주입하여 잘못된 연산을 유도하는 오류 주입 공격이 시도되었고 실험 과정을 통해 증명된 바 있다.

본 논문에서는 AES에서 사용하는 반복적인 라운드 함수를 수행하는 동안 프로그램 코드 오류를 통해 특정 라운드 함수를 생략하는 공격을 제안하였다. 기존의 공격들이 반복 함수의 앞단에서 공격한 반면 제안 논문에서는 반복문의 뒷단에서 공격하여 한 라운드를 생략하는 방법을 제안하였다. AES 알고리듬의 구현 방법에 따라 공격 형태는 다르지만 이전의 분석 방법보다 보다 용이하게 비밀 키를 추출할 수 있었다. 따라서 암호용 칩 개발자들은 이러한 오류 주입 공격에 대비하여 물리적 보완 대책과 더불어 소프트웨어 구현 기법에 대한 대응책을 강구하여야 할 것이다.

참고문헌

〔1〕 D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," EUROCRYPTO'97, LNCS 1233, pp. 37-51, 1997.

〔2〕 E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," CRYPTO'97, LNCS 1294, pp. 513-525, 1997.

〔3〕 정기태, 성재철, 홍석희, "블록 암호 SEED에 대한 차분 오류 공격," 한국정보보호학회 논문지, 20(4), pp. 17-24, 2010년 8월.

〔4〕 소현동, 김성경, 홍석희, 강은숙, "DES 알고리즘에 대한 새로운 차분오류주입공격 방법," 한국정보보호학회논문지, 20(6), pp. 3-13, 2010년 12월.

〔5〕 최두식, 오두환, 배기석 문상재, 하재철, "오류 주입을 이용한 Triple DES에 대한 라운드 축소 공격," 한국정보보호학회논문지, 21(2), pp. 91-100, 2011년 4월.

〔6〕 National Institute of Standards and Technology, "Advanced Encryption Standards," NIST FIPS PUB 197, 2001.

〔7〕 G. Piret and J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," CHES'03, LNCS 2779, pp. 77 - 88, 2003.

〔8〕 C. Giraud, "DFA on AES," Advanced Encryption Standard-AES'04, LNCS 3373, pp. 27 - 41, 2005.

〔9〕 C. Kim and J. Quisquater, "New Differential Fault Analysis on AES Key Schedule: Two Faults are enough," CARDIS'08, LNCS 5189, pp. 48-60, 2008.

〔10〕 H. Choukri and M. Tunstall, "Round reduction using faults," FDTC'05, pp. 13-24. 2005.

〔11〕 J. H. Park, S. J. Moon, D. H. Choi, Y. S. Kang, and J. C. Ha, "Differential fault analysis for round-reduced AES by fault injection," ETRI Journal, vol. 33 no. 3, pp. 434-442, 2011.

〔12〕 박제훈, 배기석, 오두환, 문상재, 하재철, "AES에 대한 반복문 오류주입 공격," 한국정보보호학회논문지, 20(6), pp. 59-65, 2010년 12월.

〔13〕 Atmel사 홈페이지, http://www.atmel.com/atmel/acrobat/doc2467.pdf

〈著者紹介〉

최 두 식 (Doo-Sik Choi) 학생회원
2010년 2월: 호서대학교 정보보호학과 졸업
2012년 2월: 호서대학교 정보보호학과 석사
2012년 6월 ~현재: ㈜ 소프트포럼
〈관심분야〉 네트워크 보안, 부채널 공격

최 용 제 (Yong-Je Choi) 정회원
1996년 8월: 전남대학교 전자공학과 졸업
1999년 2월: 전남대학교 전자공학과 석사
1999년 2월 ~ 1999년 8월: 전남대학교 전자통신연구소 인턴연구원
1999년 8월 ~ 현재: 한국전자통신연구원 선임연구원
〈관심분야〉 보안프로세서 설계, 부채널 분석 시스템, RFID/USN 보안

최 두 호 (Doo-Ho Choi) 정회원
1994년 2월: 성균관대학교 수학과 졸업
1996년 2월: KAIST 수학과 석사
2002년 2월: KAIST 수학과 박사
2002년 1월 ~ 현재: 한국전자통신연구원 선임연구원
〈관심분야〉 암호학, 부채널 분석, RFID/USN 보안

하 재 철 (Jae-Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 2월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 부교수
2007년 3월~현재: 호서대학교 정보보호학과 부교수
〈관심분야〉 정보보호, 네트워크 보안, 부채널 공격