다양한 환경에 적용 가능한 AES-CMAC에 대한 안전성 분석

Security Analysis of AES-CMAC Applicable to Various Environments

정기태*

Ki-Tae Jeong*

요 약

본 논문에서는 IETF 표준 MAC 알고리즘 AES-CMAC에 대한 오류 주입 공격을 제안한다. 본 공격에서 사용된 오류 주입 가정은 FDTC'05에서 제안된 공격 모델에 기반을 둔다. 본 논문에서 제안하는 공격은 매우 적은 수의 오류 주입만을 이용하여 AES-CMAC의 128-비트 비밀키를 복구할 수 있다. 본 공격 결과는 AES-CMAC에 대한 첫 번째 키 복구 공격 결과이다.

Abstract

In this paper, we propose a fault injection attack on AES-CMAC, which is defined by IETF. The fault assumption used in this attack is based on that introduced at FDTC'05. This attack can recover the 128-bit secret key of AES-CMAC by using only small number of fault injections. This result is the first known key recovery attack result on AES-CMAC.

Key words : Block Cipher, AES-CMAC, Fault Injection Attack

I. 서 론

MAC(Message Authentication Codes)은 키를 사용하여 임의의 길이의 메시지를 고정된 길이로 압축하는 함수이다. MAC을 사용하는 목적은 메시지에 대한 무결성을 보장하고, 메시지 출처 인증을 하기 위해서이다. 여기서, 메시지에 대한 무결성은 메시지의 변조 여부를 알 수 있다는 것을 의미하고, 메시지 출처 인증은 메시지를 보낸 사람에 대한 확인이 가능하다는 것을 말한다.

CMAC(Cipher-based MAC)은 블록 암호 기반 MAC 알고리즘으로서, 2005년 NIST에서 권고 MAC 알고리즘(NIST Special Publication 800-38B)으로 발표되었다 [1]. 대표적인 MAC 알고리즘 중의 하나인 CBC-MAC은 가변 길이의 메시지에 대해서는 안전하지 않다고 알려져 있다. 이를 개선한 버전인 XCBC (eXtended CBC) [2]는 3개의 비밀키(1개의 비밀키 및 2개의 서브키)를 사용하여 가변 길이의 메시지에 대해 안전하도록 설계되었다. CMAC는 XCBC에서 1개의 비밀키로서 2개의 서브키를 생성하여 키 크기를 줄인 것이다. 구체적인 예로서, 블록 암호 AES-128 [3]를 사용하는 CMAC인 AES-CMAC [4]은 2006년 IETF 표준 MAC 알고리즘(RFC 4493)으로 제정되었다. RFC 4493에서 권장하는 최소 MAC 값

* 고려대학교 정보보호연구원(Center for Information Security Technologies, Korea University)
· 제1저자 (First Author) : 정기태
· 투고일자 : 2012년 2월 13일
· 심사(수정)일자 : 2012년 3월 14일 (수정일자 : 2012년 4월 20일)
· 게재일자 : 2012년 4월 30일

의 길이는 64 비트이다. 이 중 96-비트 MAC 값을 생성하는 AES-CMAC-96 [5]은 동일한 해에 IETF 표준 MAC 알고리즘(RFC 4494)으로 제정되었다.

오류 주입 공격 [6]은 대표적인 부채널 공격 기법 중 하나로서, 공격 대상 알고리즘에 전력 변화, 강제 클럭킹 등을 이용하여 오류를 발생시키고 이를 이용하여 비밀키 정보를 얻는 공격 기법이다. 기제안된 대부분의 블록 암호에 대한 오류 주입 공격은 특정 라운드의 내부 레지스터에 오류를 주입함으로서 발생되는 암호문의 차분을 이용하여 비밀키를 복구한다 [7,8]. 이와 달리, FDTC'05에서는 오류 주입을 통하여 타깃 알고리즘의 라운드 수를 감소시킴으로서 AES의 비밀키를 찾을 수 있을 보였다 [9]. 이 공격의 오류 주입 가정은 “for”문과 같은 반복문에 오류를 주입하여 타깃 알고리즘의 라운드 수를 1로 감소시킨다는 것이다. [9]의 공격 아이디어를 이용하여 CISC-W'10에서는 블록 암호 Triple-DES에 대한 오류 주입 공격 결과가 소개되었다 [10].

본 논문에서는 [9]의 공격 아이디어를 AES-CMAC에 적용하여 AES-CMAC의 128-비트 비밀키를 복구할 수 있음을 보인다. 본 논문에서 제안하는 공격의 오류 주입 가정은 오류 주입을 통하여 AES-128의 라운드 수를 감소시킨다는 것이다. 앞에서 언급하였듯이, AES-CMAC으로부터 생성되는 MAC 값의 최소 길이는 64 비트이다. 본 논문에서는 구체적인 예로서 64/96/128-비트 MAC 값을 생성하는 AES-CMAC-64/96/128에 대한 오류 주입 공격을 제안한다. 표 1은 AES-CMAC에 대한 오류 주입 공격 결과를 나타낸 것이다. 표를 통해 알 수 있듯이, 본 논문에서 제안하는 공격은 매우 적은 수의 오류 주입만을 이용하여 AES-CMAC의 128-비트 비밀키를 복구할 수 있다. 본 공격 결과는 AES-CMAC에 대한 첫 번째 키 복구 공격 결과이다.

표 1. AES-CMAC에 대한 오류 주입 공격 결과
Table 1. Our attack results on AES-CMAC.

알고리즘	메시지 수	오류 주입 수	공격 복잡도
AES-CMAC-64	1	3	2^{56} AES-CMAC-64 연산
AES-CMAC-96	1	2	2^{24} AES-CMAC-96 연산
AES-CMAC-128	1	1	2 AES-CMAC-128 연산

본 논문은 다음과 같이 구성되어 있다. 먼저 2장에서는 AES-128과 AES-CMAC에 대해 간략히 소개한다. AES-CMAC-64/96/128에 대한 오류 주입 공격은 각각 3장, 4장, 5장에서 제안된다. 마지막으로 6장에서 결론을 맺는다.

Ⅱ. AES-CMAC

2-1 AES-128

AES-128은 128-비트 블록 암호로서, 128-비트 비밀키를 사용하며 10 라운드로 구성되었다. AES-128의 암호화 과정상의 128-비트 내부 상태값은 그림 1과 같이 16개 바이트로 이루어진 4×4 행렬로 나타낼 수 있다. 본 논문에서는 특정 내부 상태값 S의 i 번째 바이트 값을 $S[i]$로 표기하기로 한다 $(i=0,\cdots,15)$.

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

그림 1. AES-128의 16-바이트 내부 상태값
Fig. 1. 16-byte inner state of AES-128.

AES-128은 평문 P와 키스케줄에 의해 생성된 라운드 RK_r을 입력 받아 10번 라운드 함수를 반복적으로 수행한 후 암호문 C를 출력한다 $(r=0,\cdots,10)$. AES-128의 한 라운드는 SubBytes(SB), ShiftRows(SR), MixColumn(MC), AddRoundKey(ARK) 함수를 차례대로 사용하며, 첫 번째 라운드 전에 ARK 함수를 적용하고, 마지막 라운드에서 MC 함수를 생략한다. 각 함수는 다음과 같이 동작한다.

- SubBytes(SB) 함수는 동일한 S-box를 각각의 내부 상태값 바이트에 적용시킨 비선형 대치 연산이다.

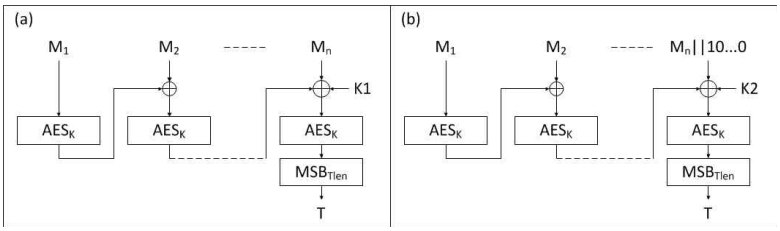

그림 3. AES-CMAC
Fig. 3. AES-CMAC.

- ShiftRows(SR) 함수는 내부 상태값의 각각의 행에 대한 바이트별 순환 이동 변환이다. 첫 번째 행은 순환 이동이 없으며, 두 번째, 세 번째, 네 번째 행들을 각각 좌측으로 $1,2,3$ 바이트만큼 순환 이동한다.
- MixColumns(MC) 함수는 선형 변환으로 4 바이트로 구성된 각 열을 변환시키는 4×4 행렬로 $GF(2^8)$ 상에서 연산된다.
- AddRoundKey(ARK) 함수는 키스케줄에 의하여 비밀키로부터 생성된 라운드 키와 내부 상태값의 비트별 XOR 연산으로 이루어진다.

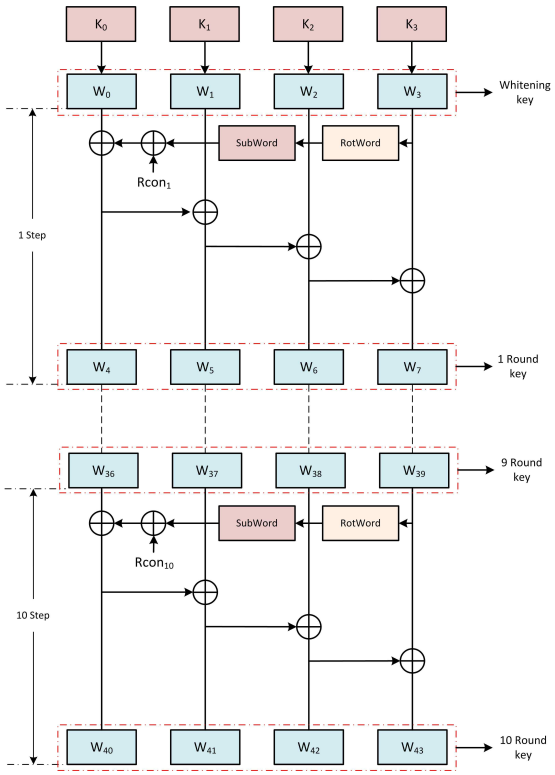

그림 2. AES-128의 키스케줄
Fig. 2. Keyschedule of AES-128.

AES-128의 키스케줄은 그림 2와 같이 4개의 32-비트 비밀키 워드($K=K_0K_1K_2K_3$)를 입력 받아 44개의 32-비트 라운드 키 워드를 생성하는 과정이다. 처음 생성되는 4개의 32-비트 워드 W_0, W_1, W_2, W_3는 비밀키를 그대로 사용한다. 그림 2에서 $Rcon_i$는 라운드 상수이며, RotWord 함수와 SubWord 함수는 다음과 같이 동작한다.

- RotWord 함수는 4-바이트 워드 입력값을 좌측으로 1-바이트 순환 이동시키는 함수이다.
- SubWord 함수는 입력 받은 4-바이트 워드 입력값의 각 바이트에 S-box를 적용하는 함수이다.

2-2 AES-CMAC

AES-CMAC은 AES-128을 사용하는 CMAC 알고리즘이며, 128-비트 비밀키를 사용한다. AES-CMAC의 구조는 그림 3과 같다. 그림 3-(a)는 메시지 $M(=M_1M_2\cdots M_n)$의 길이가 128의 배수일 때 AES-CMAC이 동작하는 과정을 나타낸 것이고, 그림 3-(b)는 메시지의 길이가 128의 배수가 아닐 경우를 나타낸 것이다. 그림에서, 128-비트 서브키 $K1, K2$는 128-비트 비밀키 K로부터 생성된다. MAC 값 T를 생성하기 전에 수행되는 MSB_{Tlen} 함수는 AES-128의 128-비트 출력값의 최상위 $Tlen$-비트 값을 출력하는 함수이다. RFC 4493에서 권장하는 $Tlen$의 최솟값은 64이다.

Ⅲ. AES-CMAC-128에 대한 오류 주입 공격

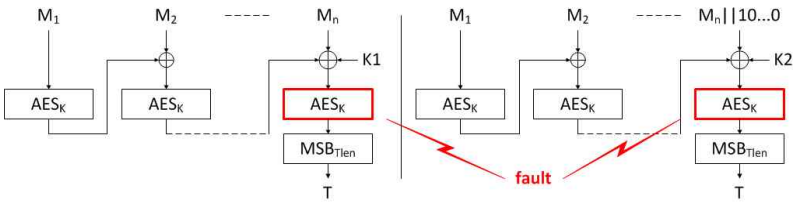

그림 4. AES-CMAC에 대한 오류 주입 공격 모델
Fig. 4. Our fault model on AES-CMAC.

본 장에서는 128-비트 MAC 값을 생성하는 AES-CMAC-128에 대한 오류 주입 공격을 제안한다. 본 논문에서 제안하는 오류 주입 공격은 AES-CMAC의 두 가지 버전(그림 3 참조)에 모두 적용 가능하며, 두 가지 버전에 대한 공격 과정이 동일하다. 또한, 한 가지 버전(예를 들어, 그림 3-(a))에 대한 공격을 수행하여 비밀키 K를 복구하였다면 쉽게 서브키 $(K1, K2)$를 계산할 수 있다. 따라서 본 논문에서는 두 가지 버전에 대한 공격을 구분 없이 소개한다.

AES-CMAC-128에 대한 오류 주입 공격은 [9]의 오류 주입 가정을 AES-128에 적용하여 AES-CMAC-128의 128-비트 비밀키를 복구한다. 즉, 그림 4와 같이 마지막 AES-128에 오류를 주입하여, 이 블록 암호가 9 라운드만 수행하도록 한다. 그러면 AES-128의 라운드 10의 입·출력값 (I_{10}, O_{10})을 알 수 있다. 이 값을 이용하여 128-비트 비밀키 K를 복구한다.

3-1. 공격 과정

본 장에서 소개하는 공격은 1개의 오류 주입만을 이용하여 128-비트 비밀키 K를 복구할 수 있다. 본 공격의 공격 과정은 다음과 같다.

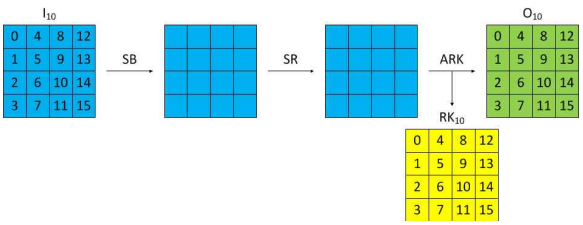

그림 5. AES-CMAC-128에 대한 오류 주입 공격의 단계 2
Fig. 5. Step 2 of our attack on AES-CMAC-128.

1. 오류를 주입하지 않은 AES-CMAC-128의 128-비트 MAC 값 $T(= O_{10})$를 얻는다. 그리고 오류를 주입한 AES-CMAC-128을 이용하여, AES-128의 라운드 9의 출력값 $O_9(= I_{10})$를 계산한다.

2. (I_{10}, O_{10})을 이용하여, 라운드 10의 라운드 키 RK_{10}을 계산한다 (그림 5 참조).

3. RK_{10}을 이용하여, AES-128의 키스케줄을 통해 128-비트 비밀키 K를 복구한다.

4. 복구한 K를 이용하여, 서브키 $(K1, K2)$를 복구한다.

3-2 공격 복잡도

본 공격은 1개의 메시지와 1개의 오류 주입만으로 적용 가능하다. 계산 복잡도의 경우, 단계 1에서의 계산 복잡도가 본 공격의 주된 계산 복잡도이다. 즉, 단계 1에서는 2개의 128-비트 MAC 값을 계산하기 위해 약 2번의 AES-CMAC-128 연산이 필요하다. 따라서 본 절에서 제안한 AES-CMAC-128에 대한 오류 주입 공격은 1개의 메시지와 1개의 오류 주입을 필요로 하며, 계산 복잡도는 약 2 AES-CMAC-128 연산이다.

Ⅳ. AES-CMAC-96에 대한 오류 주입 공격

앞에서 소개한 AES-CMAC-128에 대한 오류 주입 공격은 오류 주입을 통해 라운드 10의 입·출력값을 알 수 있기 때문에, 쉽게 비밀키를 복구할 수 있었다. 하지만 AES-CMAC-96의 경우, 앞 장의 오류 주입 가정을 적용하면 라운드 10의 96-비트 입·출력값

$(I_{10}[0,1,\cdots,11], O_{10}[0,1,\cdots,11])$만을 알 수 있다. 따라서 마지막 라운드 키를 모두 복구할 수 없다. 그래서 본 공격에서는 오류를 2번 주입한다. 즉, 그림 4와 같이 마지막 AES에 오류를 주입하여, 이 블록암호가 각각 8,9 라운드만 수행하도록 한다. 그러면 AES-128의 라운드 9,10의 96-비트 입·출력값을 알 수 있다. 이 값을 이용하여 128-비트 비밀키 K를 복구한다.

4-1 공격 과정

1. 오류를 주입하지 않은 AES-CMAC-96의 96-비트 MAC 값 $T(=O_{10}[0,1,\cdots,11])$를 얻는다. 그리고 오류를 주입한 AES-CMAC-96을 이용하여, AES-128의 라운드 8,9의 96-비트 출력값 $O_8[0,\cdots,11](=I_9[0,\cdots,11])$, $O_9[0,\cdots,11]$ $(=I_{10}[0,\cdots,11])$을 각각 계산한다.

2. $(I_{10}[0,\cdots,11], O_{10}[0,\cdots,11])$을 이용하여, 라운드 10의 72-비트 라운드 키 $RK_{10}[0,1,2,4,5,7,8,10,11]$을 계산한다 (그림 6 참조).

3. 라운드 9의 24-비트 입력값 $I_9[13,14,15]$를 추측한 후, $(I_9[0,\cdots,11], O_9[0,\cdots,11])$을 이용하여 라운드 9의 96-비트 라운드 키 $RK_9[0,\cdots,11]$을 계산한다 (그림 6 참조).

4. 추측한 각각의 24-비트 값 $I_9[13,14,15]$에 대해, AES-128의 키스케줄을 이용하여 라운드 9의 나머지 32-비트 라운드 키 $RK_9[12,13,14,15]$를 다음과 같이 계산한다 (그림 7 참조).

$$
\begin{aligned}
RK_9[12] &= \mathrm{SubWord}^{-1}(RK_9[3] \oplus RK_9[7] \\
&\quad \oplus RK_{10}[7] \oplus Rcon_{10}[3]), \\
RK_9[13] &= \mathrm{SubWord}^{-1}(RK_9[0] \oplus RK_{10}[0] \\
&\quad \oplus Rcon_{10}[0]), \\
RK_9[14] &= \mathrm{SubWord}^{-1}t(RK_9[1] \oplus RK_{10}[1] \\
&\quad \oplus Rcon_{10}[1]t), \\
RK_9[15] &= \mathrm{SubWord}^{-1}t(RK_9[2] \oplus RK_{10}[2] \\
&\quad \oplus Rcon_{10}[2]).
\end{aligned}
$$

총 2^{24}개의 후보 라운드 키 RK_9로부터 후보 128-비트 비밀키 K를 각각 복구한 후, 후보 서브키 $(K1, K2)$를 계산한다.

5. 각각의 후보 키 $(K, K1, K2)$와 주어진 메시지를 이용하여 AES-CMAC-96의 96-비트 MAC 값 T^*를 계산한다. T와 동일한 T^*를 생성하는 후보 비밀키 K를 AES-CMAC-96의 옳은 128-비트 비밀키로 출력한다.

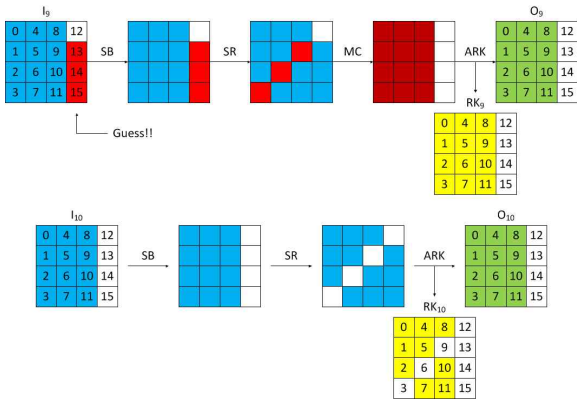

그림 6. AES-CMAC-96에 대한 오류 주입 공격의 단계 2와 단계 3
Fig. 6. Step 2 and 3 of our attack on AES-CMAC-96.

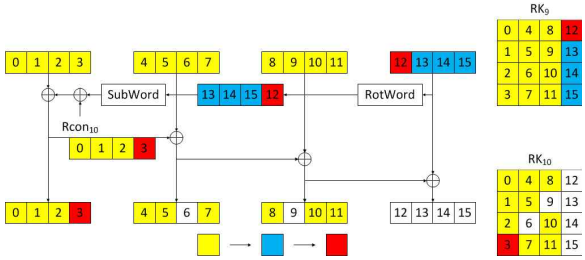

그림 7. AES-CMAC-96에 대한 오류 주입 공격의 단계 4
Fig. 7. Step 4 of our attack on AES-CMAC-96

4-2 공격 복잡도

본 공격에 필요한 데이터 복잡도는 다음과 같다. 먼저, 단계 1에서 오류가 주입되지 않은 상태 혹은 오류가 주입된 상태에서의 MAC 값을 생성하기 위해 1개의 메시지가 필요하다. 단계 5에서는 후보 비밀키로부터 옳은 비밀키를 찾기 위해 각각의 후보 비밀키를 이용하여 96-비트 MAC 값을 생성하는데, 이 과정에서 또 하나의 메시지가 필요하다. 하지만, 단계 1에서 사용된 메시지를 한 번 더 이용하면 되기 때문에 본 공격에서는 1개의 메시지만을 필요로 한

다. 그리고 라운드 9,10의 96-비트 입·출력값을 얻기 위해 2번의 오류 주입이 필요하다.

본 공격의 계산 복잡도는 단계 5에서의 계산 복잡도에 가장 큰 영향을 받는다. 단계 5에서는 총 2^{24}개의 후보 비밀키를 각각 사용하여 AES-CMAC-96의 96-비트 MAC 값을 생성한 후, 옳은 비밀키로부터 생성된 MAC 값과 비교하여 옳은 비밀키를 찾으므로, 이 단계의 계산 복잡도는 2^{24} AES-CMAC-96 연산이다. 따라서 본 절에서 제안한 AES-CMAC-96에 대한 오류 주입 공격의 계산 복잡도는 약 2^{24} AES-CMAC-96 연산이다. 한편, 틀린 후보 비밀키가 단계 5를 통과할 확률은 2^{-96}이다. 이는 매우 높은 확률로 옳은 비밀키만이 단계 5를 통과함을 의미한다.

Ⅴ. AES-CMAC-64에 대한 오류 주입 공격

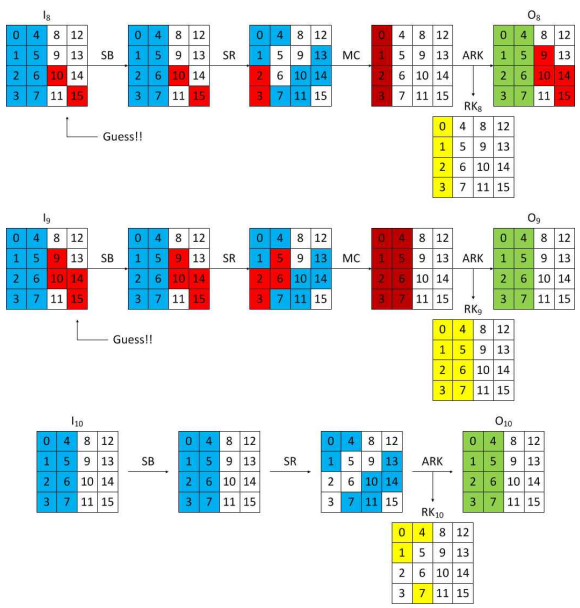

그림 8. AES-CMAC-64에 대한 오류 주입 공격의 단계 2, 3, 4
Fig. 8. Step 2, 3 and 4 of our attack on AES-CMAC-64.

AES-CMAC-96에 대한 오류 주입 공격에서는, AES-CMAC-128에 대한 오류 주입 공격과 달리, 라운드 9,10의 96-비트 입·출력값만을 알 수 있기 때문에 오류를 2번 주입하여 128-비트 비밀키를 복구할 수 있었다. AES-CMAC-64의 경우, AES-128의 128-비트 출력값 중 64 비트만을 MAC 값으로 출력하기 때문에 더 많은 오류 주입을 필요로 한다. 본 공격의 오류 주입 가정은 마지막 AES-128에 오류를 주입하여 이 블록 암호가 각각 7,8,9 라운드만 수행하도록 한다는 것이다 (그림 4 참조). 그러면 AES-128의 라운드 8,9,10의 64-비트 입·출력값을 알 수 있다. 이 값을 이용하여 128-비트 비밀키 K를 복구한다.

AES-CMAC-64에 대한 오류 주입 공격의 공격 과정은 다음과 같다. 본 공격의 데이터 복잡도는 1개의 메시지와 3번의 오류 주입이며, 계산 복잡도는 약 2^{56} AES-CMAC-64 연산이다.

1. 오류를 주입하지 않은 상태에서, AES-CMAC-64의 64-비트 MAC 값 $T(=O_{10}[0,1,\cdots,7])$를 얻는다. 그리고 오류를 주입한 AES-CMAC-64를 이용하여, AES-128의 라운드 7,8,9의 64-비트 출력값 $O_7[0,\cdots,7](=I_8[0,\cdots,7])$, $O_8[0,\cdots,7](=I_9[0,\cdots,7])$, $O_9[0,\cdots,7](=I_{10}[0,\cdots,7])$을 각각 계산한다.

2. $(I_{10}[0,\cdots,7], O_{10}[0,\cdots,7])$을 이용하여, 라운드 10의 32-비트 라운드 키 $RK_{10}[0,1,4,7]$을 계산한다 (그림 8 참조).

3. 라운드 9의 32-비트 입력값 $I_9[9,10,14,15]$를 추측한 후, $(I_9[0,\cdots,7], O_9[0,\cdots,7])$을 이용하여 라운드 9의 64-비트 라운드 키 $RK_9[0,\cdots,7]$을 계산한다 (그림 8 참조).

4. 라운드 8의 16-비트 입력값 $I_8[10,15]$를 추측한 후, $(I_8[0,\cdots,7], O_8[0,\cdots,7])$을 이용하여 라운드 8의 32-비트 라운드 키 $RK_8[0,1,2,3]$을 계산한다 (그림 8 참조).

5. 추측한 48-비트 값 $I_8[10,15]$, $I_9[9,10,14,15]$에 대해, AES-128의 키스케줄을 이용하여 라운드 8의 88-비트 라운드 키 $RK_8[4,5,6,7,8,9,10,12,13,14,15]$를 다음과 같이 계산한다.

$$RK_8[4] = RK_9[0] \oplus RK_9[4],$$
$$RK_8[5] = RK_9[1] \oplus RK_9[5],$$
$$RK_8[6] = RK_9[2] \oplus RK_9[6],$$
$$RK_8[7] = RK_9[3] \oplus RK_9[7],$$

$$RK_8[12] = \mathrm{SubWord}^{-1}(RK_8[3] \oplus RK_9[3] \oplus Rcon_9[3]),$$
$$RK_8[13] = \mathrm{SubWord}^{-1}(RK_8[2] \oplus RK_9[2] \oplus Rcon_9[2]),$$
$$RK_8[14] = \mathrm{SubWord}^{-1}(RK_8[1] \oplus RK_9[1] \oplus Rcon_9[1]),$$
$$RK_8[15] = \mathrm{SubWord}^{-1}(RK_8[0] \oplus RK_9[0] \oplus Rcon_9[0]),$$

$$RK_8[8] = RK_8[12] \oplus RK_9[4] \oplus \mathrm{SubWord}^{-1}(RK_9[3] \oplus RK_9[7] \oplus RK_{10}[7] \oplus Rcon_{10}[3]),$$
$$RK_8[9] = RK_8[13] \oplus RK_9[5] \oplus \mathrm{SubWord}^{-1}(RK_9[0] \oplus RK_{10}[0] \oplus Rcon_{10}[0]),$$
$$RK_8[10] = RK_8[14] \oplus RK_9[6] \oplus \mathrm{SubWord}^{-1}(RK_9[2] \oplus RK_{10}[1] \oplus Rcon_{10}[1]).$$

라운드 8의 나머지 8-비트 라운드 키 $RK_8[11]$을 추가로 추측한 후, 총 2^{56}개의 후보 라운드 키 RK_8로부터 후보 128-비트 비밀키 K를 복구한 후, 후보 서브키 $(K1, K2)$를 계산한다.

6. 각각의 후보 키 $(K, K1, K2)$와 주어진 메시지를 이용하여 AES-CMAC-64의 64-비트 MAC 값 T^*를 계산한다. T와 동일한 T^*를 생성하는 후보 비밀키 K를 AES-CMAC-64의 옳은 128-비트 비밀키로 출력한다.

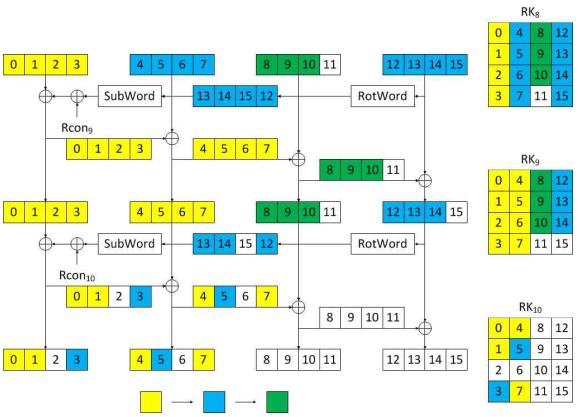

그림 9. AES-CMAC-64에 대한 오류 주입 공격의 단계 5
Fig. 5. Step 5 of our attack on AES-CMAC-64.

틀린 후보 비밀키가 단계 6을 통과할 확률은 2^{-64}이므로, 단계 6을 통과하는 틀린 후보 비밀키의 수의 기댓값은 2^{-8}이다. 이는 틀린 후보 비밀키가 옳은 비밀키로 출력될 확률이 낮음을 의미한다.

Ⅵ. 결 론

본 논문에서는 IETF 표준 MAC 알고리즘인 AES-CMAC에 대한 오류 주입 공격을 제안하였다. IETF에서 권장하는 AES-CMAC으로부터 생성되는 MAC 값의 최소 길이는 64 비트인데, 본 논문에서는 구체적인 예로서 64/96/128-비트 MAC 값을 생성하는 AES-CMAC에 대한 오류 주입 공격을 소개하였다. 본 논문에서 제안한 공격을 이용하여 매우 적은 수의 오류 주입만을 이용하여 AES-CMAC의 128-비트 비밀키를 복구할 수 있다. 다른 길이의 MAC 값을 생성하는 AES-CMAC의 경우도 본 논문에서 제안한 공격과 유사한 방식으로 적용 가능하다. 본 공격 결과는 AES-CMAC에 대한 첫 번째 키 복구 공격 결과이다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업의 연구결과로 수행되었음 (NIPA-2012-H0301-12-3007)

참 고 문 헌

[1] NIST Special Publication 800-38B, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", *NIST*, 2005.

[2] J. Black and P. Rogaway, "CBC MACs for arbitrary-length messages: The three key construction", *Crypto'00, LNCS 1880,* pp. 197-215, Springer-Verlag, Aug. 2000.

[3] FIPS 197, "Advanced Encryption Standard(AES)", *NIST*, 2001.

[4] RFC 4493, "The AES-CMAC Algorithm", *IETF,* 2006.

[5] RFC 4494, "The AES-CMAC-96 Algorithm and Its Use with IPsec", *IETF*, 2006.

[6] D. Boneh, R. DeMillo and R. Lipton, "On the importance of checking cryptographic protocols for faults", *Eurocrypt'97,*

LNCS 1233, pp. 37-51, Springer-Verlag, May 1997.
[7] 정기태, 성재철, 홍석희, “블록 암호 SEED에 대한 차분 오류 공격”, *정보보호학회논문지, 제 20권, 제 4호,* pp. 17-24, 2010. 08.
[8] R. Li, B. Sun, C. Li and J. You, “Differential Fault Analysis on SMS4 using a single fault”, *Information Processing Letters, Vol. 111*, pp. 156-163, Elsevier, Jan. 2011.
[9] H. Choukri and M. Tunstall, "Round Reduction Using Faults", *FDTC'05,* pp. 13-24, Sept. 2005.
[10] 최두식, 오두환, 배기석, 문상재, 하재철, “반복문 오류 주입을 이용한 Triple DES 차분 오류 공격”, *CISC-W'10,* pp. 308-312, 2010. 12.

정 기 태 (鄭基台)

2004년 2월 : 고려대학교 수학과 이학사
2006년 2월 : 고려대학교 정보보호 대학원 공학석사
2011년 8월 : 고려대학교 정보보호 대학원 공학박사
2011년 9월~현재 : 고려대학교 정보보호연구원 박사후연구원
관심분야 : 대칭키 암호의 분석 및 설계