情報保護學會論文誌
第19卷 第5號, 2009. 10

중간 암호문 복구 방법을 이용한 AES 차분오류공격

백 이 루,[1†] 길 광 은,[1] 박 제 훈,[2] 문 상 재,[2] 하 재 철[1‡]
[1]호서대학교, [2]경북대학교

Differential Fault Analysis on AES by Recovering of Intermediate Ciphertext

YiRoo Baek,[1†] KwangEun Gil,[1] JeaHoon Park,[2] SangJae Moon,[2] JaeCheol Ha[1‡]
[1]Hoseo University, [2]Kyungpook National University

요 약

최근 Li 등은 국내 표준 암호 알고리듬인 ARIA에 대해 차분 오류 분석(Differential Fault Analysis, DFA) 공격을 수행하면 평균 45개의 오류 암호문으로 공격이 가능함을 보였다. 본 논문에서는 Li 등의 공격 방법을 개선하여 AES에 대한 DFA 공격 방법을 제안하고자 한다. 제안하는 AES에 대한 DFA 공격은 최종 오류 암호문을 이용하여 마지막 라운드의 중간 암호문을 계산하고 마지막 라운드 키를 계산하는 기법이다. 본 논문에서는 AES에 대한 DFA 방법과 계산 복잡도를 분석하고 이를 컴퓨터 시뮬레이션한 결과, 1개의 정상 암호문과 2개의 오류 암호문으로 비밀 키를 찾아낼 수 있음을 검증하였다.

ABSTRACT

Recently, Li et al. proposed a new differential fault analysis(DFA) attack on the block cipher ARIA using about 45 ciphertexts. In this paper, we apply their DFA skill on AES and improve attack method and its analysis. The basic idea of our DFA method is that we recover intermediate ciphertexts in last round using final faulty ciphertexts and find out last round secret key. In addition, we present detail DFA procedure on AES and analysis of complexity. Furthermore, computer simulation result shows that we can recover its 128-bit secret key by introducing a correct ciphertext and 2 faulty ciphertexts.

Keywords: AES, ARIA, Differential Fault Analysis

I. 서 론

최근 부채널 공격에 대한 연구가 활발해 지면서 다양한 공격 기법과 대응 기법들이 제안되고 있다. 그 중에서 오류 주입(Fault Analysis, FA) 공격은 암호 디바이스의 실행 중에 정상 암호문과 오류를 주입하여 얻은 오류 암호문을 분석하여 해당 암호 디바이스에서 사용된 비밀 키를 추출할 수 있는 오류 공격 기법이다. 이러한 오류 공격은 1997년 Boneh 등이 RSA-CRT(Chinese Remainder Theorem) 서명 알고리즘의 동작에서 오류를 주입하여 같은 메시지에 대한 올바른 서명문과 오류가 주입된 서명문을 이용함으로써 비밀정보를 추출하는 공격 방법으로 처음 소개되었다[1]. 같은 해에 Biham과 Shamir가 블록 암호에 대해서 차분 오류 공격이 가능함을 제안하였고[2], 이를 차분 오류 분석(Differential Fault Analysis, DFA) 공격라고 부른다. 이후로 DFA 공격에 대한 연구가 활발하게 진행되었으며 그 중에서 AES[3] 알고리즘에서의 DFA에 대한 연구도 많이 이루어졌다. 국내에서는 RSA-CRT 알고리듬에 대한 연구가 주로 이루어졌으며 오류 공격을 방어할 수 있는 몇 가지 방식이 제안된 바 있다[4,5].

2003년에 Piret와 Quisquater는 2쌍의 정상/오류 암호문을 이용하여 128비트 AES 비밀 키를 찾는 방법을 제안하였고[6], 2005년에는 Giraud가 한 비

접수일(2009년 7월 24일), 게재확정일(2009년 9월 17일)
† 주저자: blr83@nate.com
‡ 교신저자: jcha@hoseo.edu

트 오류 또는 키 스케줄상의 한 바이트 오류를 주입하여 전체 키를 찾아내는 방법을 제안하였다[7]. 2007년에는 Takahashi 등이 AES 알고리듬의 키 스케줄링 상에 오류를 주입하여 비밀키를 찾는 방법으로 7개의 정상/오류 암호문 쌍을 가지고 전체 비밀키를 공격할 수 있음을 보였다[8]. 그리고 2008년에는 Kim과 Quisquater가 키 스케줄링 상에 오류를 주입함으로써 모두 8쌍의 정상/오류 암호문을 이용하여 키를 찾아내는 방법을 제안하였다[9].

본 논문에서는 2008년 Li 등이 제안한 ARIA 알고리즘에서의 DFA 공격 기법을[10,11] AES 알고리즘에 적용시켜 128비트 비밀 키를 찾을 수 있는 방법을 제안한다. 제안한 방법은 AES의 9라운드 열 혼합(MixColumns) 단계 이전의 특정 위치에 랜덤한 한 바이트의 오류를 가정할 경우, 2개의 암호문을 이용하여 4바이트의 10라운드 키를 찾을 수 있으며, 8개의 암호문만으로 10라운드 키 전체를 찾을 수 있다. 또한, AES의 8라운드 열 혼합 단계 이전에 한 바이트 오류를 주입하면 9라운드 열 혼합 단계에서 4개의 오류로 확산되는 성질이 있으므로, 결국 1개의 정상 암호문과 8라운드 열 혼합 단계에서 오류를 주입한 2개의 오류 암호문만으로도 최종 비밀 키를 계산할 수 있다. 그러나 현재까지 암호 알고리듬 수행시 한 바이트 단위까지 오류를 주입하는 공격이 실험적으로 성공하여 보고된 바는 없으나 오류 주입 기술 수준이 향상되면 조만간 현실화될 것이 예상된다[12]. 본 논문의 2장에서는 논문에 사용된 표기와 Li 등이 제안한 공격 기법을 소개한다. 제 3장에서는 AES에 대한 DFA 방법을 제안하고 4장에서 계산 복잡도 분석 및 시뮬레이션 결과를 보여 준다. 마지막으로 5장에서 결론을 맺는다.

II. 기존 ARIA 대한 DFA 공격

최근 Li 등은 국내 표준 블록 암호 알고리듬인 ARIA에 대한 DFA 공격을 제안하였다. ARIA는 2004년에 제안된 블록 암호 알고리듬으로서 128비트 메시지를 사용하고 128, 192, 256비트의 키를 사용한다. 본 장에서는 Li 등이 제안한 DFA 공격 방법을 분석하고자 한다.

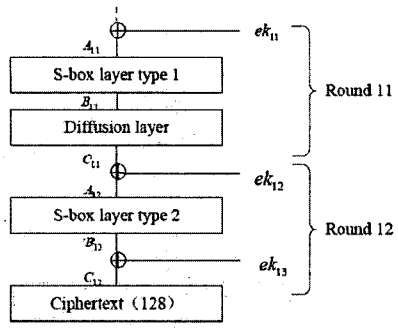
2.1 용어 및 표기

본 논문에서 기존의 DFA나 제안 방식을 설명하고 분석하는데 사용될 용어 및 표기는 다음과 같다.

- P : 평문
- SK : 비밀 키
- ek_r : r라운드 키
- Y, X : 암호문
- $\oplus$: XOR 연산
- $(X, X_1^*), (X, X_2^*)$: 정상 암호문과 오류가 주입된 암호문 쌍
- A_r : r라운드에서 치환 계층(S-Box Layer 혹은 SubBytes) 의 입력 값(16 byte)
- B_r : r라운드에서 치환 계층의 출력 값(16 byte)
- C_r : r라운드의 최종 출력 값(16 byte)
- a_i^r 또는 a_{ij}^r : r라운드에서 치환 계층의 입력 값 중 ARIA의 i번째 바이트 또는 AES의 i행 j열의 바이트
- b_i^r 또는 b_{ij}^r : r라운드에서 치환 계층의 출력 값 중 ARIA의 i번째 바이트 또는 AES의 i행 j열의 바이트
- c_i^r 또는 c_{ij}^r : r라운드의 최종 출력 값 중 ARIA의 i번째 바이트 또는 AES의 i행 j열의 바이트
- Δ : 오류가 없는 경우의 중간값과 오류가 주입된 경우의 중간 값의 차분
- $IN(x, y)$: 치환 계층에서 한 바이트의 입력 차분 x에 대한 출력 차분 y를 갖는 입력 값
- $SL(x)$: ARIA의 치환 계층 함수(S-Box Layer) 또는 AES의 바이트 치환함수(SubBytes)
- $SL^{-1}(x)$: ARIA의 역 치환 함수 또는 AES의 역 바이트 치환함수
- $DL(x)$: ARIA의 확산 계층(Diffusion Layer) 함수 또는 AES의 열 혼합(MixColumns) 단계 함수
- $DL^{-1}(x)$: ARIA의 확산 계층의 역함수 또는 AES의 열 혼합 단계 역함수

2.2 Li 등의 ARIA에 대한 DFA

Li 등은 ARIA에 대한 DFA를 처음 제안하였는데 그림 1은 ARIA 암호 알고리듬의 11라운드와 12라운드를 나타낸 것이다. Li 등은 논문에서 오류는 마지막 라운드의 확산 계층 이전에 한 바이트의 오류를 주입한다고 가정하였다. 즉, [그림 1]에서 B_{11}의 값 중에서 한 바이트에 오류가 주입되지만 오류 바이트의 위치와 주입된 오류 값은 모른다는 가정하였다. 그리고



〔그림 1〕 ARIA의 11, 12라운드

공격자는 평문을 선택하여 암호화할 수 있고, 한 평문으로부터 정상 암호문과 오류가 주입된 여러 개의 오류 암호문을 얻을 수 있다고 전제하였다.

구체적인 공격 과정은 다음과 같이 6단계로 구성되어 있다.

■ 단계 1 : 평문 P를 비밀 키 SK로 암호화하여 정상 암호문 Y를 얻는다.

■ 단계 2 : 다음 과정을 실행하여 ek_{13}을 찾는다.

(a) 암호화 단계에서 11라운드 확산 계층 이전에 랜덤한 오류를 주입하여 얻은 암호문 Y^*와 정상 암호문 Y로 $\Delta Y = Y \oplus Y^*$를 계산하여 차분 ΔY를 얻는다. 마지막 라운드는 확산 계층이 라운드 키 덧셈으로 바뀌므로 ΔB_{12}와 ΔC_{12}은 다음과 같이 계산할 수 있다.

$$\Delta C_{12} = \Delta Y$$
$$\Delta B_{12} = \Delta C_{12} \oplus \Delta ek_{13} = \Delta Y$$

(b) 12라운드에서 j번째($0 \le j \le 15$) S-box의 출력 차분은 $\Delta b_{12} = (\Delta B_{12})_j = (\Delta Y)_j$를 통해 나타낼 수 있으며 A_{12}의 j번째($0 \le j \le 15$) 바이트는 다음을 만족한다.

$$a_{j,12} \in IN(\Delta a_{j,12}, \Delta b_{j,12})$$
$$IN(\Delta a_{j,12}, \Delta b_{j,12}) = \{a_{j,12} | a_{j,12} \in \{0,1\}^8, SL(a_{j,12}) \oplus SL(a_{j,12} \oplus \Delta a_{j,12}) = \Delta b_{j,12}\}$$

(c) A_{12}의 값을 추론하기 위해 ΔA_{12}의 값에 대한 전탐색을 수행한다. 이때 ΔA_{12}는 한 바이트 오류 주입을 가정하면 $4080 (= 16 \times (2^8 - 1))$의 가능성을 갖는데, 이때 (b)의 식을 만족하는 A_{12} 값이 많은 후보자를 가지기 때문에 같은 평문에 다른 오류를 주입한 여러 개의 암호문을 수집한 후 같은 방법을 반복함으로써 A_{12}의 후보자 수를 줄일 수 있다. 이렇게 하여 A_{12}의 모든 바이트들을 구한다.

(d) A_{12}의 값을 이용하여 다음을 계산함으로써 ek_{13}을 구한다.

$$B_{12} = SL(A_{12})$$
$$ek_{13} = Y \oplus B_{12} = Y \oplus SL(A_{12})$$

■ 단계 3 : 단계 2와 유사한 방법으로 10라운드 확산 계층 이전에 랜덤한 오류를 주입하여 얻은 암호문 Y^*와 이전 단계에서 얻은 ek_{13}을 이용하여 ek_{12}를 찾는다.

(a) 오류 암호문 Y^*와 ek_{13}을 이용하여 마지막 라운드에서 S-Box의 오류가 주입된 입력 A_{12}^*와 출력 B_{12}^*는 다음과 같이 유도될 수 있다.

$$B_{12}^* = Y^* \oplus ek_{13}$$
$$A_{12}^* = SL^{-1}(B_{12}^*) = SL^{-1}(Y^* \oplus ek_{13})$$

11라운드에서 S-Box의 출력 차분은 다음과 같이 계산할 수 있다.

$$\Delta B_{11} = DL^{-1}(\Delta C_{11}) = DL^{-1}(\Delta A_{12} \oplus \Delta ek_{12}) =$$
$$DL^{-1}(\Delta A_{12}) = DL^{-1}(A_{12} \oplus A_{12}^*)$$
$$= DL^{-1}(A_{12} \oplus SL^{-1}(Y^* \oplus ek_{13}))$$

(b) 11라운드에서 j번째 S-box의 출력 차분은 $\Delta b_{j,11} = (\Delta B_{11})_j = (DL^{-1}(A_{12} \oplus SL^{-1}(Y^* \oplus ek_{13})))_j$를 통해 나타낼 수 있으며 A_{11}의 j번째($0 \le j \le 15$) 바이트는 다음을 만족한다.

$$a_{j,11} \in IN(\Delta a_{j,11}, \Delta b_{j,11})$$

(c) A_{11}의 값을 추론하기 위해 ΔA_{11}의 값에 대한 전탐색을 수행하여 단계 2의 (c)와 같은 방법으로 A_{11}의 모든 바이트들을 구한다.

(d) A_{11}을 이용하여 다음을 계산함으로써 ek_{12}를 구한다.

$$B_{11} = SL(A_{11})$$
$$C_{11} = DL(B_{11}) = DL(SL(A_{11}))$$
$$ek_{12} = A_{12} \oplus C_{11} = A_{12} \oplus DL(SL(A_{11}))$$

■ 단계 4 : 단계 3의 (a)-(d)와 같은 방법으로 ek_{11}을 구한다.

■ 단계 5 : 단계 3의 (a)-(d)와 같은 방법으로 ek_{10}을 구한다.

■ 단계 6 : 지금까지 구한 4개의 라운드 키를 이용하여 비밀키를 복구한다.

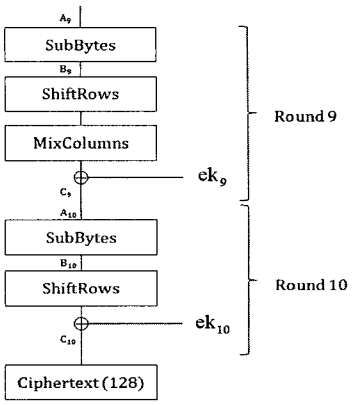
Li 등은 언급한 오류 주입 공격을 통해 평균 11개의 오류를 주입하면 하나의 128비트 라운드 키를 찾을 수 있음을 보였다. 또한 ARIA에서는 최종 비밀키를 찾기 위해서는 모두 4개의 라운드 키가 필요하다.

결론적으로 평균 45개의 오류 암호문을 사용하여 ARIA의 비밀 키를 복구할 수 있음을 컴퓨터 시뮬레이션으로 보였다. 여기서 고려할 점은 11라운드의 확산 계층 이전의 한 바이트 오류는 7바이트의 오류 출력을 가지게 되고 그 위치를 알 수 있으므로 실제 공격시 $\triangle A_{12}$는 $1785(=7\times(2^8-1))$번의 전탐색을 수행하게 된다.

III. AES에 대한 DFA 제안

AES는 2001년 미국 NIST에 의해 연방 정보처리 표준으로 지정된 대칭키 암호화 방식으로 128비트 블록과 128, 192, 256비트의 키를 사용한다. 이 장에서는 Li 등의 DFA 공격 기법을 AES에 적용해 보기로 한다. 먼저ARIA와 AES 알고리즘에서 DFA 공격의 차이점을 알아보면 먼저 ARIA는 11라운드 확산 계층 이전에 한 바이트의 오류 주입하면 확산 계층에 의해 오류가 7바이트로 전파되기 때문에 이를 이용하여 오류가 전파된 7바이트 위치의 라운드 키를 찾을 수 있다. 반면, AES는 9라운드 열 혼합 단계 이전에 한 바이트의 오류를 주입하면 열 혼합 단계에 의해 오류가 4바이트로 전파되므로 이를 이용하여 4바이트의 라운드 키를 찾을 수 있다. 그러나 여기에서 중요한 차이점이 나타난다. ARIA에서의 확산 계층은 순수하게 XOR 연산만으로 이루어져 있는 반면 AES에서는 XOR 연산뿐만 아니라 입력 값에 대해 $GF(2^8)$ 상에서 배수(2배, 혹은 3배) 연산을 처리해야 한다. 하지만 여기서 주목할 점은 $GF(2^8)$ 상에서의 배수 처리를 자세히 분석해 보면 쉬프트(shift) 연산과 XOR 연산으로 처리할 수 있어 선형성(linearity)를 유지한다는 점이다. 두 번째 차이점으로는 ARIA에서는 최소 4개의 라운드 키를 찾아야 완전한 비밀 키를 찾을 수 있는 반면, AES는 마지막 라운드 키만 찾으면 비밀 키를 복구할 수 있다. 따라서 ARIA에서 보다 AES에서 DFA 공격이 수월하다고 할 수 있다.

제안하는 공격 기법은 Li 등이 ARIA에 적용했던 DFA 공격 기법을 개선하여 AES에 적용한 것이다. AES는 한 바이트의 오류 주입으로 4바이트의 라운드 키를 찾을 수 있으므로 각각 다른 열(column)에 4개의 오류를 주입하게 되면 16바이트의 전체 라운드 키를 찾을 수 있다. [그림 2]는 AES 암호 알고리듬의 9라운드와 10라운드를 나타낸 것이다. 제안하는 공격 기법에서 오류 주입에 관한 가정은 Li 등의 DFA과 동일하며 공격자는 9라운드 열 혼합 단계 이전에서 특정한 위치에 한 바이트 오류를 주입할 수 있다고 가정한다.



[그림 2] AES의 9, 10라운드

구체적인 공격 과정은 다음과 같이 3단계로 구성되어 있다.

■ 단계 1 : 평문 P를 비밀 키 SK로 암호화하여 정상 암호문 Y를 얻는다.

■ 단계 2 : 다음을 실행하여 ek_{10}을 찾는다.

(a) 암호화 단계에서 9라운드 열 혼합 단계 이전에 랜덤한 오류를 주입하여 얻은 암호문 Y^*와 정상 암호문 Y로 $\triangle Y = Y \oplus Y^*$를 계산하여 $\triangle Y$를 얻는다. 마지막 라운드는 열 혼합 단계가 빠지므로 $\triangle B_{10}$와 $\triangle C_{10}$은 다음과 같이 계산할 수 있다.

$$\triangle C_{10} = \triangle Y$$
$$\triangle B_{10} = \triangle C_{10} \oplus \triangle ek_{10} = \triangle Y$$

(b) 10라운드에서 (i,j)번째$(0 \le i \le 3, 0 \le j \le 3)$ S-box의 출력 차분은 $\triangle b_{i,j}^{10} = (\triangle B_{10})_{i,j} = (\triangle Y)_{i,j}$를 통해 나타낼 수 있으며 A_{10}의 (i,j)번째$(0 \le i \le 3, 0 \le j \le 3)$ 바이트는 다음을 만족한다.

$$a_{i,j}^{10} \in IN(\triangle a_{i,j}^{10}, \triangle b_{i,j}^{10})$$
$$IN(\triangle a_{i,j}^{r}, \triangle b_{i,j}^{r}) =$$
$$\{a_{i,j}^{r} | a_{i,j}^{r} \in \{0,1\}^8, SL(a_{i,j}^{r}) \oplus SL(a_{i,j}^{r} \oplus \triangle a_{i,j}^{r}) = \triangle b_{i,j}^{r}\}$$

(c) A_{10}의 값을 추론하기 위해 $\triangle A_{10}$의 값에 대한 전탐색을 수행한다. 이때 $\triangle A_{10}$는 한 바이트 오류 주입을 가정하면, $4080(=16\times(2^8-1))$의 가능성을 갖는데, 이때 (b)의 식을 만족하는 A_{10}의 값이 많은 후보자를 가지기 때문에 같은 평문에 다른 오류를 주입한

여러 개의 오류 암호문을 수집하여 같은 방법을 반복함으로써 A_{10}의 후보자 수를 줄일 수 있다. 제안한 방법에서는 2개의 오류 암호문만으로 충분한데 이에 대한 분석은 4장에서 다룬다. 이와 같은 방법으로 총 8번의 오류를 주입하여 얻은 오류 암호문으로부터 A_{10}의 모든 바이트들을 구할 수 있다. 이때 주입되는 8개의 오류는 각 열에 각각 2개씩 주입되어야 한다.

(d) A_{10}의 값을 이용하여 다음을 계산함으로써 ek_{10}을 구한다.

$$B_{10} = SL(A_{10})$$
$$ek_{10} = Y \oplus B_{10} = Y \oplus SL(A_{10})$$

■ 단계 3 : 단계 2에서 얻은 ek_{10}을 이용하여 비밀키 SK를 계산한다.

제안하는 DFA 공격의 핵심 내용은 단계 2의 (c) 과정이므로 상술하기로 한다. 논문에서는 용이한 설명을 위하여 오류가 state (0, 0)에 주입되었다고 가정한다. 9라운드 확산 계층 이전에 주입된 오류는 이후에 4바이트로 전파되므로 열 혼합 단계 이후 4바이트의 차분 값은 다음과 같은 관계를 갖는다. 즉, 이것은 열 혼합 단계에서 state (0, 0)에 주입된 오류가 1배, 2배, 3배가 되어 전파됨을 의미한다.

$$SL(a_{0,0}) \oplus SL(a_{0,0} \oplus \Delta a_{0,0}) = \Delta b_{0,0}$$
$$SL(a_{1,0}) \oplus SL(a_{1,0} \oplus \Delta a_{1,0}) = \Delta b_{1,0}$$
$$SL(a_{2,0}) \oplus SL(a_{2,0} \oplus \Delta a_{2,0}) = \Delta b_{2,0}$$
$$SL(a_{3,0}) \oplus SL(a_{3,0} \oplus \Delta a_{3,0}) = \Delta b_{3,0}$$

$$\Delta a_{1,0} = \Delta a_{2,0}$$
$$\Delta a_{0,0} = 02 \cdot \Delta a_{1,0} = 02 \cdot \Delta a_{2,0}$$
$$\Delta a_{3,0} = 03 \cdot \Delta a_{1,0} = 03 \cdot \Delta a_{2,0}$$

위의 공격 방법에서 단계 2의 (c) 과정을 자세하게 기술하면 아래와 같다.

■ (c)-1 : S-box의 입력에서 4080개의 가능한 차분 ΔA_{10}를 계산하여 리스트 L로 저장한다.

■ (c)-2 : ΔA_{10}의 4바이트에 대한 전탐색을 수행한다. 단, ΔA_{10} 중에서 0이 아닌 처음 2바이트를 $(\Delta a^{10}_{0,0}, \Delta a^{10}_{1,0})$로 나타낸다.

① $(\Delta a_{0,12}, \Delta a_{1,12})$의 모든 가능한 후보에 대하여 다음을 계산한다.

$$(SL(a^{10}_{0,0}) \oplus SL(a^{10}_{0,0} \oplus \Delta a^{10}_{0,0}),$$
$$SL(a^{10}_{1,0}) \oplus SL(a^{10}_{1,0} \oplus \Delta a^{10}_{1,0}))$$
$$= (\Delta b^{10}_{0,0}, \Delta b^{10}_{1,0}) = (\Delta y_0, \Delta y_1)$$

이때 $\Delta a^{10}_{0,0}, \Delta a^{10}_{1,0}$은 $\Delta a^{10}_{0,0} = 02 \cdot \Delta a^{10}_{1,0}$의 관계가 성립하므로 후보의 선택을 위해 255개의 가능한 $\Delta a^{10}_{1,0}$ 값만을 고려하면 된다.

② 4080개의 리스트 L의 첫 번째, 두 번째 바이트와 ①에서 계산한 2바이트 값들을 비교하여 일치되는 값들을 찾았을 때, 리스트 $R = (a^{10}_{0,0}, a^{10}_{1,0})$을 생성하여 그 두 값을 R에 추가한다.

■ (c)-3 : 각각의 $r \in R$에 대해서 한 바이트의 확장을 시도한다.

① R로부터 r의 한 바이트 값을 가져온다.

② $(\Delta a_{1,10}, \Delta a_{2,10})$의 모든 가능한 후보에 대해서 다음을 계산한다.

$$(SL(r_t) \oplus SL(r_t \oplus \Delta a^{10}_{1,0}),\ SL(a^{10}_{2,0}) \oplus SL(a^{10}_{2,0} \oplus \Delta a^{10}_{2,0}))$$
$$= (\Delta b^{10}_{1,0}, \Delta b^{10}_{2,0}) = (\Delta y_1, \Delta y_2)$$

r_t는 r의 마지막 바이트를 나타내고, 이때 $\Delta a^{10}_{1,0} = \Delta a^{10}_{2,0}$의 관계를 갖는다.

③ 리스트 L에 있는 두 번째, 세 번째 바이트와 $(r_t, a^{10}_{2,0})$을 비교하여 일치되는 것을 찾았을 때, 기존의 리스트 R에 획득한 한 바이트 값을 추가하여 새로운 리스트 $R = (a^{10}_{0,0}, a^{10}_{1,0}, a^{10}_{2,0})$을 만든다.

■ (c)-4 : 리스트 R의 원소들이 4바이트의 길이가 될 때까지 한 바이트씩 확장하여 (c)-3를 반복한다.

■ (c)-5 : 리스트 R이 오직 하나의 값을 가질 때까지 같은 평문에 다른 오류를 주입하여 (c)-2부터 4까지 반복한다.

IV. 계산 복잡도 분석 및 시뮬레이션

이 장에서는 제안 기법이 공격에 성공하기 위한 계산 복잡도를 분석해 보고 시뮬레이션을 통해 성공할 수 있음을 보이고자 한다.

4.1 계산 복잡도 분석

본 논문에서는 A_{10}를 구하고 이를 이용하여 10라운드 키를 구하는 방식이다. 이 경우 A_{10}을 구하기 위해서는 어느 정도의 공격 빈도가 필요한지 분석해 본다. 먼저 A_{10}을 구하는 단계에서(제안 기법의 단계 2 (c)) 입력 값의 차분 값, ΔA_{10}의 수는 255^n개 이다. 실제로 AES인 경우 하나의 오류 바이트로 4개의 차분 값을 만들 수 있으므로 $n = 4$이다. 그 중에서 리스트 L의 원소들은

$255n$개가 된다. 따라서 한 번의 오류 주입 공격 시도로 A_{10}의 후보 수를 다음과 같은 비율로 줄일 수 있다.

$$\frac{255n}{255^n} = n \times 255^{(1-n)}$$

그러나 한 바이트의 오류가 주입되었을 때, $\triangle A_{10}$의 4바이트의 관계는 열 혼합 단계의 특성상 배수 관계가 성립되므로 하나의 차분 값만을 이용하여 계산할 수 있기 때문에 $255n$이 아닌 255개의 가능성만을 고려하면 된다(제안 기법 단계 2 (c)의 단계 2 참고). 따라서 위의 식은 아래와 같이 다시 표현할 수 있다.

$$\frac{255}{255^n} = 255^{(1-n)}$$

즉, 하나의 정상/오류 암호문 쌍을 이용하면 A_{10}의 후보자는 $255^{(1-n)}$ 비율로 줄일 수 있다. 그러므로 N개의 오류 암호문을 이용했을 때는 A_{10}의 가능한 후보의 수는 $256^n(255^{1-n})^N$개가 됨을 알 수 있다. 예를 들어 한 쌍의 암호문 (Y, Y^*)를 이용했을 때, A_{10}의 4바이트에 대해 가능한 후보의 수는 $256^4(255^{-3})$으로 약 259개가 될 수 있다. 그런데 2개의 오류 암호문을 이용하면 $256^4(255^{-3})^2 = 0.000016$이 되므로 가능한 후보의 수가 1개 미만이 되어 정확한 A_{10}을 찾을 수 있다.

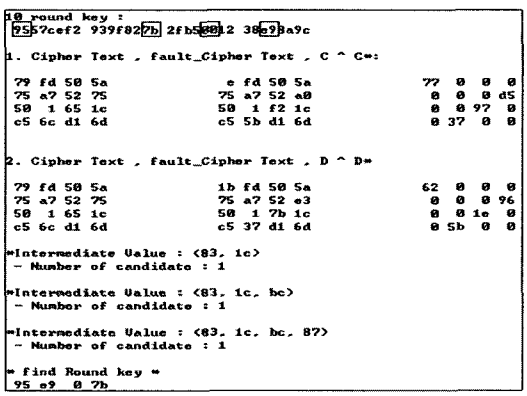
그러나 제안 논문에서는 A_{10}를 모두 찾기 위해 총 4바이트 중에서 두 바이트씩 가능한 후보를 찾고 있다. 따라서 이 경우는 $n=2$라고 둘 수 있다. 이때 한 쌍의 암호문을 사용한다면 A_{10}의 2바이트인 $(a^{10}_{0,0}, a^{10}_{1,0})$ 값의 후보수는 $256^2(255^{-1}) = 257.0039$개가 된다. 그러나 이것으로는 $(a^{10}_{0,0}, a^{10}_{1,0})$를 결정하기 어렵기 때문에 후보의 수를 줄이기 위해 같은 평문에 다른 오류가 주입된 두 쌍의 암호문 $(X, X_1^*), (X, X_2^*)$를 사용한다. 이 경우, 2바이트 $(a^{10}_{0,0}, a^{10}_{1,0})$의 후보는 $256^2(255^{-1})^2 = 1.0079$가 되어 거의 1개로 결정됨을 알 수 있다. 또한 2개의 오류 암호문을 이용하여 A_{10}의 3바이트 $(a^{10}_{0,0}, a^{10}_{1,0}, a^{10}_{2,0})$를 찾으면 후보의 수가 $256^3(255^{-2})^2 = 0.004$가 됨을 알 수 있다. 또한, 위에서 언급한 바와 같이 4바이트 $(a^{10}_{0,0}, a^{10}_{1,0}, a^{10}_{2,0}, a^{10}_{3,0})$를 찾으면 후보의 수가 $256^4(255^{-3})^2 = 0.000016$이 되어 확실하게 하나의 라운드 키를 찾아낼 수 있다. 따라서 마지막 라운드의 A_{10}을 찾는데 1개의 정상 암호문과 2개의 오류 암호문만으로도 충분하다.

분석 결과, AES의 9라운드 열 혼합 단계 이전의 특정 위치에 랜덤한 한 바이트 오류를 가정하고, 2개의 암호문을 이용하여 10라운드 키의 4바이트를 찾을 수 있으며, 각 열에 2개씩 모두 8개의 암호문만으로 10라운드 키 전체를 찾을 수 있다. 그런데 8라운드 열 혼합 단계 이전의 한 바이트 오류를 주입하면 9라운드 열 혼합 단계에서 4개의 오류로 확산되는 성질을 가지고 있다. 따라서 8라운드 열 혼합 단계에서 한 바이트 오류는 자연스럽게 9라운드 열 혼합 단계의 4바이트 오류를 유발하게 된다. 결국, 8라운드 열 혼합 단계에서 오류를 주입한다면, 2개의 오류 암호문만으로도 9라운드 열 혼합 단계의 8개의 오류를 유발한 것과 같으므로 최종 비밀 키를 계산할 수 있다.

4.2 시뮬레이션 결과

제안한 DFA 공격 기법을 컴퓨터를 이용하여 시뮬레이션해 보았다. 시뮬레이션은 AMD 애슬론 3500 프로세서, 1G RAM을 갖춘 PC에서 수행되었고, 개발 도구는 Visual Studio 6.0을 사용하였다. [그림 3]은 시뮬레이션한 결과의 일부를 보여 주고 있다.

시뮬레이션 화면은 AES 암호화 과정에서 9라운드 열 혼합 단계 이전의 state (0, 0) 위치에 랜덤한 한 바이트 오류를 주입하고, 하나의 평문에 대해 2개의 오류 암호문을 이용하여 10라운드 키 중 4바이트를 찾은 것이다. 그림의 첫 번째 줄은 사용된 10라운드 키를 나타낸 것이며 맨 밑에서 바로 윗줄은 10라운드의 중간 암호문 값 A_{10}을 구한 것이다. 그리고 맨 밑줄의 값이 실제 10라운드 키의 일부인 4바이트를 구한 것이다. 시뮬레이션 결과에서 알 수 있듯이 하나의 평문에 대해 2개의 오류 암호문을 이용하여 A_{10} 후보 수가 1개로 나



[그림 3] 제안한 공격 기법의 시뮬레이션 화면

타남을 볼 수 있고, 10라운드 키의 4바이트를 정확하게 찾아 낼 수가 있었다. 동일한 방법으로 state (0, 1), state (0, 2), state (0, 3)에 오류를 주입하면 각각 4바이트씩의 라운드 키를 구할 수 있었다. 최종적으로 하나의 평문에 대해 총 8개의 오류 암호문을 이용하여 128비트의 비밀키 SK를 구할 수 있었다.

또한 위에서 언급한 바와 같이 8라운드 열 혼합 단계에서 바이트 오류가 주입된 경우에는 2개의 오류 암호문과 1개의 정상 암호문만으로도 비밀 키를 계산해 낼 수 있었다. 이 결과는 AES에 대한 DFA 공격 중에서 가장 복잡도가 낮은 Piret-Quisquater방법8[6]과 비슷한 복잡도를 갖는다. 다만, Piret-Quisquater 방식에서는 2쌍의 정상/오류 암호문이 필요한 반면, 제안 방식에서는 1개의 정상 암호문과 2개의 오류 암호문만이 필요하다. 이와 같은 결과의 차이는 Piret-Quisquater 방식에서는 비밀 키를 추측하는 방식이므로 매번 서로 다른 평문을 사용해야 하지만, 제안 방식에서는 암호 중간 값을 계산하는 형태이므로 동일한 평문에 대한 서로 다른 오류 암호문 2개가 필요하기 때문이다.

V. 결 론

본 논문에서는 2008년 Li 등이 제안한 ARIA에서의 DFA 공격 기법을 AES에 적용하여 비밀 키를 찾을 수 있는 방법을 제안하였다. 제안 방식에서는 암호화 과정에서 공격자가 8라운드나 9라운드 열 혼합 단계 이전에 특정 한 바이트에 오류를 주입할 수 있다고 가정하였다. AES의 열 혼합 단계에서 입력은 선형성을 유지하면서 확산되므로 본 논문에서는 여러 개의 출력 차분으로부터 하나의 입력 차분을 추측하는 과정에서 중간 암호 값의 후보를 줄여가는 방식을 사용하였다. 시뮬레이션 결과, 제안 공격 방법은 8라운드 열 혼합 단계에 한 바이트 오류를 주입할 경우, 단지 2개의 오류 암호문만으로 AES의 128비트 비밀 키를 찾을 수 있었다. 따라서 암호용 디바이스에 암호 알고리듬을 구현할 경우에는 칩 설계자들이 물리적 오류 주입 공격에 대응할 수 있는 방어책을 마련하는 것이 필요하다.

참 고 문 헌

[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] National Institute of Standards and Technology, "Advanced Encryption Standards," NIST FIPS PUB 197, 2001.

[4] 하재철, 박제훈, 문상재, "오류 확산 기법을 이용한 CRT-RSA 오류 주입 공격 대응 방안," 정보보호학회논문지, 18(2), pp. 75-84, 2008년 4월.

[5] 김성경, 김태현, 한동국, 박영호, 홍석희, "비교연산을 사용하지 않는 오류주입 공격에 안전한 CRT 기반의 RSA," 정보보호학회논문지, 18(4), pp. 17-25, 2008년 8월.

[6] 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.

[7] C. Giraud, "DFA on AES," Advanced Encryption Standard-AES'04, LNCS 3373, pp. 27-41, 2005.

[8] J. Takahashi, T. Fukunaga, and K. Yamakoshi, "DFA mechanism on the AES key schedule," Workshop on Fault Diagnosis and Tolerance in Cryptography-FDTC'07, pp. 62-72, Sep. 2007.

[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.

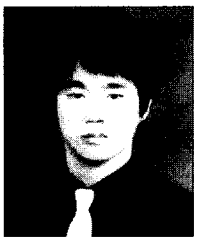
[10] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han, and J. Hong, "New Block Cipher: ARIA," ICISC'03, LNCS 2971, pp. 432-445, 2003.

[11] W. Li, D. Gu, and J. Li, "Differential fault analysis on the ARIA algorithm," Information Sciences, vol. 178, issue. 19, pp. 3727-3737, Oct. 2008.

[12] 박제훈, 문상재, 하재철, "CRT-RSA 암호시스템에 대한 광학적 오류 주입 공격의 실험적 연구," 정보보호학회논문지, 19(3), pp. 51-60, 2009년 6월.

〈著者紹介〉

백 이 루 (Yi Roo Baek) 학생회원
2008년 8월: 호서대학교 정보보호학과 졸업
2008년 9월 ~ 현재: 호서대학교 정보보호학과 석사과정
〈관심분야〉 네트워크 보안, 프로토콜, 스마트 카드 보안



길 광 은 (Kwang Eun Gil) 학생회원
2008년 2월: 호서대학교 정보보호학과 졸업
2008년 3월 ~ 현재: 호서대학교 정보보호학과 석사과정
〈관심분야〉 네트워크 보안, 암호 알고리듬, 인증 및 평가

박 제 훈 (JeaHoon Park) 학생회원
2004년 2월: 경북대학교 전자·전기공학부 졸업
2006년 2월: 경북대학교 전자공학과 석사
2006년 3월 ~ 현재: 경북대학교 전자공학과 박사과정
〈관심분야〉 정보보호, 네트워크 보안, 스마트카드 보안

문 상 재 (SangJae Moon) 종신회원
1972년 2월: 서울대학교 공업교육(전자전공)과 학사
1974년 2월: 서울대학교 전자공학과 석사
1984년 6월: 미국 UCLA 전기공학과 박사
1984년 7월 ~ 1985년 6월: UCLA Postdoctor 근무
1984년 7월 ~ 1985년 6월: 미국 OMNET 컨설턴트
1997년 9월 ~ 1998년 8월: 경북대학교 전자전기공학부 학부장
1974년 12월 ~ 현재: 경북대학교 전자전기컴퓨터공학부 교수
2000년 8월 ~ 현재: 경북대학교 이동네트워크 정보보호기술 연구센터장
2002년 2월 ~ 현재: 한국정보보호학회 명예회장
〈관심분야〉 정보보호, 디지털 통신, 이동 네트워크

하 재 철 (Jae Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월 ~ 2006년 1월: 나사렛대학교 전자계산소장, 학술정보관장, 입시학생처장
1998년 3월 ~ 2007년 2월: 나사렛대학교 정보통신학과 부교수
2006년 7월 ~ 2006년 12월: QUT in Australia 연구 교수
2007년 3월 ~ 현재: 호서대학교 정보호호학과 부교수
2002년 3월 ~ 현재: 한국정보보호학회 이사, 논문지 편집위원
2009년 1월 ~ 현재: 한국산학기술학회 이사
〈관심분야〉 정보보호, 네트워크 보안, 스마트카드 보안