무선 센서 네트워크 환경에 적합한 블록 암호 LED-64에 대한 안전성 분석

Security Analysis of Block Cipher LED-64 Suitable for Wireless Sensor Network Environments

정기태*

Ki-Tae Jeong*

요 약

CHES 2011에 제안된 64-비트 블록 암호 LED-64는 WSN과 같은 제한된 환경에서 효율적으로 구현이 가능하도록 설계된 블록 암호이다. 본 논문에서는 LED-64에 대한 차분 오류 공격을 제안한다. 본 논문에서 소개하는 공격은, 1개의 랜덤 니블 오류와 2^8의 전수조사를 이용하여, LED-64의 비밀키를 복구한다. 본 논문의 공격 결과는 LED-64에 대한 첫 번째 공격 결과이다.

Abstract

LED-64 is a 64-bit block cipher proposed in CHES 2011 and suitable for the efficient implementation in constrained hardware environments such as WSN. In this paper, we propose a differential fault analysis on LED-64. In order to recover the secret key of LED-64, this attack requires only one random nibble fault and an exhaustive search of 2^8. This work is the first known cryptanalytic result on LED-64.

Key words : Block Cipher, LED-64, differential fault analysis

I. 서 론

블록 암호에 대한 안전성 분석 기법은 차분 공격(differential cryptanalysis) [1] 등과 같이 알고리즘 자체의 이론적인 취약점을 이용하는 기법과, 부채널 공격(side channel attack) 등과 같이 암호 시스템의 실질적인 구현 과정에서 얻어지는 정보들을 이용하는 기법으로 분류된다.

부채널 공격은 암호 알고리즘을 구현하였을 때 발생하는 연산 시간, 전력, 전자기파, 오류 등의 부가적인 정보를 이용하는 공격 방법으로서, 오류 주입 공격(fault injection attack), 시차 공격(timing attack), 전력 분석 공격(power attack) 등이 있다. 1997년 Biham과 Shamir는 오류 주입 공격을 대칭키 암호 시스템에 최초로 적용하여 블록 암호 DES를 분석하였다 [2]. 차분 오류 공격(differential fault analysis, DFA)라 불리는 이 공격은 기존의 차분 공격을 오류 주입 공격에 결합하여 DES 뿐만 아니라 AES [3], ARIA [4], SEED [5] 등 대부분의 블록 암호에 적용되었다.

최근, WSN(wireless sensor network)과 같은 제한된 환경에서 효율적으로 동작하는 경량 블록 암호에 대

* 고려대학교 정보보호연구원(Center for Information Security Technologies, Korea University)
· 제1저자 (First Author) : 정기태
· 투고일자 : 2012년 2월 2일
· 심사(수정)일자 : 2012년 2월 2일 (수정일자 : 2012년 2월 20일)
· 게재일자 : 2012년 2월 28일

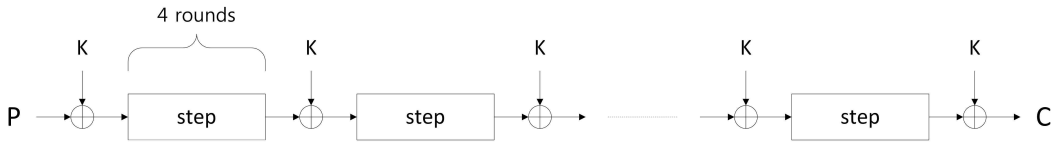

그림 1. LED-64의 암호화 과정
Fig. 1. Encryption process of LED-64.

한 연구가 활발히 진행 중이다. 그 결과로서, KATAN/KTANTAN [6], PRINTcipher [7], LED [8], Piccolo [9] 등이 제안되었다. 이 알고리즘들은 다양한 설계 논리에 기반을 두어 단순한 구조로 설계되었기 때문에, 제한된 환경에서도 높은 효율성을 갖는다.

CHES 2011에서 제안된 경량 블록 암호 LED는 64-비트 블록 암호로서 64/80/96/128-비트 비밀키를 사용한다. 비밀키의 길이에 따라, 각각 LED-64, LED-80, LED-96, LED-128로 표기된다. LED-64의 전체 라운드 수는 32이고, 나머지 LED-80/96/128의 전체 라운드 수는 48이다. 이 알고리즘은 키스케줄이 없어서 매 라운드마다 동일한 라운드 키가 사용된다는 특징을 갖고 있다. 또한, 전체 구조는 블록 암호 AES와 유사한 구조를 가지며 경량화를 위해 간단한 행렬을 이용하여 MDS 행렬을 생성한다. 기제안된 LED에 대한 안전성 분석 결과는 제안 논문에서 소개된 결과가 유일하다.

본 논문에서는 LED-64에 대한 차분 오류 공격을 제안한다. 본 논문에서 소개하는 공격은 [3]에서 제안된 공격 아이디어에 기반을 둔다. [3]에서는 AES-128에 대한 DFA가 소개되었다. 이 공격은 다음과 같은 세 개의 단계로 구성된다. 먼저, AES-128의 라운드 8에 1개의 랜덤 바이트 오류를 주입한 후 라운드 9에서 발생하는 차분 특성을 이용하여 12개의 선형 방정식을 구성한다. 그리고 라운드 10의 라운드 키를 추측한 후 구성된 선형 방정식을 만족하는 후보 라운드 키를 계산한다. 이후 복구된 후보 라운드 키와 키스케줄 특성을 이용하여 128-비트 비밀키를 복구한다. 본 논문에서 제안하는 공격은 [3]에서 제안된 공격을 LED-64에 적용한다. 즉, 라운드 30의 입력 레지스터에 1개의 랜덤 니블 오류를 주입하여 2^8개의 후보 비밀키를 계산한다. 그리고 2^8의 전수조사를 통하여 64-비트 비밀키를 복구한다. 본 논문에서 소개하는 공격 결과는 LED-64에 대한 첫 번째 안전성 분석 결과이다.

본 논문은 다음과 같이 구성되어 있다. 먼저, 2장에서는 LED-64를 간략히 소개하고, 3장에서 [3]에서 제안된 공격 과정을 소개한다. 4장에서는 LED-64에 대한 DFA를 제안한다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. LED-64

LED-64는 64-비트 블록 암호로서, 64-비트 비밀키 K를 사용하며 전체 라운드 수는 32이다. LED-64의 64-비트 내부 상태값은 다음과 같이 16개의 니블로 이루어진 4×4 행렬로 나타낼 수 있다. 본 논문에서는 특정 내부 상태값 I의 i번째 니블 값을 $I[i]$로 표기하기로 한다 $(i=0,\cdots,15)$.

$$I=\begin{pmatrix} I[0] & I[4] & I[8] & I[12] \\ I[1] & I[5] & I[9] & I[13] \\ I[2] & I[6] & I[10] & I[14] \\ I[3] & I[7] & I[11] & I[15] \end{pmatrix}$$

그림 1은 LED-64의 암호화 과정을 나타낸 것이다. 암호화 과정에서는 크게 두 개의 함수가 사용된다. 첫 번째는 64-비트 비밀키 K가 XOR되는 addRoundKey(I, K)이다. 이 함수는 행렬 내의 4-비트 성분별로 XOR 연산이 수행된다. 두 번째는 라운드 함수가 포함되어 있는 step(I)이다. 그림 1과 같이 step 함수에서는 라운드 함수가 4번 반복 적용된다. LED-64의 암호화 과정에서는 addRoundKey와 step가 8번 반복 수행되지만, step 함수에서 라운드 함수가 4번 반복 적용된다. 따라서 LED-64의 전체 라운드 수는 32이다.

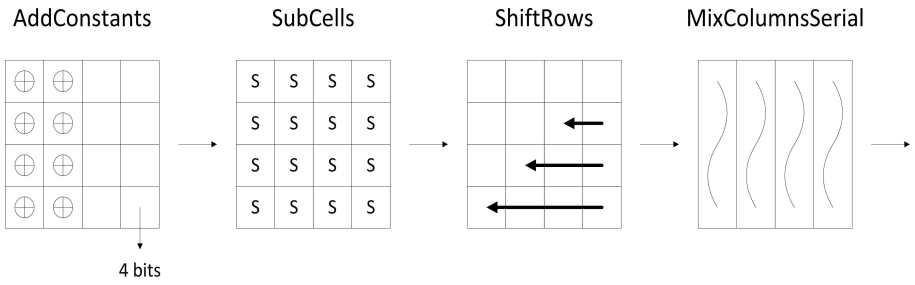

그림 2. LED-64의 라운드 함수
Fig. 2. Round function of LED-64.

LED-64의 라운드 함수는 그림 2와 같이 네 개의 함수 AddConstants, SubCells, ShiftRows, MixColumns Serial로 구성된다.

2-1 AddConstants(AC)

라운드 상수는 다음과 같이 정의된다. 먼저, 6-비트 $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$가 0으로 초기화된다. 그리고 각각의 라운드마다 1 비트씩 왼쪽으로 이동된 후 rc_0에 새로운 값 $rc_5 \oplus rc_4 \oplus 1$이 저장된다.

$$(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0) \\ \leftarrow (rc_4, rc_3, rc_2, rc_1, rc_0, rc_5 \oplus rc_4 \oplus 1)$$

AC를 수행할 시, 다음과 같은 4×4 행렬이 내부 상태값과 4-비트 단위로 XOR 연산된다.

$$\begin{pmatrix} 0 & (rc_5 \| rc_4 \| rc_3) & 0 & 0 \\ 1 & (rc_2 \| rc_1 \| rc_0) & 0 & 0 \\ 2 & (rc_5 \| rc_4 \| rc_3) & 0 & 0 \\ 3 & (rc_2 \| rc_1 \| rc_0) & 0 & 0 \end{pmatrix}$$

2-2 SubCells(SC)

SC에서는 4-비트 니블 단위로 S-box에 적용된다. 여기서 사용되는 S-box는 PRESENT에서 사용된 S-box와 동일하다.

2-3 ShiftRows(SR)

SR은 AES에서 사용된 ShiftRows와 동일하다. 즉, 첫 번째 행은 이동되지 않고, 두 번째 행에서는 1 cell씩 좌측 순환 이동된다. 세 번째 행과 네 번째 행에서는 각각 2, 3 cell씩 좌측 순환 이동된다.

2-4 MixColumnsSerial(MC)

MC에서는 열 단위로 다음과 같이 정의되는 4×4 MDS 행렬 M과의 곱셈 연산이 수행된다.

$$M = \begin{pmatrix} 4 & 2 & 1 & 1 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{pmatrix}$$

Ⅲ. AES-128에 대한 DFA

본 장에서는 [3]에서 제안된 AES-128에 대한 DFA를 간략히 소개한다. 이 공격은 라운드 8의 입력 레지스터에 1개의 랜덤 바이트 오류를 주입하여 128-비트 비밀키를 복구한다.

이 공격의 오류 주입 가정은 랜덤 바이트 오류 모델에 기반을 둔다. 하지만, 최근 AES-128의 정확한 라운드에서 정확한 위치에 오류를 주입하는 것이 가능한 것으로 알려졌다 [10]. 따라서 본 장에서는 오류가 라운드 8의 입력 레지스터 중 첫 번째 바이트 레지스터 $I_8[0]$에 오류가 주입된 경우만을 소개한다.

오류 발생 여부에 따라 평문/암호문 쌍을 다음과 같이 표기한다.

- (P, C): 오류가 발생하지 않은 알고리즘을 이용

하여 얻은 평문/암호문 쌍

- (P, C^*): 오류가 발생한 알고리즘을 이용하여 얻은 평문/암호문 쌍

오류 주입을 통해 발생한 차분이 f라고 할 때, 차분 확산 경로는 그림 3과 같다. 이를 이용하여 128-비트 비밀키를 복구하기 위해 다음과 같은 과정을 수행한다.

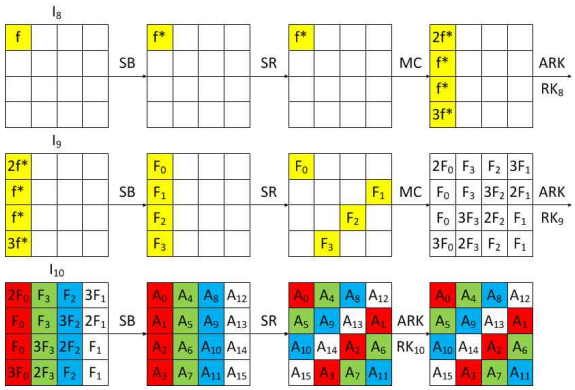

그림 3. AES-128에 대한 DFA
Fig. 3. DFA on AES-128.

먼저, 라운드 10의 32-비트 라운드 키 $RK_{10}[0,7,10,13]$을 추측한 후, 다음과 같은 방정식을 이용하여 후보 $RK_{10}[0,7,10,13]$의 수를 $2^8(=2^{32}\cdot 2^{-24})$로 줄일 수 있다.

$$2F_0 = S^{-1}(C[0]\oplus RK_{10}[0])\oplus S^{-1}(C^*[0]\oplus RK_{10}[0])$$
$$F_0 = S^{-1}(C[13]\oplus RK_{10}[13])\oplus S^{-1}(C^*[13]\oplus RK_{10}[13])$$
$$F_0 = S^{-1}(C[10]\oplus RK_{10}[10])\oplus S^{-1}(C^*[10]\oplus RK_{10}[10])$$
$$3F_0 = S^{-1}(C[7]\oplus RK_{10}[7])\oplus S^{-1}(C^*[7]\oplus RK_{10}[7])$$

위의 과정을 $RK_{10}[1,4,11,14]$, $RK_{10}[2,5,8,15]$, $RK_{10}[3,6,9,12]$에 반복 적용하여 $2^{32}(=2^{8\cdot 4})$개의 후보 RK_{10}을 얻을 수 있다.

한편, RK_{10}을 이용하여 AES-128의 키스케줄을 통해 RK_9를 계산할 수 있음을 쉽게 알 수 있다. 따라서 각각의 후보 RK_{10}으로부터 RK_9를 계산할 수 있다. 각각의 후보 (RK_9, RK_{10})으로부터, 라운드 9의 32-비트 입력 차분 $\Delta I_9[0,1,2,3]$을 계산한다. 차분 패턴 $(2f^*, f^*, f^*, 3f^*)$를 체크함으로써, 후보 (RK_9, RK_{10})의 수를 $2^8(=2^{32}\cdot 2^{-24})$로 더 줄일 수 있다. 각각의 후보 (RK_9, RK_{10})로부터 1개의 128-비트 비밀키 K를 계산할 수 있다. 따라서 이 공격은 1개의 랜덤 바이트 오류와 2^8의 전수조사를 이용하여 128-비트 비밀키를 복구할 수 있다.

Ⅳ. LED-64에 대한 DFA

본 장에서는 LED-64에 대한 차분 오류 공격을 제

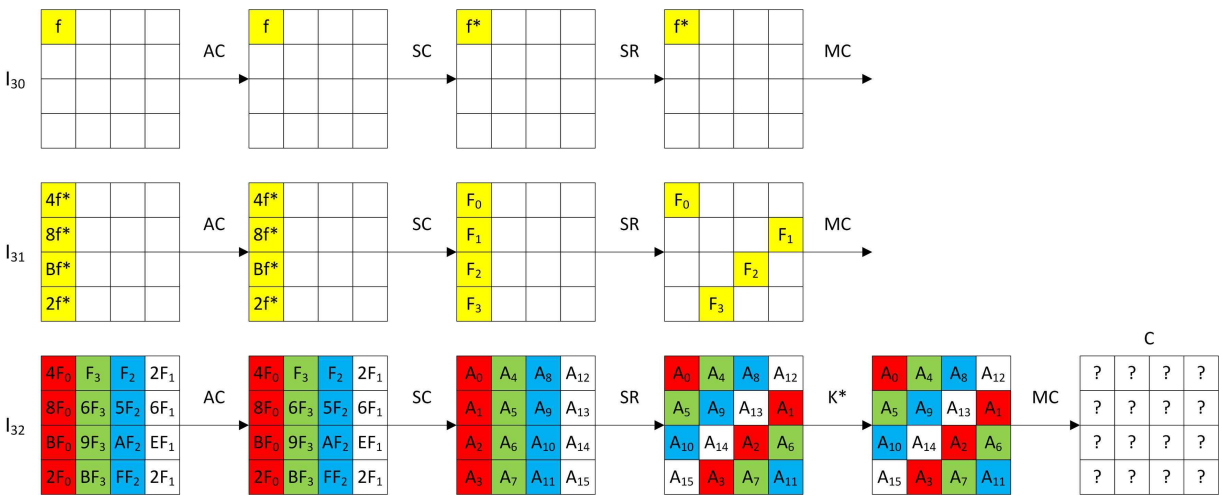

그림 4. LED-64에 대한 DFA
Fig. 4. DFA on LED-64.

안한다. 본 공격은 3장에서 소개한 공격에 기반을 두며, 라운드 30의 입력 레지스터 I_{30}에 랜덤 니블 오류를 주입하여 LED-64의 64-비트 비밀키 K를 복구한다.

라운드 30의 $I_{30}[0]$에 오류가 주입되고, 이를 통해 발생한 차분이 f일 때, 차분 확산 경로는 그림 4와 같다. 여기서 K^*는 $\mathrm{MC}^{-1}(K)$를 의미한다. 즉, K^*를 사용함으로서, 마지막 라운드에서 MC와 addRoundKey 함수의 적용 순서를 변환할 수 있다.

64-비트 비밀키 K를 복구하는 방법은 3장에서 소개한 방법과 유사하다. 먼저, (C, C^*)를 이용하여 $X = \mathrm{MC}^{-1}(C)$와 $X^* = \mathrm{MC}^{-1}(C^*)$를 각각 계산한다. $K^*[0,7,10,13]$를 추측한 후, 각각의 (X, X^*)에 대하여 $I_{32}[0,1,2,3]$을 계산한다. 그리고 다음과 같은 방정식을 이용하여 후보 $K^*[0,7,10,13]$의 수를 2^{16}개에서 $2^4 (= 2^{16} \cdot 2^{-12})$로 줄인다.

$$
\begin{aligned}
4F_0 &= AC^{-1}(S^{-1}(X[0] \oplus K^*[0])) \oplus \\
&\quad AC^{-1}(S^{-1}(X^*[0] \oplus K^*[0])) \\
8F_0 &= AC^{-1}(S^{-1}(X[13] \oplus K^*[13])) \oplus \\
&\quad AC^{-1}(S^{-1}(X^*[13] \oplus K^*[13])) \\
BF_0 &= AC^{-1}\mathrm{S}^{-1}(X[10] \oplus K^*[10])) \oplus \\
&\quad AC^{-1}\mathrm{S}^{-1}(X^*[10] \oplus K^*[10])) \\
2F_0 &= AC^{-1}\mathrm{S}^{-1}(X[7] \oplus K^*[7])) \oplus \\
&\quad AC^{-1}\mathrm{S}^{-1}(X^*[7] \oplus K^*[7]))
\end{aligned}
$$

위의 과정을 $K^*[1,4,11,14]$, $K^*[2,5,8,15]$, $K^*[3,6,9,12]$에 반복 적용하여 총 $2^{16} (= 2^{4 \cdot 4})$개의 후보 K^*를 얻을 수 있다.

위의 과정에서 얻은 2^{16}개의 후보 K^*에 대해, (C, C^*)로부터 $I_{31}[0,1,2,3]$을 각각 계산한다. 차분 패턴 $(4f^*, 8f^*, Bf^*, 2f^*)$를 체크함으로써, 후보 K^*의 수를 $2^4 (= 2^{16} \cdot 2^{-12})$로 더 줄일 수 있다.

오류가 주입될 수 있는 가능한 위치의 수는 16이므로, 후보 K^*의 수는 $2^8 (= 2^4 \cdot 2^4)$이다. 그리고 각각의 후보 K^*에 대해 $\mathrm{MC}(K^*)$을 계산하여 총 2^8개의 후보 비밀키 K를 얻을 수 있다. 따라서 본 공격은 1개의 랜덤 니블 오류 주입과 2^8의 전수조사를 이용하여 LED-64의 64-비트 비밀키 K를 복구할 수 있다.

한편, 3장에서 언급하였듯이, AES-128의 경우, 정확한 라운드에서 정확한 위치에 오류를 주입하는 것이 가능하다. LED-64의 구조는 AES-128의 구조와 유사하기 때문에, [10]의 결과가 LED-64에도 적용될 수도 있다. 만약에 그렇다면, 본 논문에서 제안하는 공격은 1개의 랜덤 니블 오류 주입과 2^4의 전수조사만을 이용하여 64-비트 비밀키 K를 복구할 수 있다.

Ⅴ. 결 론

본 논문에서는 차분 오류 공격을 이용하여 64-비트 블록 암호 LED-64에 대한 첫 번째 안전성 분석 결과를 제안하였다. 본 논문에서 소개한 공격은 1개의 랜덤 니블 오류 주입과 2^8의 전수조사를 이용하여 LED-64의 64-비트 비밀키를 복구할 수 있다.

본 논문에서 소개한 공격 결과를 통해, LED-64의 구조가 AES-128의 구조와 유사하기 때문에, AES-128에 적용되었던 공격이 LED-64에도 유사하게 적용될 수 있음을 알 수 있다. LED-64가 WSN과 같은 하드웨어 환경에 효율적으로 동작되도록 설계되었다는 점과 차분 오류 공격이 하드웨어 환경에서 적용 가능함을 고려할 때, 본 논문에서 제안한 공격 결과를 통해 LED-64가 실제 하드웨어 환경에 사용되었을 경우 안전성에 매우 취약함을 알 수 있다.

감사의 글

본 연구는 지식경제부 IT R&D 사업의 일환으로 수행하였음(유비쿼터스 환경에서의 정보보호 서비스를 위한 프라이버시 강화 암호 기술 개발)

참 고 문 헌

[1] E. Biham and A. Shamir, “Differential Cryptanalysis of DES-like Cryptosystem”, *Journal of Cryptology,* Vol. 4, No. 1, pp. 3-72, Springer-Verlag, Jan. 1991.

[2] E. Biham and A. Shamir, “Differential Fault Analysis

of Secret Key Cryptosystems", *Crypto'97, LNCS 1294*, pp. 513-525, Springer-Verlag, 1997.

[3] M. Tunstall, D. Mukhopadhyay and S. Ali, "Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault", *WISTP'11, LNCS 6633*, pp. 224-233, Springer-Verlag, 2011.

[4] 박세현, 정기태, 이유섭, 성재철, 홍석희, "블록 암호 ARIA-128에 대한 차분 오류 공격", *정보보호학회논문지,* 제 21권, 제 5호, pp. 15-25, 2011. 10.

[5] K. Jeong, Y. Lee, J. Sung and S. Hong, "Differential fault analysis on block cipher SEED", *Mathematical and Computer Modelling*, Vol. 55, Issues 1-2, pp. 26-34, Elsevier, Jan. 2012.

[6] C. Cannière, O. Dunkelman and M. Knežević, "KATAN and KTANTAN – A Family of Small and Efficient Hardware-Oriented Block Ciphers", *CHES'09, LNCS 5747*, pp. 282-288, Springer-Verlag, 2009.

[7] L. Knudsen, G. Leander, A. Poschmann and M. Robshaw, "PRINTcipher: A Block Cipher for IC-Printing", *CHES'10, LNCS 6225*, pp. 16-32, Springer-Verlag, 2010.

[8] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, "The LED Block Cipher", *CHES'11, LNCS 6917,* pp. 326-341, Springer-Verlag, 2011.

[9] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, "Piccolo: An Ultra-Lightweight Blockcipher", *CHES'11, LNCS 6917*, pp. 342-357, Springer-Verlag, 2011.

[10] T. Fukunaga and J. Takahashi, "Practical fault attack on a cryptographic LSI with IOS/IEC 18033-3 block ciphers", *FDTC'09,* pp. 84-92, IEEE, 2009.

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