情報保護學會論文誌
第18卷 第1號, 2008. 2

금융IC카드에 대한 부채널분석공격 취약성 분석

김 창 균[†‡], 박 일 환
한국전자통신연구원 부설연구소

Investigation of Side Channel Analysis Attacks on Financial IC Cards

ChangKyun Kim[†‡], IlHwan Park
The Attached Institute of ETRI

요 약

현재 국내에서는 IC카드를 이용하여 차세대 주민등록증, 금융IC카드 및 행정기관IC카드의 개발이 빠르게 진행되고 있다. 하지만 대량수요로 인한 원가절감을 위해 저가형 IC카드가 이용될 것으로 예상되며 이러한 저가형 IC카드의 경우 부채널분석공격에 매우 취약할 것으로 예측된다. 본 논문에서는 IC카드의 부채널분석공격 취약성을 조사하기 위해 현재 사용되고 있는 금융IC카드를 대상으로 차분전력분석공격을 실험해 보았다. 실험결과 100개의 소비전력파형으로도 차분전력분석공격을 성공할 수 있었으며 이를 통해 계좌비밀번호를 암호화하는데 사용되는 IC카드의 마스터키를 알아낼 수 있었다.

ABSTRACT

The development of next-generation resident registration cards, financial IC cards and administrative agency IC cards based on a smart card is currently coming out in Korea. However, the low-price IC cards without countermeasures against side channel analysis attacks are expected to be used for cost reduction. This paper has investigated the side channel resistance of financial IC cards that are currently in use and have performed DPA attacks on the financial IC cards. We have been able to perform successful DPA attacks on these cards by using only 100 power measurement traces. From our experiment results, we have been able to extract the master key used for encryption of a count PIN number.

Keywords : *부채널분석공격, DPA, 금융IC카드*

Ⅰ. 서 론

기존에 널리 사용되었던 자기카드의 보안성 문제 및 기능적 제안으로 인하여 금융권을 포함한 여러 분야에서 IC카드로의 전환이 빠르게 진행되고 있다. 최근 국내에서도 금융IC카드 표준[1] 및 행정기관 IC카드 표준[2] 등을 내놓으면서 IC카드의 사용에 발맞추고 있다. 하지만 자기카드에 비해 보안기능이 한 층 강화된 IC카드에도 여전히 보안적 취약성을 안고 있다. 그 대표적인 예가 IC카드에 대한 부채널분석공격이다.

부채널분석공격은 수학적 기반을 둔 여타 암호알고리즘 분석기술보다 매우 위협적이며 실효적인 분석기법으로써 1996년 P. Kocher가 시차분석공격[3]을 발표한 이후 다양한 분석기법이 소개되고 연구되어 왔다[4-6]. 그 중 전력분석공격[4]과 전자기분석공격[5]은 가장 강력

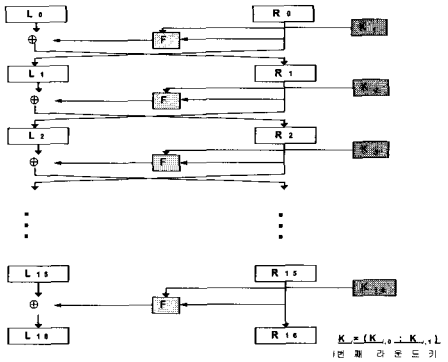
접수일: 2007년 10월 5일; 채택일: 2007년 11월 7일
† 주저자, kimck@ensec.re.kr
‡ 교신저자, kimck@ensec.re.kr

한 부채널분석공격 중 하나로 알려져 있으며 그와 관련된 많은 논문이 지금까지 발표되고 있다

IC카드와 같은 하드웨어 암호모듈의 경우 암호알고리즘이 동작하는 동안 소비되는 전력은 하드웨어 특성뿐만 아니라 암호모듈 내에서 처리되는 데이터 값에도 매우 민감하게 반응한다. 즉, 전력분석공격은 비밀키에 따른 데이터 값의 변화와 소비전력간의 상관관계를 분석하여 비밀키를 알아내는 공격방법이다. 전자기분석공격은 전자기신호를 이용하여 비밀정보를 알아내는 점을 제외하면 전력분석공격과 거의 유사하다. 국내에서는 국내표준알고리즘인 ARIA[7] 및 SEED[8]를 IC카드에 소프트웨어로 구현하여 전력분석공격 및 전자기분석공격을 실험한 사례[9-10]는 있으나 금융IC카드와 같이 실제 사용되고 있는 상용제품에 대한 시도는 보고된 사례가 없다. 하지만 대량의 자기카드에서 IC카드로 전환하고 있는 금융권의 입장에서는 카드원가 절감차원에서 저가형 IC카드를 선호하고 있기 때문에 부채널분석공격에 아무런 대책이 없는 저가형 IC카드를 사용할 경우 보안상의 문제점을 일으킬 수 있다. 또한 대부분의 저가형 IC카드의 경우 ARIA나 SEED를 위한 전용 하드웨어가 지원되지 않아 소프트웨어로 구현해야 되며 이럴 경우 암·복호 속도저하는 물론 또 다른 보안상의 문제점을 유발시킬 수 있다.

본 논문에서는 앞서 언급한 바와 같이 보안상에 다소 문제점이 있을 수 있는 금융IC카드를 대상으로 부채널분석공격 취약성에 대해 연구해 보았다. 먼저 금융IC카드의 보안알고리즘인 SEED에 대한 전력분석공격 기법을 살펴본 후 실제 금융IC카드를 이용하여 차분전력분석공격을 수행하였다. 금융IC카드의 내부명령어 중 하나인 GET ENCIPHER 명령어를 실행하여 IC카드에 저장된 마스터키가 사용되도록 유도한 후 차분전력분석공격을 적용하였다. 실험 결과 대응기법이 없는 것으로 여겨지는 금융IC카드의 경우 100개의 전력신호만으로도 카드의 마스터키를 찾을 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 SEED에 대한 전력분석공격 기법을 살펴보고, 3장에서는 금융IC카드에 대한 차분전력분석공격 적용방안을 제시하였다. 4장에서는 실제 금융IC카드에 대한 차분전력분석공격의 실험 결과를 기술하고, 마지막으로 간단한 요약과 함께 결론을 5장에서 맺었다.



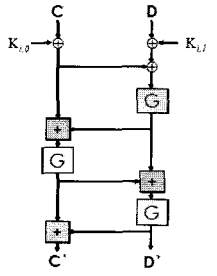
〔그림 1〕. SEED 전체 구조도

Ⅱ. SEED에 대한 차분전력분석공격

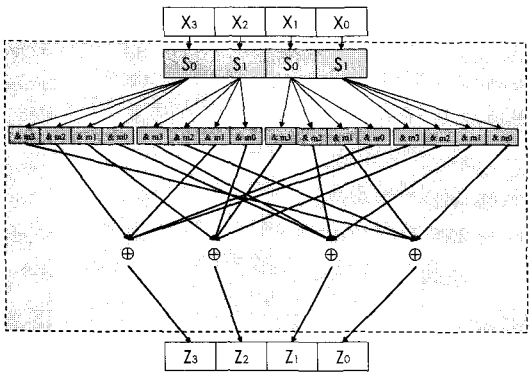
2.1. SEED 알고리즘 개요

SEED는 블록 단위로 메시지를 처리하는 대칭키 블록암호알고리즘으로서 고정된 128비트 평문을 같은 길이의 128비트 암호문으로 바꾸는 암호알고리듬이다. SEED는 Feistel 구조로 되어 있으며 128비트의 평문 블록단위당 128비트 키로부터 생성된 16개의 64비트 라운드 키를 입력으로 사용하여 총 16라운드를 거치면서 128비트 암호문 블록을 출력한다. [그림 1]은 SEED 알고리즘의 전체구조를 도식화한 것이다.

SEED에 사용되는 F함수는 수정된 64비트 Feistel 형태로 구성된다. F함수는 각 32비트 블록 2개(C, D)를 입력으로 받아, 32비트 블록 2개(C', D')를 출력한다. 즉, 64비트 블록(C, D)과 64비트 라운드 키 $K_i = (K_{i,0} ; K_{i,1})$ 를 F함수의 입력으로 받아 64비트 블록(C', D')을 출력한다. [그림 2]는 i번째 라운드의 F함수의 구조를 나타낸 것이다.



〔그림 2〕. F함수의 구조



〔그림 3〕. G함수

F함수의 핵심이라 할 수 있는 G함수는 다음과 같은 연산을 수행한다([그림 3] 참조). G함수에 사용된 두 종류의 S-box는 8비트 입력을 받아 8비트 출력을 내는 함수로서 비선형성을 제공하는 중요한 역할을 수행한다. G함수에 입력된 32비트를 $X_3\|X_2\|X_1\|X_0$라하고 $S_0()$과 $S_1()$을 각각 S-box에 의한 대치라고 하면 G함수의 32비트 출력 $Z_3\|Z_2\|Z_1\|Z_0$는 다음과 같이 계산된다.

$$Y_3 = S_0(X_3), Y_2 = S_1(X_2),$$

$$Y_1 = S_0(X_1), Y_0 = S_1(X_0),$$

$$\begin{aligned} Z_3 &= (Y_0 \& m_3) \oplus (Y_1 \& m_0) \oplus (Y_2 \& m_1) \oplus (Y_3 \& m_2) \\ Z_2 &= (Y_0 \& m_2) \oplus (Y_1 \& m_3) \oplus (Y_2 \& m_0) \oplus (Y_3 \& m_1) \\ Z_1 &= (Y_0 \& m_1) \oplus (Y_1 \& m_2) \oplus (Y_2 \& m_3) \oplus (Y_3 \& m_0) \\ Z_0 &= (Y_0 \& m_0) \oplus (Y_1 \& m_1) \oplus (Y_2 \& m_2) \oplus (Y_3 \& m_3) \\ & (m_0 = 0\text{xfc}, m_1 = 0\text{xf3}, m_2 = 0\text{xcf}, m_3 = 0\text{x3f}) \end{aligned}$$

2.2. SEED의 S-box에 대한 차분전력분석공격

2.2.1. 소비전력 모델링

차분전력분석공격은 소비되는 전력과 비밀키에 따라 처리되는 데이터간의 상관관계를 통하여 비밀키를 찾는 방법으로써 통상 차분전력분석공격이 단순전력분석공격보다 더욱 강력하다고 알려져 있다. 차분전력분석공격은 처리되는 데이터 값에 따른 소비전력의 미세한 변화를 이용하기 때문에 소비전력의 특성을 잘 이해해야 한다. 통상 소비전력의 모델링은 해밍수모델(Hamming weight model)과 해밍차모델(Hamming distance model)로 구분된다. 해밍수모델은 처리되는 데이터 중 '1'이 '0'보다 많은 전력을 소비한다는 사실에 기반을 두고 있으며 해밍차모델은 데이터의 상태천이가 소비전력에 영향을 미치는 사실에 기반을 둔다. 자세히 말하면 '1'에서 '0'으로 '0'에서 '1'로 상태가 변할 때 전력소모가 일어나고 상태가 변하지 않을 때는 전력소모가 거의 일어나지 않음을 뜻한다. 본 논문에서는 해밍수모델을 가정하여 기술하였으며 실험대상이 되는 IC카드 역시 해밍수모델을 따른다고 가정하였다.

2.2.2. S-box 출력에 대한 차분전력분석공격 방안

차분전력분석공격에는 데이터와 소비전력사이의 상관관계를 판단하기 위해 여러 가지 기법들이 사용된다. 대표적인 방법으로는 평균차(diffrence of means) 방법, 평균거리(distance of means), 최대우도(maximum likelihood) 방법, 상관도(correlation) 방법 등이 있다. 본 논문에서는 위 4가지 방법 중 가장 효율적이라고 알려진 상관도 방법을 이용하였다.

상관도 방법은 평균차 방법과는 달리 두 그룹으로 전력파형을 분류하지 않고 해밍수와 전력파형간의 상관도만을 계산한다. 따라서 수집한 전력파형을 모두 이용할 수 있으며 다중비트 공격 시 모든 비트를 고려할 수 있는 장점이 있다. 다음은 S-box 출력에 대해 상관도 방법을 이용한 차분전력분석공격을 단계별로 기술하였다.

단계 1 : 동일한 키를 사용하여 S개의 서로 다른 입력 데이터를 암호화하는 동안 소비전력을 측정한다. 이 때, 측정된 소비전력파형을 $P_{1\cdots S,1\cdots T}$로 표기한다. 여기서 T는 오실로스코프에 측정된 샘플 수를 뜻한다.

단계 2 : 입력 평문에 대해 0부터 255까지 8비트 킷값을 고려하여 S-box 출력을 계산한다. S-box 출력의 해밍수를 고려하여 행렬 $H_{0\cdots 255,1\cdots S}$를 계산한다.

단계 3 : 수집한 소비전력신호 P와 S-box 출력 H간 상관계수는 아래 수식을 이용하여 계산한다.

$$C(P,H) = \frac{E(P \cdot H) - E(P) \cdot E(H)}{\sqrt{Var(P) \cdot Var(H)}}$$

단계 4 : 단계 3으로부터 얻은 상관계수 $C_{0\cdots 255,1\cdots T}$에서 올바른 키 $k \in \{0\cdots 255\}$에 대한 상관계

수를 $C_{k,1\cdots T}$라 하자. 만약 S-box 출력에 의한 소비전력이 $1 \le j \le T$ 샘플에서 발생했다고 가정하면 입력 데이터 S가 클수록 상관계수는 $0 < C_{k,j}(P,H) \neq 0 < 1$ 가 되며 틀린 키를 포함한 나머지 샘플구간에서는 거의 '0' 으로 수렴한다. 즉, 상관계수가 높은 것이 찾고자 하는 부분키이다.

위 방법을 SEED 알고리즘에 적용하면 다음과 같다. 먼저 차분전력분석공격은 SEED의 첫 번째 라운드에서 수행된다고 가정한다. F함수의 각 입력 32비트를 $C = C_3 \| C_2 \| C_1 \| C_0$, $D = D_3 \| D_2 \| D_1 \| D_0$, 왼쪽 라운드 키를 $K_{1,0} = K_{1,30} \| K_{1,20} \| K_{1,10} \| K_{1,00}$ 오른쪽 라운드 키를 $K_{1,1} = K_{1,13} \| K_{1,12} \| K_{1,11} \| K_{1,10}$라 하자. 그러면 G함수 내부의 첫 번째 S-box 연산은 다음과 같다.

$$S_0(X_3) = S_0((C_3 \oplus D_3) \oplus (K_{1,03} \oplus K_{1,13})) \quad (1)$$

비록 $K_{1,30}$와 $K_{1,31}$을 동시에 고려해야 되지만 $k = K_{1,30} \oplus K_{1,31}$라 두면 위에서 기술한 상관도 방법을 그대로 적용할 수 있다. 결국 공격자는 $K_{1,30} \oplus K_{1,31}$을 찾을 수 있으며 나머지 바이트에 대해서도 동일하게 찾을 수 있기 때문에 G함수 출력 32비트를 알아낼 수 있다.

$K_{1,30}$와 $K_{1,31}$을 얻기 위해서 F함수 내의 두 번째 G함수에서 차분전력분석공격을 한 번 더 수행하면 된다. 첫 번째 G함수 출력을 $G_1(C \oplus D \oplus K_{1,0} \oplus K_{1,1})$라 두고 두 번째 G함수 입력을 B라 두면 B는 다음과 같다.

$$\begin{aligned} B &= B_3 \| B_2 \| B_1 \| B_0 \\ &= (C \oplus K_{1,0}) + G_1(C \oplus D \oplus K_{1,0} \oplus K_{1,1}) \bmod 2^{32} \end{aligned} \quad (2)$$

이 때 B는 덧셈연산에 의해 결정되기 때문에 캐리가 발생된다. 따라서 최하위 바이트 B_0부터 차분전력분석공격을 수행하여 $K_{1,00}$을 알아낸 후 다음 상위 바이트를 찾아내야 한다. 결국 공격자는 32비트 $K_{1,0}$을 찾은 후 이로부터 32비트 $K_{1,1}$을 찾을 수 있다.

Ⅲ. 금융IC카드에 대한 차분전력분석공격

금융IC카드는 K-CASH와 현금 IC카드로 나눠볼 수 있다. 그 중 K-CASH는 금융결제원과 전 국내은행이 참여하여 개발한 한국형 전자화폐로서 현금의 가치저장 및 지급수단으로 사용할 수 있으며 그 외 공인인증서, ID 카드 등 다양한 부가기능을 가지고 있다. K-CASH와 현금 IC카드는 기존 금융망에서 사용되었던 3-DES를 사용하지 않고 128비트 대칭키 암호알고리즘인 SEED를 채택하여 시스템의 보안성 및 신뢰성을 제공하고 있다.

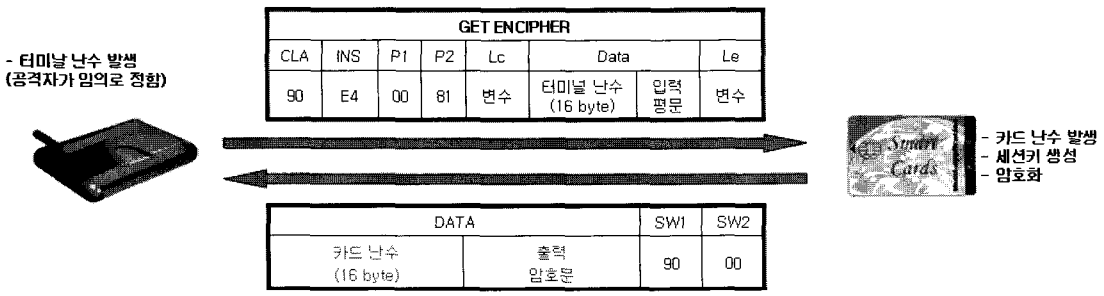
본 장에서는 금융IC카드를 대상으로 차분전력분석공격을 수행하기 위한 방안을 제시한다. 먼저 차분전력분석공격을 수행하기 위해서는 동일한 키에 대해 수많은 평문을 입력할 수 있어야 한다. 하지만 금융IC카드의 경우 매번 갱신되는 세션키를 이용하여 암호화과정을 수행하기 때문에 차분전력분석공격을 그대로 적용하기 어렵다. 위와 같은 문제를 해결하기 위해 세션키가 사용되는 암호화과정에서 공격을 수행하는 것이 아니라 마스터키가 이용되는 세션키 생성과정에서 차분전력분석공격을 수행한다. 먼저 세션키 생성과정과 관련된 'GET ENCIPHER' 명령어에 대해 알아본 후 이를 이용한 차분전력분석공격 방법에 대해 살펴본다.

3.1. 세션키 생성과정 및 GET ENCIPHER 명령어

GET ENCIPHER 명령어는 금융IC카드에서 제공되는 APDU 명령어로서 내부 키파일에 저장되어 있는 키를 이용하여 임시 세션을 형성한 후, 외부에서 들어온 데이터를 암호화하여 출력해 준다. 현금IC카드의 경우 사용자가 계좌비밀번호를 입력하면 IC카드는 GET ENCIPHER 명령어에 의해 계좌비밀번호를 암호화하여 해당은행에 전송한다. 결국 IC카드와 동일한 마스터키를 보관하고 있는 해당은행은 암호화 메시지를 복호하여 계좌비밀번호를 인증한다. [표 1]은 터미널에서 카드로 보내는 GET ENCIPHER 명령어 메시지이며 [표 2]는 응답 메시지를 나타낸 것이다.

[표 1]. GET ENCIPHER Command APDU

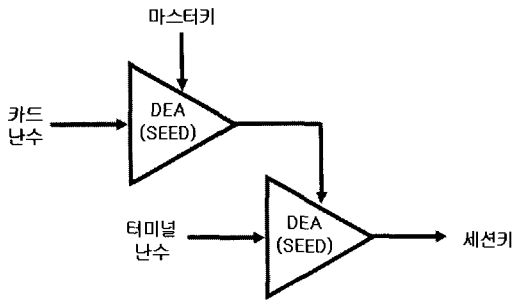
코 드	내 용
CLA	90
INS	E4
P1	00
P2	0A
Lc	Variable
Data	Terminal Random(16 byte) + 암호입력데이터
Le	Variable



〔그림 5〕. GET ENCIPHER 명령어 수행과정

〔표 2〕. GET ENCIPHER Response APDU

코 드	내 용	길이
DATA	Card Random(16 byte) + 암호출력데이터	Var.
SW1, SW2	COMPLETION CODE	2



〔그림 4〕. 세션키 생성과정

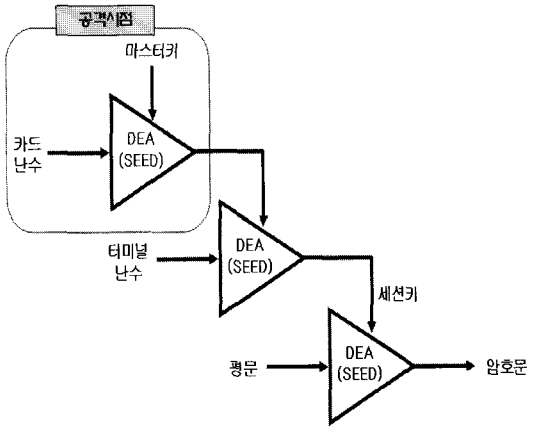
[표 2]에서 암호화 과정이 성공하면 'SW1 SW2 = 90 00'을 내보내고 오류가 발생하면 각 상황에 맞는 오류 코드를 보낸다.

[그림 4]는 GET ENCIPHER 명령어에서 세션키 생성을 설명한 그림이다.

세션키 생성과정을 보면 카드의 난수, 터미널의 난수, 카드의 마스터키를 입력으로 하고 DEA(Data Encryption Algorithm)에 해당하는 SEED를 이용해 세션키를 만든다. 여기서, 각각의 난수는 매번 갱신된다.

3.2. SEED의 S-box에 대한 차분전력분석공격

전술한 바와 같이 차분전력분석공격을 수행하기 위



〔그림 6〕. 공격시점을 도시한 암호화과정

해서는 동일한 키에 대해 수많은 평문을 입력할 수 있어야 한다. [그림 4]의 세션키 생성과정에서 마스터키를 고정된 키로 보면 카드 난수를 평문으로 간주할 수 있다. 비록 카드 난수는 공격자가 의도한 값은 아니지만 최종 암호문과 함께 카드 난수가 터미널로 보내지기 때문에 공격자는 카드 난수를 알 수 있다. [그림 5]는 지금까지 설명한 GET ENCIPHER의 동작순서를 한눈에 볼 수 있도록 그려 놓았다.

[그림 5]에서 공격자가 정의할 수 있는 값은 파란색으로 표시하였으며 공격자와 상관없이 정해지는 값은 빨간색으로 표시하였다. [그림 6]에서 표시한데로 실제 차분전력분석공격이 수행되는 시점은 GET ENCIPHER의 세션키 생성과정 중 마스터키를 이용해 카드난수를 암호화하는 과정이다.

결국 공격자는 수많은 GET ENCIPHER 명령어를 실행하여 차분전력분석공격을 수행할 수 있다.

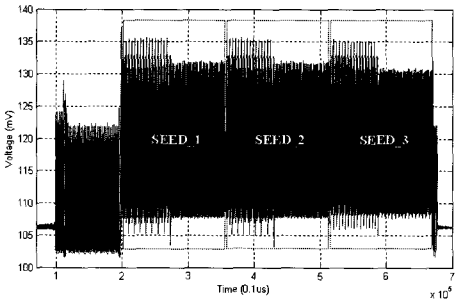
Ⅳ. 실험 결과

4.1. 암호알고리즘 연산에 따른 소비전력특성 분석

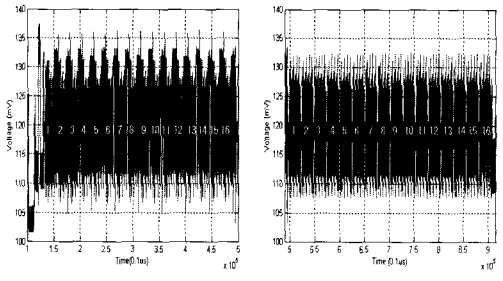
차분전력분석공격을 수행하기에 앞서 본 절에서는 금융IC카드의 암호알고리즘 동작 시 소모되는 전력을 분석하였다. 실험에 사용된 금융IC카드는 접촉식 방식만을 지원하는 폐쇄형 IC카드로서 삼성 IC카드 컨트롤러가 탑재된 8비트 IC카드이다. IC카드 내 SEED 알고리즘은 소프트웨어로 구현되어 있다.

[그림 7]은 GET ENCIPHER 명령어를 500번 수행하여 수집한 파형을 평균하여 잡음을 제거한 평균파형이다. 16바이트 평문에 대한 GET ENCIPHER 명령어 수행 시 3번의 SEED 알고리즘이 수행되는데 이는 [그림 7]에서도 확인할 수 있다([그림 7]에서 3번의 규칙적인 소비전력패턴을 볼 수 있음).

[그림 8]은 첫 번째 SEED 알고리즘 수행에 대한 소비전력파형(SEED_1)을 확대한 것이다. 두 그림 모두에서 16번의 동일패턴이 보이며 이는 라운드키 생성과정과 암호화과정으로 간주될 수 있다. 각각의 수행시간을 살펴보면 라운드키 생성과정이 대략 37ms 소요되며 암호화과정이 대략 40ms 소요된다. 따라서 패딩이 필요



[그림 7]. GET ENCIPHER 수행 시 소비전력파형



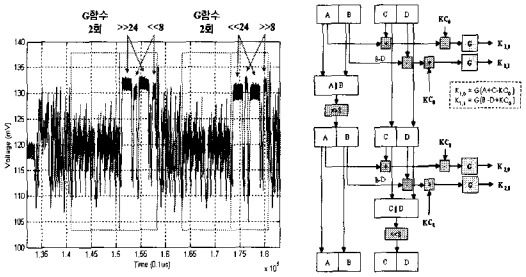
[그림 8]. 라운드키 생성과정(왼쪽)과 암호화 과정(오른쪽)

없는 128비트 평문일 경우 약 77ms의 시간이 소요됨을 알 수 있다.

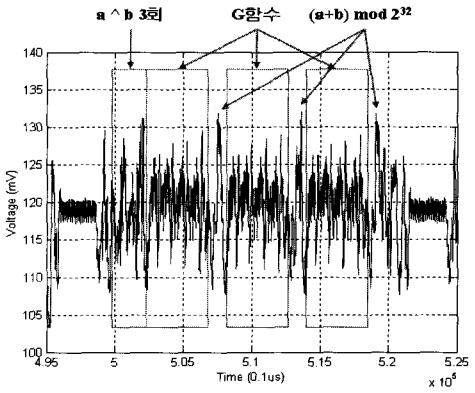
[그림 9]의 왼쪽 그림은 라운드키 생성과정 중 첫 번째와 두 번째 라운드키 계산 시 소모되는 전력파형이다. [그림 9]의 오른쪽에 있는 라운드키 생성과정을 살펴보면 한 라운드에 두 번의 G함수 계산, 4번의 모듈러 덧셈/뺄셈 연산, 쉬프트 연산이 필요하다. 소비전력파형을 자세히 보면 두 번의 일정한 패턴(14.2 ~ 15.0ms)을 가진 것이 G함수에 의한 것임을 알 수 있으며 G함수 이전에 보이는 파형(13.7 ~ 14.2ms)은 모듈러 연산에 의한 파형임을 추측할 수 있다.

쉬프트 연산은 32비트 레지스터 2개를 붙인 64비트 연산이기 때문에 '>>8'을 프로그램으로 구현할 경우 2번의 '<<24'와 2번의 '>>8'이 필요하다. 이와 유사하게 '<<8' 역시 2번의 '<<8'과 2번의 '>>24'가 필요하다. 따라서 쉬프트 연산에 대한 전력소비도 구분될 수 있다. 실험에서는 '>>'이 '<<'보다 상대적으로 많은 전력을 소비하였다.

[그림 10]은 암호화 과정 중 한 라운드만 살펴보기



[그림 9]. 라운드키 생성 시 소비전력파형(왼쪽)과 라운드키 생성블록도(오른쪽)



[그림 10]. 1라운드 암호화 과정 시 소비전력파형

위해 [그림 8]에서 49.5 ~ 52.5ms 구간을 확대한 것이다. 먼저 손쉽게 G함수에 대한 소비전력파형을 찾을 수 있었다. 그 이유는 [그림 9]에서 나타난 G함수에 대한 소비전력파형을 [그림 10]에서도 볼 수 있기 때문이다. 또한 G함수에 의한 소비전력파형 사이에 있는 파형은 모듈러 연산임을 유추해 볼 수 있으며 약 50.0 ~ 50.2ms에 나타난 파형이 3번의 XOR임을 추측할 수 있다.

4.2. 차분전력분석공격 결과

본 절에서는 앞 절에서 분석한 내용을 바탕으로 SEED의 S-box 출력에 대한 차분전력분석공격을 수행하였다. 공격을 수행하기에 앞서 수집한 소비전력신호는 동기화 되어있지 않은 신호이므로 신호정렬을 위한 정렬기법을 구현하였다.

4.2.1. 전력신호정렬기법

본 절에서는 실험에 사용된 pearson correlation을 이용한 신호정렬기법을 소개하고 실험 결과를 분석하였다. 임의의 N 샘플 데이터를 가지는 두 신호 $T_1[t], T_2[t]$에 대한 pearson correlation은 다음과 같이 정의할 수 있다.

$$\rho(T_1, T_2) = \frac{E(T_1 \cdot T_2) - E(T_1) \cdot E(T_2)}{\sqrt{Var(T_1) \cdot Var(T_2)}} \quad (3)$$

만약 T_2 신호가 T_1에 비해 δ만큼 샘플링 데이터가 어긋나 있는 경우를 고려해 보자. 이 경우 $\rho(T_1[t], T_2[t])$에 비해 $\rho(T_1[t], T_2[t-\delta])$가 더 높은 상관계수를 가질 것이다. 실제 신호정렬을 하기 위해서는 수많은 신호에 모두 적용해야 하기 때문에 모든 샘플구간에 적용하기에는 시간적인 측면에서 매우 비효율적이다. 따라서 적당한 샘플링 구간을 정하여 모든 신호에 적용해야 한다.

[그림 11]은 임의의 두 신호 T_1, T_2에 대해 효율적으로 δ 값을 찾는 상관도 기반의 신호정렬 알고리즘이다.

통상 100만개의 샘플링 데이터를 가지는 두 신호에 대해 상관도를 계산할 경우 짧은 시간이 소요되지만 1,000개 이상의 신호에 대해 모두 적용하면 상당히 많은 시간이 소요된다. 따라서 적절한 윈도우 크기를 잡아서 상관도를 계산해야 한다. 또한 모든 샘플링 구간에

```
Input : T1[t], T2[t], w(window), s(search range)
Output : δ
1. comp = 0
1. for( i=−s, i<s, i++)
2.     temp = ρ( T1[t], T2[t−i])
3.     if (temp > comp)
4.         comp = temp
5.         δ = i
6. return( δ)
```

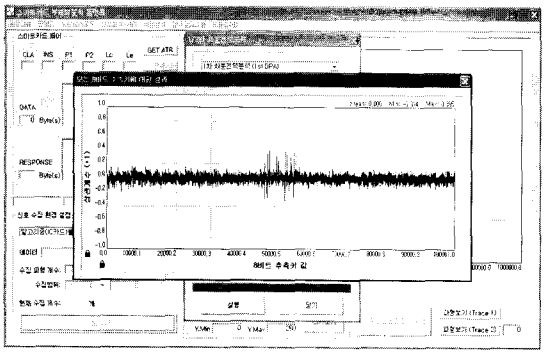
[그림 11]. δ 값을 찾기 위한 상관도 기반 신호정렬 알고리즘

걸쳐 윈도우를 옮겨가며 최적의 δ값을 찾는 것은 매우 비효율적이므로 윈도우가 움직이는 범위(search range)도 임의로 정해야 한다. T_1 신호를 기준 신호로 잡은 뒤 500개의 신호에 대해 $w=2000$, $s=3000$로 설정한 후 PC(Pentium 3GHz CPU, RAM 2GByte)를 이용하여 알고리즘 수행한 결과, 1분 이내에 모든 δ값을 찾을 수 있었다. [그림 12]는 10개의 데이터에 대해서 정렬 알고리즘 수행 후 얻은 δ과 그에 대한 ρ이다.

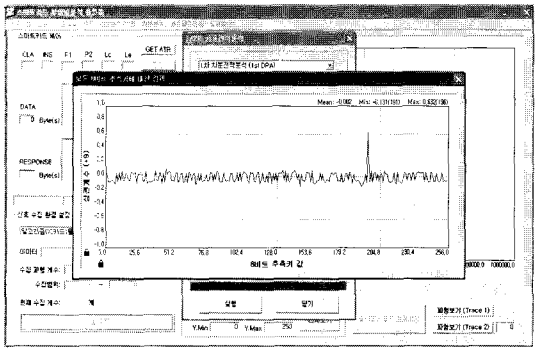
[그림 12](a)에서 4, 8, 9번 신호의 상관계수가 타 신호에 비해 현저하게 낮은 이유는 δ가 탐색범위에 있지 않기 때문이다. 따라서 탐색범위를 3000에서 4000으로 넓히면 [그림 12](b)와 같이 4, 8, 9번 모두 0.99이상의 상관계수를 가지게 되어 완벽하게 정렬된 신호를 얻을 수 있다. 하지만 탐색범위를 넓일 경우 신호정렬에 많은

$w=2000$, $s=3000$			$w=2000$, $s=4000$		
T_i	δ	ρ	T_i	δ	ρ
1	0	1.000000	1	0	1.000000
2	82	0.996221	2	82	0.996221
3	-998	0.998113	3	-998	0.998113
4	-3000	0.805903	4	-3005	0.998127
5	-893	0.997436	5	-893	0.997436
6	-933	0.997831	6	-933	0.997831
7	501	0.996999	7	501	0.996999
8	2746	0.802184	8	-3679	0.996447
9	2529	0.787591	9	-3897	0.994105
10	340	0.995731	10	340	0.995731
(a)			(b)		

[그림 12]. 신호정렬알고리즘 수행 후 s에 따른 ρ, δ



〔그림 13〕. 8비트 키 추측 시 차분전력분석공격 결과



〔그림 14〕. 모든 키(256가지)에 대한 차분전력분석공격 결과

시간이 소요될 수 있으므로 적당한 범위 안에서 실행한 후 낮은 상관계수를 가지는 신호는 제외시키는 방법을 택함으로써 시간적 효율을 높을 수 있다.

4.2.2. S-box에 대한 차분전력분석공격 결과

IC카드에 대한 차분전력분석공격을 수행하기 위해 Visual C++ 6.0을 이용하여 부채널분석 전용 소프트웨어를 개발하였다. 개발한 소프트웨어는 크게 두 부분으로 나눌 수 있는데 하나는 계측장비를 제어하는 부분이며 나머지 하나는 수집한 신호의 분석 및 처리를 담당하는 부분이다.

[그림 13]에서 보이는 파형은 올바른 부분키(8비트) 추측 시 차분전력분석공격 결과이다. 그림에서와 같이 S-box 출력이 계산되는 시점으로 추정되는 부분에서 높은 상관계수를 가지고 있으며 이는 추측한 키가 실제 킷값과 동일함을 뜻한다. 비록 본 실험에서는 상관계수 결과가 뚜렷하게 구분되도록 500개의 정렬화 된 수집 파형을 사용하였지만 100개의 파형으로도 올바른 키를 찾을 수 있었다.

[그림 14]는 8비트 라운드키에 대한 256가지 모든 경우를 고려한 차분전력분석공격 결과이다. 잘못된 킷값 255경우에 비해 올바른 킷값(196)에서 가장 높은 상관계수가 관측됨을 볼 수 있다. 따라서 공격자는 손쉽게 8비트 라운드키를 찾을 수 있으며 나머지 120비트는 위 과정을 반복하여 모두 찾을 수 있다.

V. 결 론

본 논문에서는 국내표준 알고리즘인 SEED에 대한 차분전력분석공격에 대해 설명한 후 SEED를 보안알고리즘으로 사용하는 금융IC카드를 대상으로 차분전력분석공격을 수행하였다. 실험 결과, 부채널분석공격 대응기법이 구현되지 않은 카드로 생각되는 금융IC카드의 경우 차분전력분석공격에 매우 취약한 것으로 나타났으며 이로 인해 계좌비밀번호를 암호화하는데 이용되는 마스터키가 공격에 의해 노출됨을 확인할 수 있었다. 현재 금융권에서는 자기카드에서 IC카드로 빠르게 전환되고 있으나 비용측면에서 저가형 IC카드를 선호하고 있다. 결국 저가형 IC카드를 사용할 경우 부채널분석공격 대응기법이 마련되지 않아서 공격에 취약할 수 있기 때문에 금융거래에 보안상의 문제점을 유발시킬 수 있다. 따라서 금융IC카드에 대한 표준 및 품질인증 지침에 부채널분석공격 안전성에 대한 평가검증 요구항목을 추가하는 등 제도적인 뒷받침이 마련되어야 할 것으로 판단된다. 아울러 앞으로 진행될 행정기관IC카드 및 차세대전자주민등록증에도 금융IC카드와 같이 부채널분석공격에 대한 대비책을 마련해야할 것이다.

참고문헌

[1] 금융결제원, "금융IC카드표준," 2005년 1월.

[2] 행정자치부, "행정기관 IC카드 표준규격," 2005년 6월.

[3] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," CRYPTO'96, LNCS 1109, pp. 104-113, Springer-Verlag, 1996.

[4] P. Kocher, J. Jaffe and B.Jun, "Differential Power Analysis," CRYPTO'99, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.

[5] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," CHES'01, LNCS 2162, pp. 251-261, Springer-Verlag, 2001.

[6] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," CRYPTO'97, LNCS 1294, pp. 513-525, Springer-Verlag, 1997.

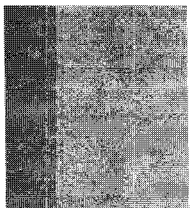
[7] D. Kwon, J. Kim, S. Park, S. Sung, Y. Sohn, J. Song, Y. Yeom, E. Yoon, S. Lee, J.Lee, S. Chee, D. Han and J. Hong, "New Block Cipher : ARIA," ICISC'03, LNCS 2971, pp. 432-445, Springer-Verlag, 2003.

[8] 한국정보통신기술협회, TTAS.KO-12.0004 : 128비트 블록암호알고리즘 표준, 1999.

[9] J. Ha, C. Kim, S. Moon, I. Park, and H. Yoo, "Differential Power Analysis on Block Cipher ARIA," HPCC'05, LNCS 3726, pp. 541-548, Springer-Verlag, 2005.

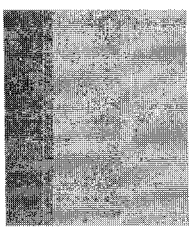
[10] H. Yoo, C. Herbst, S. Mangard, E. Oswald, and S. Moon, "Investigations of Power Analysis Attacks and Countermeasures for ARIA," WISA'06, LNCS 4298, pp. 160-172, Springer-Verlag, 2007.

〈著者紹介〉



김 창 균 (ChangKyun Kim) 정회원

2001년 2월 : 경북대학교 전자전기공학부 졸업
2003년 2월 : 경북대학교 전자공학과 석사
2003년 3월~현재 : 경북대학교 전자공학과 박사과정
2004년 11월~현재 : 한국전자통신연구원 부설연구소
<관심분야> 부채널분석, 스마트카드보안, 암호알고리즘 구현



박 일 환 (IlHwan Park) 정회원

1988년 2월 : 고려대학교 수학과 졸업
1990년 2월 : 고려대학교 수학과 석사
1996년 2월 : 고려대학교 수학과 박사
1996년 5월~1999년 12월 : 한국전자통신연구원
2000년 1월~현재 : 한국전자통신연구원 부설연구소
<관심분야> 정보보호이론