情 報 保 護 學 會 誌
第 15 卷 第 2 號, 2005. 4

스마트카드의 보안성에 관한 연구

조 은 성*, 원 동 규*, 양 형 규**, 김 승 주*** , 원 동 호****

요 약

기존의 신용카드 등이 최근에는 암호 기능을 갖춘 스마트카드로 대체되고 있다. 그러나 스마트카드의 제한적인 연산기능으로 인하여 탑재되는 암호 알고리즘을 고속화하여 탑재해야 하는데, 이렇게 고속화된 암호 알고리즘은 사이드 채널 공격(Side Channel analysis)에 취약점을 갖는다. 암호 알고리즘의 동작 중에 시간차, 전자파, 전력 등 부가적으로 얻어지는 정보를 분석하는 사이드 채널 공격은 이론적으로 안전성이 증명된 알고리즘에서도 구현상의 문제로 인하여 공격이 가능하기 때문에 그 위험성이 매우 높다. 본 고에서는 2003년 신규 정보보호제품 평가대상으로 확대된 스마트카드의 안전성 평가방안에 대하여 설명하고 스마트카드 상에서 공격 가능한 사이드 채널 공격을 타이밍 공격, 오류삽입 공격, 단순/차분 전력분석 공격으로 나누어 기술하고 이러한 공격에 대한 대응기법을 소개한다.

1. 서 론

스마트카드에서의 보안 특징들은 암호 기능들에 의해 더욱 강화될 수 있다. 카드 내에 저장된 데이터는 물리적 메모리의 프라이버시를 보호하기 위해 암호화될 수 있으며, 또한 외부와 교환되는 데이터의 안전을 위해 서명하거나 암호화할 수 있다. 이러한 스마트카드의 장점으로 인하여 신분확인, 접근통제에서부터 신용카드를 대신하는 안전한 결제수단으로 광범위하게 사용될 수 있다.

그러나, 스마트카드에 내장된 IC(Integrated Circuit) 칩의 제한적인 성능으로 인해 기존의 컴퓨터 환경에서 구현된 알고리즘을 탑재하는 것이 어렵기 때문에 CRT(중국인의 잉여정리) 기반의 RSA 서명 등 그 안전성에는 변화가 없으면서도 고속화된 알고리즘 구현에 대한 연구가 활발히 진행되고 있다. 하지만 이렇게 고속화된 알고리즘은 암호기의 동작 동안에 발생하는 시간차, 전자파, 전력, 동작에러 등을 분석하여 키를 추출해 내는 사이드 채널 공격(Side Channel Analysis) 등에 취약하다. 사이드 채널 공격은 크게 타이밍, 오류삽입, 전력분석 공격으로 분류할 수 있다.

타이밍 공격은 1995년 kocher[1]에 의해 소개된 공격방법으로 암호 알고리즘이 동작하는 동안 각 연산에 대한 수행시간을 측정하고 이를 통계적인 방법으로 분석하여 비밀키에 대한 정보를 추출하는 공격방식이다.

오류삽입 공격은 Biham & Shamir[3]에 의해 제안된 공격방식으로 공격자가 스마트카드에 제한 된 범위 이상의 전압, 온도, 전파 등을 이용하여 시스템 오류를 발생시켜 정규 값 이외의 정보를 얻어낸 후 이를 분석하여 키의 정보를 추론하는 공격방식이다.

전력분석 공격 역시 Kocher, Jaff and Jun[2]에 의해 제안된 방식으로 마이크로컨트롤러가 비밀키를 포함한 연산을 수행할 때 순간적으로 변화하는 전력소비의 차를 수집하여 통계적으로 분석하는 공격이다.

1990년대 중반까지 암호학자들은 이러한 공격들에 대한 위험성을 고려하지 않았다. 그러나 수학적으로 안전성이 증명된 알고리즘이라 하더라도 그 구현방식이나 동작과정의 내구성에 따라 취약점을 내포하고 있

* 성균관대학교 컴퓨터공학과 ({escho, dkwon}@dosan.skku.ac.kr)
** 강남대학교 컴퓨터미디어공학부 부교수 (hkyang@kangnam.ac.kr)
*** 성균관대학교 정보통신공학부 조교수 (skim@ece.skku.ac.kr)
**** 성균관대학교 정보통신공학부 교수 (dhwon@dosan.skku.ac.kr)

다는 것이 알려지면서 이에 대한 연구가 활발히 진행되고 있다.

본 논문은 총 4장으로 구성되어 있으며, 2장에서는 스마트카드 제품의 안전성 평가를 위한 평가방안에 대하여 기술하고, 3장에서는 스마트카드에 적용 가능한 사이드 채널 공격에 대하여 기술한다. 마지막으로 4장에서는 결론을 맺는다.

II. 스마트카드 보안성 평가

2.1 정보보호제품 평가 확대

2003년 11월27일 정보통신부고시 제2003-52호"정보보호시스템 공통평가기준 중 개정"에서 정보보호제품평가 대상이 확대되었다.[8] 제2조에 명시된 "가상사설망"을 "가상사설망, 지문인식시스템, 운영체제보안시스템, 스마트카드"까지 포함하여 확대하고, 제6조에 스마트카드를 정의하였다. 실제 스마트카드에 대한 평가는 2004년 말경부터 시행 예정에 있다.

2.2 스마트카드 평가방안

스마트카드는 기본적으로 암호 기능을 탑재하고 있어 현재 신용카드를 위시하여 신분확인을 위한 목적으로 광범위하게 사용될 것이 예상된다. 스마트카드는 사용자의 비밀키와 같은 중요정보가 포함되어 있기 때문에 여러 가지 보안요구사항을 만족하여야한다.

스마트카드는 IC 칩과 이를 운용하기 위한 COS (Card Operating System)를 기본적으로 탑재하고 있으며, 여러 가지 서비스를 지원하기 위한 Application이 제공된다. 따라서 스마트카드의 보안성 평가 항목은 이 세가지를 기준으로 다양한 범위에서 개발되고 있다. [표 1]은 국외 스마트카드 관련 보호프로파일 현황을 정리한 것이다.

NIAP(NIST & NSA)이 후원하는 스마트카드 보안 사용자 그룹(SCSUG)에서 개발된 SCSUG-PP와 유럽에서 개발된 EURO-SMART PP, 국제공통기준을 참조하여 분석한 사이드채널 공격 위협요소는 다음과 같다.[7]

- 물리적 공격 관련 위협
  - IC의 물리적 검사(T.P_Probe) : 공격자가 설계정보와 동작 내용을 밝혀내기 위해 TOE에 대한 물리적 검사를 수행할 수 있다.
  - IC에 대한 물리적 변경(T.P_Alter) : 공격자가 운용내용과 설계정보를 밝혀내거나, TSF (TOE Security Function) 데이터나 TOE 보안 기능을 변경하여 결국 TOE가 부정하게 사용될 수 있도록 TOE에 대한 물리적인 변경을 수행할 수 있다.

[표 1] 국외 세부평가기준(PP) 현황

	IC 칩	COS		Application
		Embedded software	Platform for Multi-application	
프랑스	PP0010			-
	PP 9911		..	
	PP 9806	PP 9810		
미국	SCSUG			-
	(IC Package)	(OS package)		
독일	BSI-PP-0002	-	-	-
일본	IPA-PP	JUKI-PP		
		NMDA-PP		PKI-PP
기타	-	VISA OP3PP		
	SSCD-PP			
	-	..		EMV-APP PP

- 논리적 공격 관련 위협
  - 오류삽입(T.Flt_Ins) : 공격자가 선택된 데이터의 반복주입 결과를 관찰함으로서 사용자 정보 및 TSF 정보를 결정할 수 있다.
- 정보 감시 위협
  - 정보누출(T.I_Leak) : 공격자가 TOE의 정상적인 사용으로부터 누출된 TSF 데이터를 이용할 수 있다. 전력분석(Power Analysis)는 정보누출의 한 예가 된다.
  - 다수관찰의 결합(T_Link) : 공격자가 자원 또는 서비스에 대해 여러 가지 사용점을 관찰하고 이러한 관찰을 결합하여 TSF 데이터를 추출할 수 있는 정보를 유추한다.
- 기타 위협
  - 환경적 스트레스(T.Env_Str) : 공격자가 TOE를 환경적인 스트레스 상에 노출시킴으로써 TSF 데

이터에서 에러를 발생시킨다. 기온, 전압, 클럭 주파수 등의 정상적인 파라미터 극한값 또는 외부 에너지 장과 같은 비정상적인 조건이 될 수 있다.

현재 스마트카드 제품 평가는 신분확인, 접근통제 등의 보안기능을 구현하고 응용인터페이스를 지원하는 개방형 플랫폼 COS와 IC 칩에 단순전력분석(SPA)/차분전력분석(DPA) 등의 취약성 분석 수행까지를 평가대상(TOE)의 범위에 포함하고 있으며 Application의 경우 대하여서는 평가대상에는 포함되지 않으나 평가신청자의 요청 시에 평가대상으로 포함이 가능하다.

평가보증등급(EAL)은 국내업체 및 사용자의 요구사항과 평가기술 및 장비 등을 고려하여 결정될 것이며 현재로는 EAL3+ ~ EAL4+등급이 유력하다.

Ⅲ. 스마트카드 상의 사이드 채널 공격

본 장에서는 타이밍, 오류삽입, 전력분석의 세가지 대표적인 사이드 채널 공격 유형을 살펴본다.

3.1 타이밍 공격

타이밍 공격은 장치내의 알고리즘 수행시간을 측정하고, 이를 분석하여 암호 키와 같은 비밀 정보를 추출해내는 공격 방식이다.

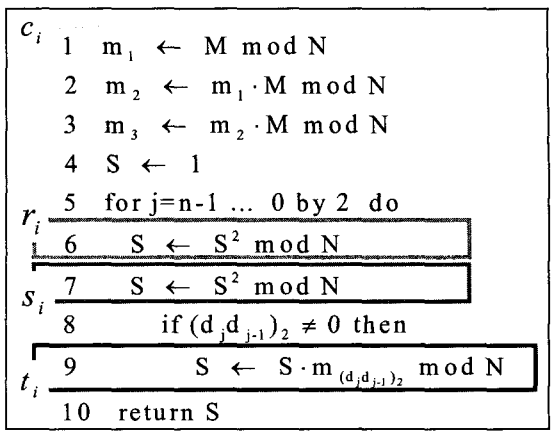
스마트카드 상에서 타이밍 공격을 하기위한 가정은 다음과 같다.

- o 공격자는 스마트카드의 암호 알고리즘에 접근하여 매번 메시지에 대한 서명값을 얻어낼 수 있다.
- o 공격자는 사용된 암호 알고리즘을 정확하게 알고 있다.
- o 항상 동일한 키 값이 사용된다.

타이밍 공격은 타이밍 측정값들 간의 상관관계를 분석하여 암호 키 비트를 추측하는 통계 모델이다. 따라서, 공격자는 각 연산 실행시간을 세밀하게 측정하여 분석함으로써 암호시스템에 관련된 키 또는 비밀정보를 알아낼 수 있다.

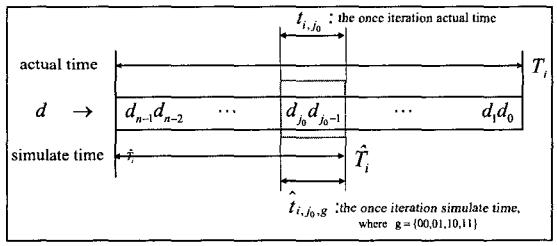
Diffie-Hellman, RSA, CRT기반 RSA, DSS 등 여러 알고리즘에서 타이밍 공격방법이 제안되었다. 이 중 RSA 서명방식에 대한 타이밍 공격을 간단히 설명하면 다음과 같다.

그림 1은 RSA사에서 개발한 RSA 서명 알고리즘



〔그림 1〕 RSAREF 2.0 RSA서명 알고리즘

RSAREF 2.0 이다. M은 메시지, d는 비밀키의 이진배열, N은 큰 소수 p, q값의 곱을 의미한다. RSAREF 2.0에서는 한 번의 루프에 두 비트의 키 값을 취하여 중간 서명 값을 생성하고, 모든 루프가 실행되면 완전한 서명값 S를 얻을 수 있다.



〔그림 2〕 T_i와 $\hat{t}_i$의 상관관계

타이밍 공격은 각 메시지들에 대해 서명을 수행한 총 시간을 구하는 것으로부터 시작한다. 그림 2에서 T_i는 스마트카드를 이용하여 i번째 메시지를 서명하는데 걸린 총 시간을 의미하며 다음과 같이 표현된다.

$$\begin{aligned} T_i &= e_i + c_i + (r_{i,n-1} + s_{i,n-1} + t_{i,n-1}) \\ &\quad + (r_{i,n-3} + s_{i,n-3} + t_{i,n-3}) + \cdots \\ &\quad + (r_{i,1} + s_{i,1} + t_{i,1}) \\ &= e_i + c_i + \sum_j (r_{i,j} + s_{i,j} + t_{i,j}) \end{aligned}$$

여기서 e_i는 시간 측정 시에 발생하는 측정오류 값이다.

$\hat{T}_i$는 $(d_{n-1}d_{n-2})$부터 (d_jd_{j-1})까지 키 비트가 정확하게 추측되었다는 가정 하에 시뮬레이터를 통해 스

마트카드와 동일한 알고리즘으로 다음 j_0번째 키 비트까지 수행된 시간을 측정한 값이다.

$$\hat{T}_i = c_i + \sum_{j>j_0} (r_{i,j} + s_{i,j} + t_{i,j}) + (r_{i,j_0} + s_{i,j_0} + \hat{t}_{i,j_0,g}) \quad ,where\ g \in \{00,01,10,11\}$$

여기서 g는 j_0번째 루프에 입력될 수 있는 4가지 경우(00,01,10,11)를 갖는 키 비트 값으로 $d_{j_0} d_{j_0-1}$의 추정치이다. 결국 $\hat{t}_{i,j_0,g}$는 추측된 g 비트 값으로 시뮬레이터에서 j_0번째 루프를 수행하는데 걸린 시간이다.

타이밍 공격은 추정한 g에 대하여 $T_i - \hat{T}_i$의 분산을 비교하여 올바른 키 비트를 추정한다. $T_i - \hat{T}_i$의 분산은 다음과 같다.

$$T_i - \hat{T}_{i,j_0,g} = c_i + \sum_{j<j_0} (r_i + s_i + t_i) + (t_{i,j_0} - \hat{t}_{i,j_0,g}),$$

$$Var(T - \hat{T}_{j_0,g}) = Var(c) + Var(\sum_{j<j_0} r_j) + Var(\sum_{j<j_0} s_j) + Var(\sum_{j<j_0} t_j) + Var(t_{j_0} - \hat{t}_{j_0,g})$$

여기서, $r_j = s_j$이고, 각 t_j는 3/4의 확률로 수행시간을 갖는다. 결국, $Var(T - \hat{T}_{j_0,g})$는 다음과 같다.

$$Var(T - \hat{T}_{j_0,g}) = Var(c) + (j_0 - 2)\,Var(s) + \left(\frac{3}{4}\frac{j_0-2}{2}\right) Var(t) + Var(t_{j_0} - \hat{t}_{j_0,g})$$

따라서, 올바른 키를 추측하였을 경우, 즉 $t_{j_0} = \hat{t}_{j_0,g}$에서는 분산이 다음과 같다.

$$Var(T - \hat{T}_{j_0,g}) = Var(c) + (j_0 - 2)\,Var(s) + \left(\frac{3}{4}\frac{j_0-2}{2}\right) Var(t)$$

추측한 키 값이 틀렸을 경우에는 두 가지의 경우로 나누어 분산을 구할 수 있는데, 첫째 t_{j_0}와 $\hat{t}_{j_0,g}$ 둘다 0이 아니고, $t_{j_0} \neq \hat{t}_{j_0,g}$인 경우 분산은 다음과 같다.

$$Var(T - \hat{T}_{j_0,g}) = Var(c) + (j_0 - 2)\,Var(s) + \left(\frac{3}{4}\frac{j_0-2}{2}\right) Var(t) + 2\,Var(t)$$

둘째로 t_{j_0}와 $\hat{t}_{j_0,g}$ 둘 중 하나가 0이고, $t_{j_0} \neq \hat{t}_{j_0,g}$인 경우의 분산은 다음과 같다.

$$Var(T - \hat{T}_{j_0,g}) = Var(c) + (j_0 - 2)\,Var(s) + \left(\frac{3}{4}\frac{j_0-2}{2}\right) Var(t) + Var(t)$$

따라서, 각 루프에서 올바른 키를 추측하였을 경우에는 분산 $Var(T - \hat{T}_{j_0,g})$이 가장 작은 값을 갖게 된다. 이러한 키 추정과정을 반복하여 RSAREF 2.0 서명 알고리즘의 키 값을 추정할 수 있다.

타이밍 공격에 대한 대응기법으로는 메시지 블라인딩 기법이 가장 효율적으로 사용된다. 서명할 메시지 M을 $M' = r^e \cdot M \mod N$, $random\ r \in Z_N^*$로 블라인딩하여 서명 알고리즘을 수행하면, 공격자는 블라인딩된 메시지 M'를 알 수 없기 때문에 시뮬레이터를 통하여 정확한 루프 수행 시간의 측정이 어렵다. 현재 이 대응기법은 RSA사의 새로운 버전의 서명 알고리즘에 적용되어있다.

3.2 오류삽입 공격

스마트카드 상의 암호 알고리즘이 동작하는 과정에서 발생하는 하드웨어 혹은 소프트웨어적인 오류 및 에러는 보안에 영향을 미친다. 예를들어, 알고리즘 수행도중 저장되어 있는 비트가 갑작스런 전력의 과부하에 영향을 받게 된다면, 스마트카드는 그 이후의 알고리즘을 수행한 결과로 의도하지 않은 메시지를 출력하게 된다.

Differential Fault Attack(DFA, 차분오류공격)으로 알려진 공격방식은 실제 DES 암호 알고리즘에 사용되는 키를 추출해낼 수 있다. 또한 블록 알고리즘인 DES 뿐만 아니라 RSA 공개키 알고리즘 또한 유사한 공격방식에 취약함이 증명되었다.

RSA에 대한 오류 공격을 간단하게 살펴보면 다음과 같다. RSA의 안전성은 큰 정수 N에 대한 인수분해의 어려움에 기반 한다. 큰 정수 N은 소수 p와 q의 곱으로 이뤄져 있다. 만약 공격자가 N의 인수인 p와 q값을 추론해 낼 수 있다면 RSA에 대한 키 값은 쉽게 추론할 수 있다. 이러한 RSA 알고리즘에서의 오류공격은 정수 N의 인수인 소수 p, q값 중 하나를 추론하여 비밀키를 획득하는 방법을 사용하고 있다.

오류공격은 메시지와 이를 서명하는 도중 시스템 오류를 발생시켜 생성한 잘못된 서명 값과 공개키를

사용하여 이뤄진다. 세 개의 값을 사용한 간단한 수식을 통해 소수 p 혹은 q 값을 생성할 수 있다.

CRT(중국인의 잉여정리)는 RSA 서명의 속도를 높이기 위해 사용된다. RSA는 메시지 M을 p와 q의 곱인 N상에서 모듈러 연산을 하는데 중국인의 잉여정리를 이용한 CRT기반 RSA는 그림 3과 같이 p와 q상에서 각각 모듈러 연산을 이용한다.

$$S = M^d \bmod N, \quad N = p \cdot q$$

$$S_p = M^{d_p} \bmod p, \quad d_p = d \bmod (p-1)$$

$$S_q = M^{d_q} \bmod q, \quad d_q = d \bmod (q-1)$$

$$S = u_p S_p + u_q S_q \bmod N$$

$$where\ u_p = qq^{-1} \bmod p,\ u_q = pp^{-1} \bmod q$$

〔그림 3〕 CRT 기반 RSA 서명방식

여기서 u_p와 u_q는 다음과 같다.

$$u_p = \begin{cases} 1 \mod p \\ 0 \mod q \end{cases}, \quad u_q = \begin{cases} 0 \mod p \\ 1 \mod q \end{cases}$$

S_p와 S_q의 선형 계산은 각각의 모듈러 연산에 비교하여 작은 연산 시간이 요구된다. CRT를 이용한 서명은 mod N 상에서 한 번의 연산을 수행하는 것 보다 약 N의 절반 크기를 가진 p, q상에서 분리하여 연산할 수 있어 RSA 서명을 하는데 더 효율적이게 된다. 이러한 이유로 인해 많은 알고리즘에서 CRT 기반의 RSA를 사용하고 있다.

Bonch, DeMillo 그리고 Lipton은 만약 같은 메시지에 대해 하나는 잘못된 서명값(S_p와 S_q 값 중 하나가 잘못된 값을 갖는 경우)을 다른 하나는 정당한 서명값인 두 개의 서명을 공격자가 가지게 된다면, N을 소인수 분해할 수 있음을 관측해 냈다. 이 기술을 정리하면 다음과 같다.

$$\gcd(N, S - S') = p$$

S' : 잘못된 서명값

여기서 S'는 S_p를 계산하는 도중에 오류의 발생으로 잘못 생성된 값이다. 이것은 $(M^{d'} \cdot p \cdot (p^{-1} \bmod q))$으로 표시할 수 있다.

$$\begin{aligned} S - S' &= \\ &(M^d \cdot q(q^{-1} \bmod p)) + (M^{d'} \cdot p(p^{-1} \bmod q)) \\ &-(M^d \cdot q(q^{-1} \bmod p)) + (M^d \cdot p(p^{-1} \bmod q)) \\ &= (M^{d'} \cdot p(p^{-1} \bmod q)) - (M^d \cdot p(p^{-1} \bmod q)) \\ &= (M^{d'} - M^d) \cdot p \cdot (p^{-1} \bmod q) \end{aligned}$$

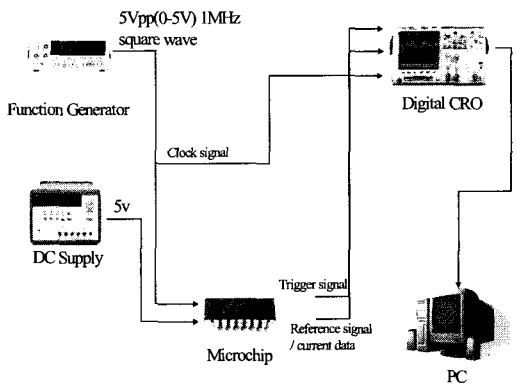
따라서, $S - S'$는 p의 값을 가지고 있기 때문에 $\gcd(N, S\text{-}S') = p$가 됨을 알 수 있다.

오류공격에 대한 대응방법으로 암호문을 출력하기 전에 암호화된 메시지를 복호화하여 입력값과 비교하여 오류여부를 검증하는 방법이 있다. 이에 걸리는 시간은 공개키의 길이에 의존하게 된다. 보통 작은 길이의 키 값을 사용하는 공개키 ($e = 3$)로 인해 서명의 검증에는 많은 시간이 들지 않는다. Shamir는 공개키의 길이가 클 경우, CRT를 사용함으로서 전체서명 검증보다 보다 효율적으로 검증할 수 있는 기법을 제안하였다.[4] 그러나 이러한 대응기법은 오류 검증을 위한 알고리즘의 수행 시에도 오류가 발생할 수 있어 완벽한 대응책이 되지는 못한다.

3.3 전력 분석 공격(SPA/DPA)

전력 분석 공격은 스마트카드가 암호화 기능을 실행하면서 소비하는 전력을 단순 또는 차분 분석함으로써, 공격자는 스마트카드 내에서 일어나는 프로세스를 분석하여 비밀키를 복구할 수 있도록 지원하는 일부 정보를 얻을 수 있다.

이와 같은 공격을 수행하기 위해 필요한 장비는 그림 4와 같이 오실로스코프와, IC 칩, PC, 전력 공급



〔그림 4〕 전력 분석 공격 장비 구성도

기, 파형 생성기 등이 있다. 오실로스코프와 같은 장비는 설비가 잘 갖추어진 연구소들의 경우, 표본 전압 차이를 1GHz 이상의 높은 샘플링(sampling)과 정확성을 가지고 디지털 방식으로 측정할 수 있는 설비를 갖추고 있는 경우도 있다. 또한 20 MHz 이상에서 표본 추출이 가능하며 데이터를 PC로 전송할 수 있는 장치들은 저렴한 비용으로 구비할 수 있어 공격의 위험성이 높다.

전력 분석 공격을 하기 위한 가정은 다음과 같다.

- o 공격자는 사용된 암호 알고리즘을 정확하게 알고 있다.
- o 항상 동일한 키 값이 사용된다.
- o 공격자는 상당수의 평문과 암호문 쌍을 가지고 있다.

3.3.1 단순 전력 분석(SPA, Simple Power Analysis)

일반적으로 단순 전력 분석(SPA, Simple Power Analysis)은 암호 기능이 실행되는 동안 장치의 전력 소비 측정값을 직접 해석하는 기법이다. SPA는 키 자료뿐만 아니라 스마트카드의 작동에 대한 정보까지 얻어낼 수 있다.

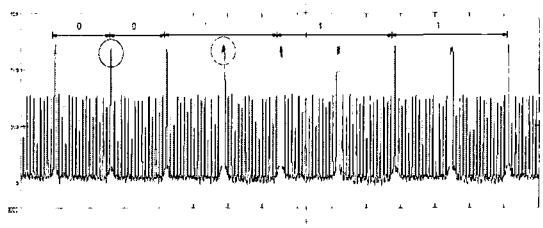
전력 소비는 마이크로프로세서 명령에 따라 다양하게 나타난다. 따라서, 마이크로프로세서에 의해 실행된 각각의 암호 연산들의 전력 특징이 서로 다르기 때문에 DES 암호과정의 라운드, RSA 연산 등과 같은 광범위한 특징들을 식별할 수 있으며, 보다 높은 샘플링에서는 개별 명령까지도 구별할 수 있다.

SPA는 타이밍 공격과 마찬가지로 곱셈과 제곱 연산 사이에 소비되는 전력의 차이를 규명함으로써 RSA의 개인키를 추론하는 데 사용될 수 있다. 이와 유사하게, 대다수 DES 구현들도 순열과 정보의 이동 내에서 명확한 차이를 보이기 때문에 비밀 키를 추론할 수 있다.

SPA는 관측 데이터를 분석하여 비밀정보를 알아내는 방식이므로 전력 소비 데이터를 얼마만큼 자세히 관측하느냐가 공격의 성공 여부를 결정한다. 즉, 장치의 전력 소비를 측정하는 오실로스코프의 샘플링률에 따라서 보다 정확한 값들을 얻어낼 수 있다.

그림 5는 RSA 알고리즘 수행과정에서 전력 소비를 측정한 것이다. 비밀키 값의 비트열에서 0비트로 연산이 되는 부분과 1비트로 연산되는 부분에서 전력 소비의 차이를 확인할 수 있다.

이렇게 전력 소비 데이터에서 비트 값들에 의해 발생되는 전력 소비의 특징을 분석하여 비밀키 값에 대



〔그림 5〕 전력 소비 측정 데이터

한 추정이 가능하다.

단순 전력분석공격에 대한 대응방법으로 몇 가지가 있다. 첫째, 조건문과 루프 카운터(loop counter)에 비밀 값의 사용을 피하는 것이다. 이 조건은 추가적인 최적화 수행 코드를 작성할 필요가 있다. 둘째로, 명령 사이의 전력 소비의 차를 줄이는 것이다. 이러한 방법은 하위단계에서 로직 설계를 고려해야한다. 마지막으로 랜덤한 전력 잡음을 발생시켜주는 것이다. 프로세서의 동작과 상관없는 전력소비를 발생시켜 SPA 공격을 방해하는 방식이다.

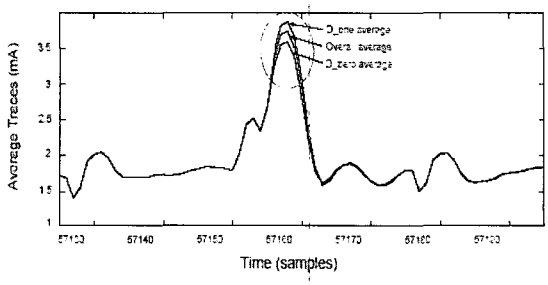
3.3.2 차분 전력 분석(DPA, Differential Power Analysis)

차분 전력 분석(DPA, Differential Power Analysis)은 키를 추정하기 위해 전력 소비 데이터에 대해 통계분석 방법을 사용한다. 따라서, 단순 전력 분석에 비하여 샘플링 능력이 낮은 오실로스코프로도 공격이 가능한 효율적인 공격법이다.

일반적인 조건에서, DPA 분석은 평문 또는 암호문과 전력 소비량 측정값을 사용한 통계적인 방법을 사용한다. 이는 그림 6과 같이 비밀키의 값이 1비트일 때와 0비트일 때 전력 소비의 차이를 이용한 것이다.

어떤 암호학적 알고리즘이 있어 공격자가 자유롭게 암호문을 생성할 수 있고 중간의 과정을 볼 수 있다고 가정하자.

입력된 각 메시지에 대해 암호 연산 중 생성되는 중간 값 중의 한 비트를 b라고 하자. 이 비트 b는 각 연산



〔그림 6〕 비트 값에 따른 전력 소비의 차이

과정에 어떻게든 영향을 받을 것이며 따라서, 그 영향력은 수집한 전력 소비 파형에 포함되어 있을 것이다. 각 메시지별로 비트 b가 갖는 값에 따라 전력 소비 파형을 다음과 같이 나눈다.

$$T_0 = \{T_i : b = 0\}$$
$$T_1 = \{T_i : b = 1\}$$

샘플링 률이 k이면 각 파형은 k개의 값을 갖는다. 따라서, 비트 b에 의해 분류된 파형들의 평균 파형은 다음과 같다.

$$A_0[j] = \frac{1}{|T_0|} \sum_{T_i \in T_0} T_i[j]$$
$$A_1[j] = \frac{1}{|T_1|} \sum_{T_i \in T_1} T_i[j]$$

여기서 , $|T_1|+|T_0| = n$이다. 즉 $T_i[j]$는 i번째 파형의 측정값 중 j번째 전력 소비 데이터이다.

비트 b의 연산시점이 j^*에서 일어났다고 가정하면, 결국 비트 b에 따라 나누어진 각 그룹의 평균차는 다음과 같이 2가지로 발생하게 된다.

$$E[T_i[j^*] \mid b=1] - E[T_i[j^*] \mid b=0] = \epsilon,$$

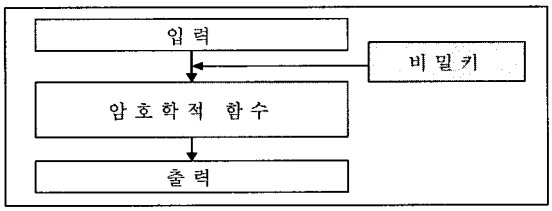
$j \neq j^*$일때,

$$E[T_i[j^*] \mid b=1] - E[T_i[j^*] \mid b=0] = 0$$

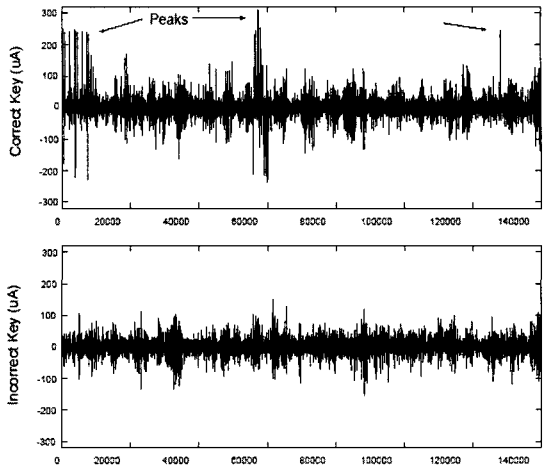
결국, 측정된 메시지가 충분히 많다면, 다음과 같은 결과를 얻게 된다.

$$\lim_{\infty} \Delta[j] = \begin{cases} \epsilon & \text{for} j = j^* \\ 0 & otherwise \end{cases}$$

아래의 방법으로 비밀키를 추출하는 과정은 다음과 같다. 대부분의 대칭키 암호는 그림 7과 같은 프로세스로 이루어져있다. 결국 공격자는 키 값을 모르기 때문에 입력으로부터 출력을 생성하지 못한다.



〔그림 7〕 암호학적 알고리즘



〔그림 8〕 통계적 방법을 사용하여 정확히 키를 유추했을 경우와 유추하지 못했을 경우

따라서, 입력값에 가능한 키 값을 모두 적용하여 실제 함수에 입력되는 값을 생성한다. 그 비트에 따라 측정된 파형을 그룹화하여 위의 공격 방법을 적용한다. 각각의 파형에서 그림 8과 같이 정확한 키를 추측하였을 경우와 그렇지 못 한 경우에서 서로 다른 파형이 나타나기 때문에 올바른 키를 추정할 수 있다.

DES의 경우 한 라운드의 f 함수는 그림 7에서 암호학적 함수로 볼 수 있으며, 입력 값과 키 값이 6비트의 값을 갖기 때문에 오직 2^6개의 키만 모두 적용하여 파형을 분류하면 올바른 키를 복구해 나갈 수 있다. 이러한 과정을 반복적으로 적용하면 결국, DES의 완벽한 키를 추출할 수가 있다.

차분 전력 분석의 대응방안 중 하나로 프로세스 수행도중 발생되는 전력의 크기를 줄여 그 크기가 최소한의 값을 가지도록 최적화 하는 것이 있다. 공격자는 이러한 대응방안을 무력화하기 위해서 더 많은 메시지와 전력소비 데이터가 요구되므로 공격이 어려워진다.

또 다른 방법으로 전력공급 장치와 그라운드 라인 사이에 전력파동을 제거하여 공격자가 전력을 측정하더라도 일정한 전력값이 출력되도록 케패시터(capacitor)를 설치하는 것이 있다.

또한 타이밍 공격과 마찬가지로 메시지의 블라인딩 기법을 이용하여 전력분석을 방지할 수 있다. 스마트카드가 알고리즘을 수행하는 동안 블라인딩 된 메시지가 연산되기 때문에 전력 소비 파형도 그에 따라 달라지므로 공격자의 분석을 어렵게 한다.

표 2는 단순 전력 분석과 차분 전력 분석의 차이는

〔표 2〕 SPA와 DPA 비교

	단순전력분석	차분전력분석
필요 관측 데이터	단일	400 ~ 1000
장비 성능	고	저
분석방법	소모 전력의 직관적 분석	통계 분석

다음과 같다.

현재까지 여러 알고리즘이 구현 방식에 따라 사이드 채널 공격에 대한 취약점을 가지고 있음이 밝혀졌다. 다음 표 3은 여러 알고리즘에 대한 사이드 채널 공격의 수행 가능 여부를 정리한 것이다.

〔표 3〕 사이드 채널 공격의 수행가능 여부

	타이밍공격	오류공격	전력분석
RSA	O	O	O
DES	O	O	O
AES	O	O	O
Elgamal Signature	O	O	O
IDEA	O	O	O
DSA	O	O	O
RC5	O	O	O
ECDSA	△	O	O

Ⅳ. 결 론

본 고에서는 수학적으로 이미 안전성이 증명된 알고리즘이 구현 방식에 따라 사이드 채널 공격에 취약할 수 있음을 기술하였다. 차분전력분석의 경우 거의 대부분의 암호 알고리즘에 적용이 가능한 강력한 공격 방법이다. 이러한 공격에 대응하는 것은 앞으로 광범위하게 사용될 스마트카드의 안전성에 있어서도 매우 중요하다. 스마트카드 관련 제품의 제조자들 또한 칩의 설계 단계에서부터 이에 대한 대응기법을 적용하는 것이 필요하다고 판단된다.

참 고 문 헌

〔1〕 P.Kocher, "Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS and Other Systems", *Advance in cryptology-CRYPTO '96*, LNCS 1109, pp.104-113, Springer-Verlag, 1996

〔2〕 P.Kocher, J. Jaff and B. Jun, "Introduction to Differential Power Analysis and related Attacks", Technical report, Cryptography Research Inc., 1998

〔3〕 E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", *Advanced in Cryptology- CRYPTO '97*, LNCS 1294, pp.513-525, Springer-Verlag, 1997

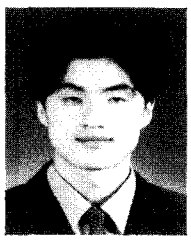
〔4〕 A. Shamir, "How to check modular exponentiation", presented at the rump session of EUROCRYPT '97, 1997

〔5〕 W. Shindler, "A Timing Attack against RSA with the Chinese Reminder Theorem", *Cryptographic Hardware and Embedded Systems-CHESS 2000*, LNCS 1965, pp.109-124, 2000

〔6〕 이훈재, 이상곤 외, "스마트카드 비밀채널 평가/분석기술 연구", 한국전자통신연구원 부설 국가보안기술연구소, 최종보고서, 2002

〔7〕 http://www.kisa.or.kr 2003-52호(정보보호시스템공통평가기준중개정)

〈著 者 紹 介〉



조 은 성 (Eunsung Cho)
학생회원

2000년 8월 : 성균관대학교 산업공학과(공학사)
2002년 8월 : 성균관대학교 산업공학과 석사
2003년 3월~현재 : 성균관대학교 정보통신공학부 컴퓨터공학과 박사과정
관심분야 : 전자공학, 통신공학, 정보보호

원 동 규 (Dongkyu Won)
학생회원

2003년 2월 : 인천시립대학교전자공학과(공학사)
2003년 3월~현재 : 성균관대학교 정보통신공학부 석사과정

양 형 규 (Hyungkyu Yang)
정회원

1983년 2월 : 성균관대학교 전자공학과 졸업(공학사)
1985년 2월 : 성균관대학교 대학원전자공학과(공학석사1984년 12월~1991년 2월 : 삼성전자 선임연구원
1995년 2월 : 성균관대학교 대학원 정보공학과(공학박사)
1995년 3월~현재 : 강남대학교 컴퓨터미디어공학부 부교수

김 승 주 (Seungjoo Kim)
종신회원

1994년 2월 : 성균관대학교 정보공학과(공학사)
1996년 2월 : 성균관대학교 대학원 정보공학과 (공학석사)
1999년 2월 : 성균관대학교 대학원 정보공학과(공학박사)
1998년 12월~2004년 2월 : 한국정보보호진흥원(KISA) 팀장
2004년 3월~현재 : 성균관대학교 정보통신공학부 조교수

원 동 호 (Dongho Won)
종신회원

1976년~1988년 : 성균관대학교 전자공학과 (학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구소 전임 연구원
1985년~1986년 : 일본 동경공대 객원연구원
1988년~1999년 : 성균관대학교 교학처장, 전기·전자 및 컴퓨터공학부장, 정보통신대학원장
1996년~1998년 : 국무총리실 정보화추진위원회 자문위원
2002년~2003년 : 한국정보보호학회장
2002년~2004년 : 성균관대학교 연구지원처장
현재 : 성균관대학교 정보통신공학부 교수, 한국정보보호학회 예회장, 정통부지정 정보보호인증기술연구센터 센터장