데이터 상태천이를 이용한 S/N비를 향상시킨 DPA공격

구경본*, 하재철**, 문상재*, 임선간***, 김승주***

*경북대학교 전자전기공학부, **나사렛대학교 전산정보학과, ***한국정보보호진흥원

DPA attack with high S/N ratio using data transition

Kyung-Bon Koo*, Jae-Cheol Ha**, Sang-Jae Moon*, Seon-Gan Lim***,

Seung-Joo Kim***

*School of Electronic & Electrical Eng., KyungPook National University

**Dept. of Computer Science and Information, Korea Nazarene University

***Korea Information Security Agency

요 약

스마트카드의 가장 큰 특징 중 하나로 자체적인 보안 기능을 들 수 있다. 하지만, 스마트카드는 내부의 암호 시스템이 수행될 때, 비밀키와 관련된 여러 가지 물리적인 정보를 누출하게 된다. 본 논문에서는 스마트카드의 전력 소비 신호를 이용하여, 내장된 암호 알고리듬의 비밀키를 알아내는 개선된 DPA(differential power analysis)공격을 제안한다. 제안하는 DPA공격은 SRAM에서의 데이터 상태천이를 이용하여 DPA신호의 S/N비를 높임으로써, 보다 효과적인고 강력한 DPA공격이다. 따라서 스마트카드 설계자는 이러한 점을 고려하려 시스템을 설계해야 할 것이다.

I. 서론

스마트카드는 마이크로 프로세서와 메모리를 내장하고 있는 플라스틱 카드로서, 자체적인 정보의 저장과 연산 능력을 가진다. 또한, 기존의 자기카드와는 달리 인증과 암호를 위한 고유의 보안 영역을 가지고 있어 각종 카드의 역할을 통합적으로 수행하는데 많이 사용되고 있다. 최근, 스마트카드가 동작될 때, 물리적으로 노출되는 정보를 이용하여 내장된 암호 알고리듬을 공격하는 여러 가지 방법들이 소개되었다. 이러한 암호 공격을 부-채널 공격이라 한다. 부-채널 공격에는 수행시간 정보를 이용한 시차 공격, 오류 주입을 통한 오류 공격, 소비전력 정보를 이용한 DPA공격 등이 있다[1][2][3]. 이러한 부-채널 공격 중에서 DPA공격은 현재 가장 강력한 공격 방법으로 보고되고 있다. 지금까지 스마트카드 내부에서 소프트웨어적으로 구현된 여러 가지 암호 알고리듬에 대한 DPA공격 및 그 대응 방안이 제시되어 왔으며, 대부분의 암호 알고리듬이 이러한 공격에 취약한 것으로 보고되고 있다[4],[5],[6].

본 논문에서는 휘발성 메모리인 SRAM의 전력 소비 형태를 분석함으로써, 데이터의 상태천이를 이용한 S/N비를 향상시킨 DPA공격을 제안한다.

II. DPA공격

DPA공격은 많은 평문에 대한 스마트카드 암호 알고리듬의 전력 소비 신호를 통계적으로 분석하여 비밀키를 알아내는 전력 분석 공격이다. 일반적인 DPA공격은 스마트카드의 전력 소비 모델 중 hamming weight 모델을 바탕으로 한다. Hamming weight 모델은 데이터가 메모리로 전송될 때, 데이터의 hamming weight에 따라 전력 소비가 다르게 나타난다는 것이다. 일반적인 스마트카드 암호 알고리듬에 대한 DPA공격 과정은 다음과 같다.

단계1 : 임의의 평문 $P_i(1 \le i \le m)$ 에 대한 암호문 C_i와 전력 소비 신호 $T_i[j]$쌍을 실험을 통하여 구한다.

단계2 : 비밀키에 대하여 공격자는 최상위 비트부터 차례대로 추측한다.

단계3 : 공격자가 정의하는 분류함수 D를 이용하여 암호 알고리듬의 전력 소비 신호를 두 부류로 분류한다.

$$D_0 = \{T_i[j] | D(K_s, C_i \text{or } P_i, b) = 0\},$$

$$D_1 = \{T_i[j] | D(K_s, C_i \text{or } P_i, b) = 1\}$$

(K_s: 비밀키의 일부, b: 분류되는 데이터위치)

단계4 : 양분한 전력 소비 신호를 각각 평균하여 그 차분 신호 $\Delta_D[j]$를 구한다.

단계5 : 만약, 추측한 비밀키의 일부가 실제 비밀키와 같다면, 차분 신호 $\Delta_D[j]$는 spike신호를 가진다.

단계6 : 만약, 추측한 비밀키의 일부가 옳지 않다면, 차분 신호 $\Delta_D[j]$는 spike신호를 가지지 않으므로, 다시 단계2로 간다.

위의 과정에서 단계 2부터 6까지를 반복 수행함으로써, 사용되는 암호 알고리듬의 전체 비밀키를 구할 수 있다. 만약, 추측한 비밀키가 실제 암호 시스템의 비밀키와 동일하다면, 분류함수의 출력값은 실제 암호 알고리듬이 동작할 때, 생성하는 특정 위치의 데이터와 동일할 것이다. 그리고, 분류함수의 출력 값 즉 특정 위치의 데이터가 스마트카드에서 반응하는 지점을 j^*라고 한다면, 평균된 전력 소비 신호인 $\overline{D_0}[j], \overline{D_1}[j]$는 $j = j^*$지점에서 서로 다른 바이어스값을 가지게 되므로, 그 차분 신호인 $\Delta_D[j]$는 j^*지점에서 spike신호를 생성한다.

■추측한 비밀키가 실제 비밀키와 동일한 경우:

$$\Delta_D[j] = \overline{D_1}[j] - \overline{D_0}[j] \approx \begin{pmatrix} \varepsilon, & j = j^* \\ 0, & \forall j \neq j^* \end{pmatrix} \quad (1)$$

■추측한 비밀키가 실제 비밀키와 다를 경우:

$$\Delta_D[j] = \overline{D_1}[j] - \overline{D_0}[j] \approx \begin{pmatrix} 0, & j = j^* \\ 0, & \forall j \neq j^* \end{pmatrix} \quad (2)$$

Ⅲ.제안하는 데이터 상태천이를 이용한 DPA공격

본 장에서는 데이터 상태천이를 이용한 S/N비를 향상시킨 DPA공격을 제안한다. 먼저, 스마트카드의 SRAM에 대한 전력 소비 형태에 대하여 간단히 알아보고, 이러한 전력 소비 형태를 바탕으로 제안하는 DPA공격을 설명한다. 그리고, 여러 암호 알고리듬에 대하여 적용한 후, 기존의 DPA공격과 제안하는 DPA공격에 대하여 S/N비 측면에서 비교 분석한다.

1. SRAM의 전력 소비 형태

스마트카드의 SRAM은 기본적으로 CMOS 회로로 이루어져 있으며, 그림 1은 한 비트를 저장할 수 있는 SRAM의 구조이다.

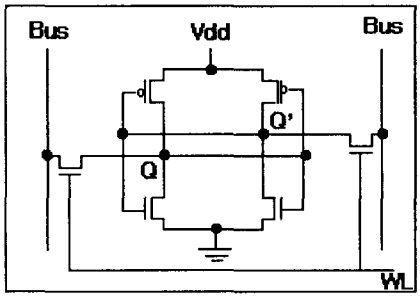

그림 1 : SRAM의 구조

일반적인 디지털 CMOS 회로의 전력 소비는 static power dissipation과 short circuit dissipation 그리고, dynamic switching dissipation의 형태로 분리된다[7]. static power dissipation은 입력되는 데이터와 관계없이 CMOS 회로에서 고정으로 소비되는 전력이다. 그리고, short circuit dissipation과 dynamic switching dissipation는 데이터의 상태 천이에 의해 소비되는 전력으로써, 전체 전력 소비의 대부분을 차지한다.

$$P_{dynamic} = C \cdot V_{dd}^2 \cdot f \quad (3)$$

(C:capacitance, V_{dd}:supply voltage, f:frequency)

식 3은 dynamic switching dissipation이 크기를 나타낸다. 이와 같이, CMOS회로로 구성된 SRAM의 소비 전력은 입력되는 데이터의 상태천이 유무에 많은 영향을 받음을 알 수 있다.

2. 제안하는 DPA공격

본 절에서는 SRAM에서의 데이터 상태 천이를 이용한 S/N비를 향상시킨 DPA공격을 제안한다. 제안하는 DPA공격은 스마트카드의 SRAM이 전원을 끄기 전까지 이전의 데이터를 저장하고 있는 성질을 이용한다. 만약, 공격자가 SRAM의 이전 데이터와 새롭게 입력되는 데이터에 대한 정보를 추측할 수 있다면, SRAM에서의 데이터 상태천이 유무를 판단할 수 있을 것이다. 제안하는 데이터 상태천이를 이용한 DPA공격 과정은 다음과 같다.

단계1 : 임의의 평문 $P_i(1 \le i \le m)$에 대한 암호문 C_i와 전력 소비 신호 $T_i[j]$ 쌍을 실험을 통하여 구한다.

단계2 : 비밀키에 대하여, 최상위 비트부터 차례대로 추측한다.

단계3 : 공격자가 정의하는 분류함수 D를 이용하여 암호 알고리듬의 전력 소비 신호 $T_i[j]$를 분류한다.

$D_0 = \{ T_i[j] | D(K_s, C_{i-1}, C_i, b) =$

$(0 \rightarrow 0, 1 \rightarrow 1)$ or (low transition count) $\}$,

$D_1 = \{ T_i[j] | D(K_s, C_{i-1}, C_i, b) =$

$(0 \rightarrow 1, 1 \rightarrow 0)$ or (high transition count) $\}$

(K_s:비밀키의 일부, b:분류되는 데이터위치)

단계4 : 양분한 전력 소비 신호를 각각 평균하여 그 차분 신호 $\Delta_D[j]$를 구한다.

단계5 : 만약, 추측한 비밀키의 일부가 실제 비밀키와 같다면, 차분 신호 $\Delta_D[j]$는 spike신호를 가진다.

단계6 : 만약, 추측한 비밀키가 옳지 않다면, 다시 단계 2로 간다.

제안하는 DPA공격은 단계 3에서 데이터의 상태천이 따라 전력 소비 신호를 분류한다. 단계 3에서 분류함수 $D(K_s, C_{i-1}, C_i, b)$는 평문 P_i가 암호화 될 때, 메모리에 입력되는 데이터의 상태천이 유무를 추측하기 위하여 C_{i-1}이 추가된다. 적용되는 암호 알고리듬에 따라 분류함수에 입력되는 암호문 C_{i-1}과 C_i는 평문 P_{i-1}과 P_i로 대신할 수 있다. 분류함수의 b는 분류되는 데이터의 특정 위치이다. 먼저, 공격자는 암호문 C_{i-1}와 추측하는 비밀키 K_s를 이용하여 분류하고자하는 데이터의 특정 위치 b에 입력되는 데이터를 계산한다. 그리고, 다시 암호문 C_i와 추측하는 비밀키 K_s를 이용하여 b에 입력되는 데이터를 계산한다. 이렇게 함으로써, 공격자는 평문 P_i가 암호화될 때, b에 입력되는 데이터의 상태천이 유무를 추측할 수 있다. 만약, 추측한 비밀키가 실제 스마트카드에 내장된 비밀키와 동일하다면, 차분 신호 $\Delta_D[j]$는 분류하는 데이터가 SRAM에 저장되는 시점에서 spike신호를 만들 것이다. 하지만, 추측한 비밀키가 옳지 않다면, 분류함수와 전력 소비 신호의 상관 관계가 없으므로 차분 신호는 spike 신호를 가지지 않는다.

3.1절에서 설명했듯이, SRAM에 의한 전력 소비는 입력되는 데이터의 상태천이에 의존하므로, 제안하는 DPA공격의 분류함수를 이용한다면, 데이터의 상태천이에 대한 정보를 추측할 수 있다. 데이터의 상태 천이가 많을 수록, 전력 소비가 많게 되고, 데이터의 상태 천이가 적을수록 그만큼 전력 소비가 작으므로 정확한 비밀키 추측에 대하여 그 차분 신호는 구별된다.

3. 제안하는 DPA공격 적용

1) DES 에 대한 적용

DES에 대하여 제안하는 DPA공격은 마지막 라운드의 비밀키 K_{16}를 공격함으로써, 전체적인 DES의 56비트 비밀키를 알아낸다. 먼저, 임의의 평문 $P_i(1 \le i \le m)$에 대한 암호문 C_i와 전력 소비 신호 $T_i[j]$ 쌍을 실험을 통하여 구한다. 그리고, 비밀키 K_{16}의 상위 6비트부터 순차적으로 추측한다. 분류함수 $D(K_s, C_{i-1}, C_i, b)$의 b는 S-box의 출력값 4비트로 정의한다. 추측하는 비밀키 K_s가 K_{16}의 최상위 6비트이면, 분류 기준이 되는 데이터의 위치는 그림 2의 S-box1에 해당되는 출력값 4비트이다.

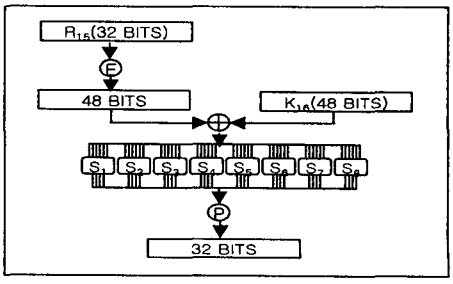

그림 2 : DES f 함수

공격자는 분류 함수를 이용하여 b에 해당되는 데이터의 상태천이 수에 따라 전력 소비 신호를 분류하고, 차분 신호를 구한다. 그리고, 데이터의 상태천이 수가 2인 경우, 해당되는 전력 소비 신호는 분류 대상에서 제외시킨다. 따라서, 공격에 필요한 전력 소비 신호의 수는 대략 $10m/16$이 된다. 6비트 단위로 비밀키를 추측하므로, 64개의 차분 신호가 생성된다. 만약, 차분 신호가 spike 신호를 가진다면, 공격자는 추측한 비밀키가 실제 비밀키와 동일하다고 판단할 수 있다.

2) Rijndael 에 대한 적용

Rijndael은 NIST에서 채택한 차세대 블록 암호 알고리듬이다. 그림 3은 Rijndael의 암호화 과정을 나타낸다[8].

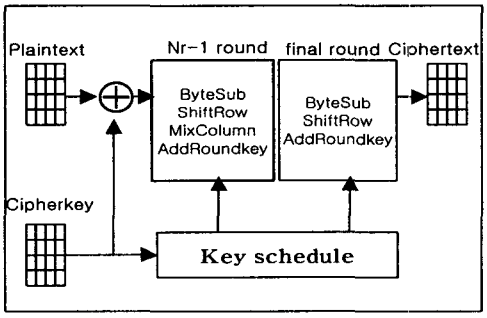

그림 3: Rijndael 암호화 과정

그림 3에서 알 수 있듯이 Rijndael은 initial round key addition 부분과 final round 부분에서 DPA공격 가능하다. 본 논문에서는 initial round key addition 부분에 대하여 제안하는 DPA 공격을 적용하며, 비밀키 길이와 평문의 블록 길이는 128비트라고 가정한다.

Rijndael에 있어서 분류 함수의 입력 중 하나인 b의 위치를 MixColumn연산 직후의 메모리에 저장되는 데이터로 정의한다. 먼저, 공격자는 비밀키 K의 최상위 8비트부터 추측한다. 그리고, 분류함수 $D(K_s, P_{i-1}, P_i, b)$를 이용하여 MixColumn연산 직후의 최상위 8비트 위치에 대하여 데이터 상태천이 수를 조사한다. 데이터 상태 천이 수에 따라 분류된 전력 소비 신호의 차분 신호는 256개가 존재하며, 추측한 비밀키가 옳을 경우, 차분 신호는 spike를 가지게 된다. 분류 대상이 되는 전력 소비 신호의 개수는 대략 $93m/128$이다.

4. 제안하는 DPA공격 분석

본 절에서는 제안하는 데이터 상태천이를 이용한 DPA공격에 대하여 S/N비의 측면에서 분석한다. S/N비의 계산은 다음의 표기 방식을 사용하여 표현한다.

표 1 : 표기와 의미

표 기	의 미
$T_i[j]$	전력 소비 신호, $1 \le i \le m$
j^*	분류되는 데이터가 반응하는 지점
σ^2	$T_1[j=k], \ldots, T_m[j=k]$의 분산
n	스마트카드의 데이터 처리 단위
d	분류기준이 되는 데이터의 비트수
a	잡음에 대한 데이터의 의존도[%]
$\Delta_D[j]$	차분 전력 소비 신호

그리고 다음과 같이 가정한다.

가정 1 : m개의 전력 소비 신호 $T_i[j]$에서 $T_1[j=k], \ldots, T_m[j=k]$는 가우시안 랜덤 변수의 성질을 가진다.

가정 2 : $|D_0|, |D_1|$는 $\frac{m}{2}$으로 동일하다.

가정 3 : 공격자가 추측한 비밀키는 실제 비밀키와 동일하다.

위와 같은 가정 아래, $j \neq j^*$ 일 때, D_0, D_1, Δ_D의 분산은 다음과 같다.

$$Var(\overline{D_0}[j \neq j^*]) \approx \frac{2\sigma^2}{m} \quad (4)$$

$$Var(\overline{D_1}[j \neq j^*]) \approx \frac{2\sigma^2}{m} \quad (5)$$

$$\therefore Var(\Delta_D[j \neq j^*]) = Var(\overline{D_1}[j] - \overline{D_0}[j]) \approx \frac{4\sigma^2}{m} \quad (6)$$

그리고, 스마트카드의 데이터 처리 단위가 n이고, hamming weight 1에 대한 $\Delta_D[j=j^*]$의 크기가 ε이라면, S/N비는 다음과 같이 표현된다[9].

■ 잡음 신호:

$$E[\Delta_D[j] | (j \neq j^*)] = 0 \quad (7)$$

$$Var[\Delta_D[j] | (j \neq j^*)] = \frac{4\sigma^2 + a \cdot n \cdot \varepsilon^2}{m} \quad (8)$$

■ spike 신호:

$$E[\Delta_D[j] | (j = j^*)] = \varepsilon \quad (9)$$

$$Var[\Delta_D[j] | (j = j^*)] = \frac{4\sigma^2 + (n-1) \cdot \varepsilon^2}{m} \quad (10)$$

$$SNR=\frac{\sqrt{m}\cdot\varepsilon}{\sqrt{8\sigma^2+\varepsilon^2(a\cdot n+n-1)}} \quad (11)$$

m개의 전력 소비 신호에 대하여 차분 신호 $\Delta_D[j]$의 S/N비를 향상시키기 위해서 분류되는 데이터를 다중 비트 d로할 경우, S/N비는 다음과 같다.

$$SNR=\frac{\sqrt{m'}\cdot\beta\cdot\varepsilon}{\sqrt{8\sigma^2+\varepsilon^2(a\cdot n+n-d)}} \quad (12)$$

식 12에서 m'는 실제로 분류되는 전력 소비 신호의 개수이고, β는 d의 값에 따른 ε의 가중치이다. T를 데이터 상태 천이 수라고 할 때, 제안하는 DPA공격에서 β는 다음과 같이 구할 수 있다

$$d=4:\ \langle T\rangle_{4,3}-\langle T\rangle_{0,1}=16/5-4/5=12/5 \quad (13)$$

$$d=8:\ \langle T\rangle_{8,7,6,5}-\langle T\rangle_{0,1,2,3}=$$
$$(1/93\cdot 8+8/93\cdot 7+28/93\cdot 6+56/93\cdot 5)-$$
$$(1/93\cdot 0+8/93\cdot 1+28/93\cdot 2+56/93\cdot 3)$$
$$=280/93 \quad (14)$$

표 2는 기존의 DPA공격과 제안한 DPA공격에 대한 m'와 β를 나타낸다.

표 2 : d에 따른 m'와 β의 값

	기존 DPA 공격		제안 DPA 공격	
	m'	β	m'	β
d=1	m	1	m	1
d=4	$m/8$	4	$5m/8$	12/5
d=8	$m/128$	8	$93m/128$	280/93

표 2의 값을 식 12에 적용하여, 기존 DPA공격과 제안한 DPA공격에 대한 S/N비를 표 3에 나타내었다.

표 3 : 신호 대 잡음 비
($m=1000$, $n=8, a\approx 0$, $\sigma^2=7.5mV$, $\varepsilon=6.5mV$)

SNR	기존 DPA 공격	제안 DPA 공격
d=1	7.52	7.52
d=4	11.68	15.7
d=8	6.84	25.59

σ^2과 ε의 값은 문헌 [9]를 참고하였다. 기존의 DPA공격인 경우, 분류되는 데이터의 비트 수가 증가한다고 해서 반드시 차분 신호의 S/N비가 증가하는 것은 아니다. 제안하는 DPA공격은 기존의 DPA공격에 비해 S/N비 측면에서 훨씬 효율적이다. 이와 같은 결과는 공격자로 하여금 적은 양의 데이터로 보다 효율적인 공격을 할 수 있게 할 것이다.

Ⅳ.결론

오늘날 그 응용 분야가 점점 확대되어 지고 있는 스마트카드는 전력 분석 공격과 같은 새로운 형태의 물리적 공격에 보안상 결함을 갖고 있다. 본 논문에서는 기존의 DPA공격과 스마트카드의 전력 소비 형태를 분석하고, 데이터 상태천이를 이용한 S/N를 향상시킨 DPA공격을 제안하였다.

제안한 DPA공격은 스마트카드의 전력 소비 신호와 비밀키의 상관 관계를 더욱 강화시킴으로써, 차분 전력 신호의 향상된 S/N비를 이론적으로 계산하였다. 스마트카드는 이러한 DPA공격에 대해 보안을 고려하여 설계 및 구현되어야 할 것이다.

참고문헌

[1] Paul Kocher, "Timing Attacks on Imple-mentations of Diffie-Hellman, RSA, DSS, and Other Systems," in *CRYPTO '96*, 1996.

[2] D. Boneh,"On the Importance of Checking Cryptographic Protocols for Faults," in *Cryptology-Eurocrypt '97*, 1997.

[3] Paul Kocher,"Differential Power analysis,"in proceedings of *CRYPTO'99*, 1999.

[4] Thamas S. Messerges," Power Analysis Attacks of Modular Exponentiation in Smartcards," in *CHES'99*, 1999.

[5] J. S. Coron, "Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems," in *CHES'99*, 1999.

[6] P.-Y. Liardet ," Preventing SPA/DPA in ECC systems using the Jacobi Form," *CHES'01*.

[7] N. Weste, K. Eshraghian, Principles of CMOS VLSI design, Addison-Wesley, 1993.

[8] Joan Daemen and Vincent Rijmen, "Rijndael Block Cipher," 1999.

[9] T. S. Messerges,"Investigations of Power Analysis Attacks on Smartcards," in *USENIX*, 1999.