情報保護學會論文誌
第13券 第3號, 2003. 6

타원곡선 암호시스템에서 랜덤 m-ary 방법을 사용한 전력분석 공격의 대응방법*

안 만 기**, 하 재 철***, 이 훈 재****, 문 상 재*****

A Random M-ary Method-Based Countermeasure against Power Analysis Attacks on ECC

MahnKi Ahn**, JaeCheol Ha***, HoonJae Lee****, SangJae Moon*****

요 약

타원곡선 암호시스템에서 스칼라 곱셈의 랜덤화는 부-채널공격 대응방법의 기본적인 개념 중의 하나이다. 본 논문에서는 랜덤 m-ary 리코딩 알고리듬에 기반한 랜덤 m-ary 방법으로 단순/차분 전력분석 공격의 대응 방법을 제안한다. 제안 방법은 표준의 m-ary 방법보다 부가적인 연산량이 요구되지만 비밀키와 독립적인 소모전력을 생성한다. 따라서 랜덤한 윈도우 사이즈를 이용한 연산 과정이 SPA/DPA 공격에 대응할 수 있으므로 제안하는 대응방법은 스마트카드의 부-채널공격에 향상된 안전성을 제공한다.

ABSTRACT

The randomization of scalar multiplication in ECC is one of the fundamental concepts in defense methods against side-channel attacks. This paper proposes a countermeasure against simple and differential power analysis attacks through randomizing the transformed m-ary method based on a random m-ary recoding algorithm. The proposed method requires an additional computational load compared to the standard m-ary method, yet the power consumption is independent of the secret key. Accordingly, since computational tracks using random window width can resist against SPA and DPA, the proposed countermeasure can improve the security for smart cards.

keyword : *smartcard, side channel attacks, power analysis attacks,* SPA/DPA, *random m-ary recoding algorithm*

Ⅰ. 서 론

스마트카드(smartcard)란 마이크로프로세서와 메모리를 내장하고 카드 내에서 데이터 연산 처리와 저장이 가능한 플라스틱 카드를 말한다. 특히, 접촉식 카드는 칩의 동작을 위한 전원과 클럭 신호를 얻기 위하여 리더기와 물리적인 접촉이 필요하다. 따라서 암호 알고리듬 설계 단계에서 고려되지 못한 부가적인 정보 누출로부터 비밀 정보를 알아내는 부-채널 공격(side-channel attack)의 대상이 될 수 있다. 이러한 부-채널공격은 시차 공격(timing attack)[1], 오류주입 공격(fault insertion attack)[2], 전력분석 공격(power

* 본 연구는 대학 IT 연구센터 육성·지원 사업의 연구 결과로 수행되었습니다.
** 국방품질관리소 연구원(mkahn@dqaa.go.kr)
*** 나사렛대학교 정보과학부 교수(jcha@kornu.ac.kr)
**** 동서대학교 인터넷공학부 교수(hjlee@dongseo.ac.kr)
***** 경북대학교 전자전기컴퓨터학부 교수(sjmoon@knu.ac.kr)

analysis attack)[3] 그리고 전자기 누출 공격(electromagnetic emission attack)[4]등으로 나눌 수 있다. 전력분석 공격은 스마트카드에 물리적 변환을 가하지 않고 직접 소모전력 신호의 특성을 파악하여 비밀키에 대한 정보를 알아내는 SPA(simple power analysis)와 SPA에 통계적인 분석 방법 및 에러 정정 기술을 도입한 DPA(differential power analysis)로 분류된다.

T. S. Messerges는 RSA 멱승 알고리듬에 대하여 Single exponent multiple data(SEMD) 공격, Multiple exponent single data(MESD) 공격, 그리고 Zero exponent multiple data(ZEMD) 공격 등을 적용하였다.[5] RSA 암호 시스템[6]과 타원곡선 암호 시스템(Elliptic Curve Cryptosystem, ECC)[7]에서 MESD 공격은 두 개의 스마트카드(공격대상 카드와 키 값을 변경할 수 있는 비교용 카드)를 이용하여 공격하는 기술로서, 동일한 메시지를 적용하여 비밀키를 모르는 공격대상카드의 평균전력 파형과 비밀키 변경이 가능한 비교용 카드의 평균전력 파형을 차분함으로써 키 비트를 순차적으로 알아내는 방법이다.

본 논문에서는 타원곡선 암호시스템이 소프트웨어적으로 내장된 스마트카드에서 MESD공격이 가능함을 검증하고 전력분석 공격에 대한 대응방법을 제시한다. 타원곡선 상의 임의의 한 점에 대한 스칼라 곱셈 연산에 대한 전력분석 공격을 적용한 후, 이에 대한 대응방법으로 랜덤 m-진 방법을 사용한 스칼라 곱셈 알고리듬을 제안하고자 한다. 또한 제안하는 리코딩(recoding)알고리듬, 스칼라 곱셈 알고리듬의 연산 과정, 안전성, 연산량 등을 분석하고 MESD공격을 실험적으로 분석하고자 한다.

II. 스칼라 곱셈에 대한 전력분석 공격 실험

2.1 두배-덧셈 알고리듬

타원곡선 암호 알고리듬은 타원곡선 상의 한 점 P에 대해서 $Q=d\cdot P$를 계산하는 비밀키 d에 대한 스칼라 곱셈 연산을 행한다. 일반적으로 스칼라 곱셈 연산은 구현이 용이한 두배-덧셈 알고리듬을 사용한다. 따라서 공격자는 [그림 1]와 같은 left-to-right(LR) 방법의 두배-덧셈 알고리듬이 수행될 때, 소비되는 전력을 조사함으로써 전력 공격을 적용할 수 있다. 공격자는 비밀키를 제외한 나머지 공개 정보는 알고 있다고 가정한다.

```
INPUT : P, d=(d_{l-1}, ..., d_0)_2
OUTPUT : Q= d · P.
    Q = P
    for i from l-2 downto 0 do
      Q = 2 · Q
      if (d_i == 1) then Q = Q+P
    return Q
```

〔그림 1〕 스칼라 곱셈 알고리듬

공격 대상의 비밀키를 한 비트씩 추측하는 과정은 비밀키의 두 번째 비트를 '0'으로 가정하고 점 P가 두배 연산이 된 후 소모전력을 측정한다. 이때 한 점을 두배하는 연산($2Q$)과 두 점을 더하는 연산($P+Q$)의 소모전력이 다르게 나타날 것이다. 그러나 실제 비밀 키 비트가 "1"이라면 조건문을 수행할 때 Q가 $2Q+P$가 되어 소모전력의 차이로 SPA 공격이 가능하다. 또한 두배와 덧셈의 수행 시간 차이로 시차공격도 가능하다.

암호시스템 설계자는 랜덤 수를 생성하여 잡음을 첨가하거나 부가(dummy) 명령어를 삽입하여 쉽게 SPA 공격을 방어할 수 있다. Coron은 [그림 2]과 같은 방법을 제안했다.[8] 하지만 이 방법은 항상 두배와 덧셈 연산을 실시하기 때문에 연산량의 증가를 가져오며, 연산되는 중간값들의 변화에 따른 DPA 공격에는 대응할 수 없다.

```
INPUT : P, d=(d_{l-1}, ..., d_0)_2
OUTPUT : Q[0]= dP.
    Q[0] = P
    for i from l-2 downto 0 do
        Q[0] = 2Q[0]        //Doubling
        Q[1] = Q[0] + P     //Addition
        Q[0] = Q[ d_i]      //Selection
    return Q[0]
```

〔그림 2〕 SPA에 대응하는 두배-덧셈 알고리듬

2.2 Coron의 방법에 대한 MESD 공격

비밀키 $d=(d_{l-1},d_{l-2},\ldots,d_0)$라 할 때, d_{l-1}는 최상위 비트이며 '1'로 가정한다. 공격자는 공격 대상의 스마트카드에서 소모전력 $S_i(t)$를 수집한다. 마찬가지로 키 값을 변경할 수 있는 비교용 스마트카드에서 소모전력 $T_i(t)$를 수집한다. 위의 두 소모전력 파형들을 평균하고 차분한다.

$$\triangle(t)=\frac{1}{h}\left(\sum_{i=0}^{h-1}S_i(t)-\sum_{i=0}^{h-1}T_i(t)\right) \quad (1)$$

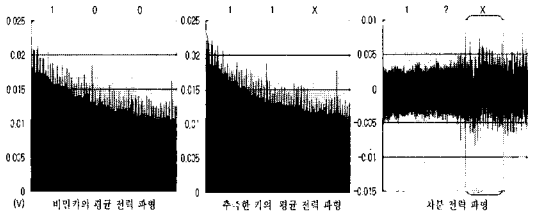
여기서 h는 소모전력 측정회수이다. 공격자는 $\triangle(t)$에서 피크(peak)를 확인하여 추측의 정확성을 판단한다. 추측한 비트의 다음 비트 위치부터 피크를 확인함으로써 정확한 비트를 알 수 있다.

스마트카드에 사용된 ECC가 $GF(p)$상에서 비밀키 $d=10010\cdots\cdots_{(2)}$로 설정할 때, 한 비트를 추측하기 위해 $Q_0=dP_0$에서 $Q_{h-1}=dP_{h-1}$까지 h회를 실시한다. 이때 점 P_i는 스마트카드에서 고정된 점이다. [표 1]에서 두 번째 비트 d_{l-2}을 추측한다고 가정하면 실제 비트와 추측한 비트가 동일하면 전력 파형에는 피크가 나타나지 않는다. 만약 실제 비트와 추측한 비트가 d_{l-2}에서 다르다면 두 번째 비트의 선택(selection) 단계가 끝나고 세 번째 비트와 관련한 연산을 수행할 때부터 전력 파형이 달라진다. 따라서 세 번째 비트에 해당하는 연산을 수행할 때부터 피크 값을 볼 수 있다.

〔표 1〕 스칼라 곱셈 알고리듬에 의한 연산되는 과정
(X는 예상할 수 없는 값)

비밀키	Bit	1	0	0	1	…
	Doubling		2P	4P	8P	…
	Addition				9P	…
	Selection		2P	4P	9P	…
추측키	Bit	1	1	X	X	…
	Doubling		2P	6P	·	…
	Addition		3P	7P	·	…
	Selection		3P	X	·	…

[그림 3]은 측정회수 h=300으로 측정한 두 소모전력 파형을 평균하고 차분한 결과이다. 실험 결과에서 잘못 예측된 경우 세 번째 비트부터 피크가 생성되어 추측한 두 번째 비트 값이 '1'이 아님을 알 수 있다. 따라서 공격자는 비교용 스마트카드의 두 번째 비트를 수정한 후, 나머지 비트에 대해서도 반복적인 공격을 실시한다.



〔그림 3〕 SPA 대응하는 두배-덧셈 과정의 소모전력 파형

Ⅲ. ECC에서 전력분석 공격의 새로운 대응방법 제안

3.1 제안하는 리코딩 알고리듬(recoding algorithm)

두배-덧셈 방법은 n-비트의 비밀키 d를 이진수로 표현하여 스칼라 곱셈 연산을 수행한다.

$$d=\sum_{i=0}^{n-1}d_i\cdot 2^i,\ d_i\in\{0,1\} \quad (2)$$

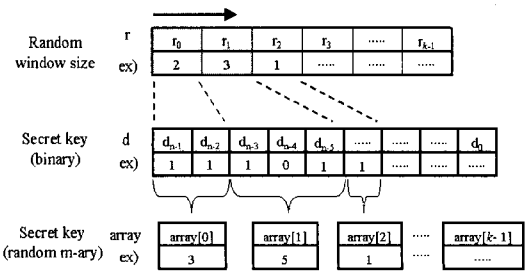
이러한 비밀키를 LR 방식의 m-진수 ($m=2^r$)로 확장한다. 이때 랜덤 윈도우 사이즈(window size) r은 1, 2, 3 중에서 선택되어진다. k는 생성된 배열쌍의 길이로 항상 가변적이다. [그림 4]는 비밀키를 새로운 배열로 변형하는 과정을 보여준다. 각각의 r_i와 $array[i]$는 하나의 배열쌍(array pairs)으로 구성된다. r_i는 랜덤 윈도우 사이즈의 값이며, $array[i]$는 윈도우 사이즈가 정해질 때 해당 비트열의 값으로 사전 계산된 점 $P[array[i]]$를 저장하고 읽을 경우에 사용된다. 보조인자 c_i는 SPA에 대응하는 알고리듬을 설계하기 위해 사용되는데 점 P의 값이 '0'이면 '0'의 값을 가지며 그 외에는 '1'의 값을 가진다.

$$r=\sum_{i=0}^{k-1}r_i\cdot 10^i,\ r_i\in\{1,2,3\}. \quad (3)$$

$$array=\sum_{i=0}^{k-1}array[i]\cdot 10^i, \quad (4)$$

$$array[i]\in\{0,1,2,3,4,5,6,7\}.$$

$$c=\sum_{i=0}^{k-1}c_i\cdot 2^i,\quad c_i\in\{0,1\} \quad (5)$$



〔그림 4〕 새로운 배열쌍의 구성

제안하는 리코딩 알고리듬은 [표 2]와 같은 비트

열을 생성한다. 연속적인 '0'이 생성되더라도 각각의 비트에 대해 m-진 방식을 사용한다. 각각의 윈도우 사이즈는 랜덤수 생성기에 의해 선택되어진다. 그러나 선택한 윈도우 사이즈가 남은 비밀키의 비트열보다 클 경우에는 윈도우 사이즈를 '1'로 선택하여 만들 수 있다. 비트열의 경우가 다르므로 각각의 발생 확률은 다르게 설계한다. 즉, r=1은 2가지의 비트열이 있는 반면, r=3일 때는 8가지의 비트열이 있어 랜덤수가 동일한 발생 확률을 가질 경우에는 최소 윈도우 사이즈인 '0'과 '1'이 상대적으로 많이 생성되게 된다. 이는 스칼라 곱셈 알고리듬을 설계할 때 안전성과 연산량 측면에서 비효율적이다. 따라서 r을 효과적으로 선택하기 위하여 [그림 5]과 같은 의사코드(Pseudo-code)의 알고리듬을 구성한다. 입력되는 x는 임의 양의 정수이다. 따라서 랜덤 윈도우 사이즈의 발생확률은 (6)식과 같으며 이는 배열쌍의 길이 k에 영향을 주며 각각의 비트열들이 선택될 경우가 비슷해진다.

$$P_{(r=1)}=\frac{1}{6},\ P_{(r=2)}=\frac{1}{3},\ P_{(r=3)}=\frac{1}{2} \qquad (6)$$

따라서 r=1일 때 '0'의 비트열이 발생할 확률은 $\frac{1}{6}\times\frac{1}{2}=\frac{1}{12}$이며 r=3일 때 '111'의 비트열이 발생할 확률은 $\frac{1}{2}\times\frac{1}{8}=\frac{1}{16}$이다.

〔표 2〕 랜덤한 m-ary 방법에 따른 비트열의 분류

윈도우 사이즈	r = 1 (2진 방법)	r = 2 (4진 방법)	r = 3 (8진 방법)
비트열	0, 1	00, 01, 10, 11	000, 001, 010, 011, 100, 101, 110, 111
P의 값	0, 1	0, 1, 2, 3	0, 1, 2, 3, 4, 5, 6, 7

```
INPUT : The positive integer x.
OUTPUT : The random window size r.
Begin
    Temp = random( x) mod 6
    If (Temp >= 3) then r = 3
    Else if (Temp >= 1) then r = 2
    Else r = 1
End
```

〔그림 5〕 랜덤 윈도우 사이즈를 발생하는 알고리듬

3.2 제안하는 스칼라 곱셈 알고리듬

3.2.1 DPA에 대응하는 알고리듬

타원곡선 암호시스템에서 기존의 m-진 방법은 점 $P[j]$을 사전계산하고 임의 메모리에 저장한 후, 점 Q를 mQ하는 연산과 사전 계산된 점 $P[j]$를 덧셈하는 연산을 반복한다. 그러나 제안하는 새로운 스칼라 곱셈 알고리듬은 동일하게 사전계산을 하지만 연산하는 과정이 다르다. 즉, 이진 방식의 두배 연산을 윈도우 사이즈만큼 수행하여 두배-덧셈 연산을 실행하도록 [그림 6]과 같이 구성하였다. 배열쌍을 생성하는 리코딩 수행시간은 스칼라 곱셈 연산 시간에 비해 무시해도 될 정도의 빠른 시간이므로 연산량 측면에서는 고려하지 않는다. 그러나 사전 연산 과정에서 실시한 두배-덧셈의 연산량(3D+3A)은 고려해야 한다. 배열쌍은 스마트카드가 실행될 때마다 랜덤하게 생성되므로 DPA 공격에 대응할 수 있으며, 수행시간도 배열쌍의 개수에 따라 유동적이다. 그러나 [그림 6]의 알고리듬은 공격자가 두배 연산과 덧셈 연산의 소모전력 파형을 구분할 수 있으면 SPA 공격에는 대응할 수 없다.

```
INPUT : The point P, r, array[ k].
OUTPUT : Q = dP.
/Precomputation phase/
P[1]=P
for j=2 to 6 by 2 {
   P[j]=2P[j/2]
   P[j+1]=P[j-1]+P[2] }
/Evaluation phase/
Q = ∅ where ∅ is the point at infinity.
for i = 0 to k-1 do {
   Q = 2Q                 /* 기존의 mQ 연산을
   if( r_i >=2) Q = 2Q      수행
   if( r_i ==3) Q = 2Q   */
   if(array[i] ≠0)Q=Q+P[array[i]] }
return Q
```

〔그림 6〕 DPA에 대응하는 스칼라 곱셈 알고리듬

3.2.2 SPA/DPA에 대응하는 알고리듬

SPA/DPA에 대응하는 알고리듬은 조건문이 없이 항상 세 번의 두배 연산과 한번의 덧셈 연산을 실시하도록 [그림 7]과 같이 구성한다. 두배 연산과 덧셈 연산 과정의 소모전력이 구분되어도 SPA 공격을 방어할 수 있다. Q[1], Q[2] 그리고 Q[3]은 무조건 두배

씩 연산된 값을 가진다. 덧셈 연산은 array[i]가 '0'일 때도 항상 수행되지만 최종적인 결과 값은 Q[0] = Q[c_i+ r_i]에서 선택되어진다. 즉, c_i이 '0'일 때는 덧셈 연산된 값이 선택되지 않고 두배 연산을 한 결과만 선택한다. 이러한 방법은 비밀키 열과 무관하게 항상 "Doubling, Doubling, Doubling, Addition (DDDA)"의 순으로 연산하여 연산량이 약간 증가하지만 SPA 공격에도 방어할 수 있다.

```
INPUT : The point P, r, c, array[ k].
OUTPUT : Q[0] = dP.
{Precomputation phase}
P[1]=P
for j=2 to 6 by 2 {
   P[j]= 2P[j/2]
   P[j+1]=P[j-1]+P[2] }
{Evaluation phase}
Q[0] = ∅, where ∅ is the point at infinity.
for i = 0 to k-1 do {
   Q[1] = 2Q[0]   /* 기존의 mQ 연산을
   Q[2] = 2Q[1]     윈도우 사이즈에 관계없이
   Q[3] = 2Q[2]     항상 두배 연산으로 수행 */
   Q[1+ r_i] = Q[ r_i] + P[ array[i] ]
   Q[0] = Q[ c_i+ r_i]  }
return Q[0]
```

〔그림 7〕 SPA/DPA에 대응하는 스칼라 곱셈 알고리듬

Ⅳ. 제안 방법 분석

제안된 리코딩 알고리듬에서 배열쌍의 생성 과정을 살펴하고, 기존의 스칼라 곱셈 알고리듬의 연산 과정과 비교하여 제안된 스칼라 곱셈 알고리듬을 분석한다. 그리고 전력분석 공격에 대한 대응방법의 안전성과 연산량을 분석한다. 마지막으로 MESD 공격의 대응 여부를 실험적으로 검증한다.

4.1 리코딩 알고리듬의 분석

n-비트 이진 비밀키가 $[0, 2^{n-1}]$의 범위 내에서 이진 랜덤수 생성기에 의해 '0'과 '1'로 생성될 때 확률 $P(d_i=0)=P(d_i=1)=\frac{1}{2}$을 가진다고 가정한다. 실험을 위하여 하나의 비밀키마다 10,000회씩 반복하여 새로운 배열쌍을 생성하였다. 10,000개의 경우에 대해 생성된 배열쌍의 분포는 [표 3]과 같았다.

〔표 3〕 160비트의 비밀키에 대한 새로운 배열쌍의 개수

배열쌍의 개수	생성회수	배열쌍의 개수	생성회수
58	0	71	1162
59	0	72	830
60	1	73	525
61	6	74	322
62	21	75	158
63	78	76	73
64	201	77	35
65	417	78	14
66	756	79	6
67	1113	80	2
68	1379	81	1
69	**1492**	82	0
70	1408	83	0

결론적으로 160비트의 비밀키들은 평균 69개의 배열쌍이 가장 많이 생성되는 것을 표에서 볼 수 있다. 이러한 실험은 정확도를 높이기 위해 총 100개의 서로 다른 비밀키를 선택하여 10,000번씩 시도하여 이를 평균한 것이다. 또한 비밀키 길이 n을 변화시키면서 평균 배열쌍의 개수를 조사하였는데 [표 4]와 같은 결과를 가지는 것을 실험을 통하여 확인하였다.

〔표 4〕 비밀키의 비트수에 따른 배열쌍의 생성 비율

비밀키의 비트수 (n)	가장 많이 생성된 배열쌍의 개수	전체 배열쌍에 대한 비율	전체 50%를 차지하는 배열쌍의 개수
128	55	16%	53 ~ 56
160	**69**	**15%**	**67 ~ 70**
192	83	13%	81 ~ 84
224	96	12%	94 ~ 98

4.2 스칼라 곱셈 알고리듬의 분석

4.2.1 연산 과정 분석

제안된 랜덤 m-진 방법의 스칼라 곱셈 알고리듬의 연산 과정을 이진 스칼라 곱셈 알고리듬과 비교하면 [표 5]와 같다. SPA 공격에 대응할 때 이진 알고리듬은 비트열에 따라 항상 "Doubling, Addition (DA)"를 실시하지만 제안된 알고리듬은 비트열에 무관하게 항상 동일한 "DDDA"를 실시하므로 SPA/DPA 공격으로 정확한 비트열을 얻기 힘들다.

〔표 5〕 비트열에 따른 연산과정의 비교

비트열	배열쌍	이진 스칼라 곱셈 알고리듬		랜덤 m-ary 방법의 스칼라 곱셈 알고리듬	
		[그림 1]	[그림 2]	[그림 6]	[그림 7]
		SPA에 대응하지 않을 때	SPA에 대응할 때	DPA만 대응할 때	SPA/DPA에 대응할 때
0	(1,0)	D	DA	D	DDDA
1	(1,1)	DA	DA	DA	DDDA
00	(2,0)	DD	DADA	DD	DDDA
01	(2,1)	DDA	DADA	DDA	DDDA
10	(2,2)	DAD	DADA	DDA	DDDA
11	(2,3)	DADA	DADA	DDA	DDDA
000	(3,0)	DDD	DADADA	DDD	DDDA
001	(3,1)	DDDA	DADADA	DDDA	DDDA
010	(3,2)	DDAD	DADADA	DDDA	DDDA
011	(3,3)	DDADA	DADADA	DDDA	DDDA
100	(3,4)	DADD	DADADA	DDDA	DDDA
101	(3,5)	DADDA	DADADA	DDDA	DDDA
110	(3,6)	DADAD	DADADA	DDDA	DDDA
111	(3,7)	DADADA	DADADA	DDDA	DDDA

예를 들어, 비밀키 d=1011001101$_{(2)}$일 때 리코딩 알고리듬에 의해 연산 과정의 순서는 다양하다. 즉, 스칼라 곱셈 알고리듬이 최상위 비트(MSB)를 제외한 두 번째 비트부터 연산을 실시하면 최소 3개부터 최대 9개의 배열쌍이 나올 수 있다. 만약 5개의 배열쌍에서 윈도우 사이즈의 값이 [표 6]과 같을 때 생성될 연산 순서의 경우는 45가지이다. 따라서 3개와 9개의 배열쌍은 한번만 생성되지만 기타 다른 경우는 다양하게 분포되어 공격자는 비트열의 연산 순서를 추측하기 힘들다. 스마트카드가 실행될 때마다 리코딩 알고리듬에 의해 배열쌍의 생성 개수와 생성된 배열쌍의 연산 순서도 다양하다.

〔표 6〕 생성된 배열쌍이 5개일 경우의 연산 순서

r의 종류	배열쌍의 연산 순서 (r , array[i])	경우의 수
1,1,1,3,3	(1,0) (1,1) (1,1) (3,1) (3,5) (3,3) (1,0) (3,3) (1,0) (1,1) ...	$\frac{5!}{3!2!}=10$
1,1,2,2,3	(1,0) (1,1) (2,2) (2,1) (3,5) (1,0) (2,3) (1,0) (3,3) (2,1) ...	$\frac{5!}{2!2!}=30$
1,2,2,2,2	(1,0) (2,3) (2,0) (2,3) (2,1) (2,1) (1,1) (2,0) (2,3) (2,1) ...	$\frac{5!}{4!}=5$

4.2.2 안전성 분석

비밀키의 리코딩과 스칼라 곱셈 과정에서 비트에 대한 정보를 누출할 가능성이 있는지 분석하도록 한다. 다음은 부-채널 공격의 안전성에 요구되는 두 가지 조건으로 Okeya와 Sakurai가 제안하였다.[15]

1. 비밀정보와 연산 과정의 독립성.
2. 연산되는 정보의 표현과정에서 랜덤성.

요구사항 1은 SPA 공격에 대응하는 방법으로 간단하게 구현될 수 있다. 따라서 비밀키와 관련된 연산 과정이 항상 일정하고 독립적으로 동작하는 제안된 [그림 7]의 알고리듬은 요구사항 1을 만족한다. 또한 연산하는 대상의 표현과정에서 랜덤성이 요구된다. 이는 비밀정보와 관련된 과정이 서로 무관하게 동작하여도 연산하는 대상이 랜덤성이 없으면 DPA 공격에 대응할 수 없다. 이러한 요구사항 2도 랜덤한 m-ary 리코딩 알고리듬을 이용하여 만족시킬 수 있다. 비밀키를 랜덤한 새로운 배열쌍으로 생성하여 비밀키를 추측하기 어렵도록 한다. 따라서 정확한 비트나 비트열을 차분 전력분석 공격으로 얻고자 하여도 두 소모전력의 차분 파형에서 랜덤한 피크가 모든 영역에 나타난다.

4.2.3 연산량 분석

제안된 스칼라 곱셈 방법의 연산량을 아핀 좌표계에서 이진 방법과 윈도우 방법의 연산량과 비교한다. DPA에만 대응하는 [그림 6]의 알고리듬은 배열쌍 중에서 array[i]가 '0'이면 덧셈 연산을 실시하지 않는다.

$$P(array[i]\neq 0)=\frac{1}{6}\times\frac{1}{2}+\frac{1}{3}\times\frac{3}{4}+\frac{1}{2}\times\frac{7}{8}=\frac{37}{48}$$

따라서 사전계산의 3A와 3D를 포함한 연산량은 (7)식과 같다.

$$\left[\frac{69}{160}\times\frac{37}{48}\times n+3\right]A+[n+3]D=[0.33n+3]A+[1.0n+3]D \quad (7)$$

그러나 SPA/DPA에 대응하는 [그림 7]의 스칼라 곱셈 알고리듬을 사용하면 array[i]의 값에 무관하게 세 번의 두배 연산과 한번의 덧셈 연산을 항상 실시

하기 때문에 연산량이 약간 증가한다.

$$\left[\frac{69}{160}\times n+3\right]A+\left[\frac{69}{160}\times 3\times n+3\right]D = [0.43n+3]A+[1.29n+3]D \quad (8)$$

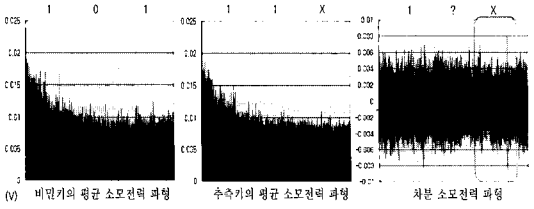
〔표 7〕 제안된 대응방법과 대응여부에 따른 연산량
(○ :대응함, △ :공격 가능성이 있음, × :대응하지 못함)

Algorithm	n-bit	SPA 대응 여부	DPA 대응 여부	연산량		덧셈 연산의 시간 비교 (D=0.7A)	Ref.
				Additions (subtractions)	Doublings		
Ordinary binary method	160	×	×	0.5n	n	1.2n	
Coron's method	160	○	×	n	n	1.7n	[8]
Hitchcock-Montague's method	160	×	×	0.33n	n	1.03n	[9]
		○	×	0.56n	1.11n	1.34n	
Ha-Moon's method	160	×	○	0.5n	n+1	1.2n+0.7	[10]
		○	○	n+1	n+1	1.7n+1.7	
Möller's method (min. window)	160	○	×	0.5n	n	1.2n	[11]
Randomized signed window method	192	×	○	0.2n	n-1	0.9n-0.7	[12] [13]
Window method (O-WM : 4-ary)	160	○	△	0.5n	n+16	1.2n+11.2	[14]
Proposed Random m-ary method (2, 4, 8-ary)	평균 69개의 배열 쌍	×	○	0.33n+3	n+3	1.03n+4.1	
		○	○	0.43n+3	1.29n+3	1.33n+4.1	

[표 7]은 D=0.7A[17]일 때 제안된 랜덤 m-진 방법과 기존의 다른 대응방법을 보여주고 있다. 160 비트의 비밀키를 NAF(non-adjacent form)로 구성하고 새로운 심벌(symbol)로 만들어 스칼라 곱셈하는 알고리듬[9]와 비교할 때 랜덤 m-진 방법은 DPA 공격에 대응하면서도 비슷한 연산량을 가진다. 또한 비밀키를 랜덤하게 새로운 NAF 형태로 구성하고 스칼라 곱셈하는 Ha-Moon's method [10]보다 DPA만 대응하는 방법은 연산량이 12%정도 감소하며, SPA /DPA에 대응하는 방법은 21%정도 감소한다. 또한 [11]의 알고리듬에 비해 연산량은 많지만 [11]은 2차 DPA 공격이 가능하며, 1차 DPA 공격 가능성에 대해서도 논란의 여지가 있다.[16] 그리고 부호화된 m-진 윈도우 재배열 리코딩과 부호화된 m-진 윈도우 방식의 알고리듬[12][13]에 비해 제안하는 방법은 SPA에도 대응할 수 있다. O-WM(overlapping window method)[14]과 비교할 때 [14]의 방법은 세 배 정도의 사전 연산을 실시하여 메모리를 많이 차지하며 고정된 윈도우를 사용할 경우 DPA 공격의 가능성이 있다. 이와같이 제안된 방법은 변형된 스칼라 곱셈으로 SPA 공격을 방어하며 랜덤 리코딩 방법으로 DPA 공격에도 대응할 수 있어 기존의 대응방법과 비교하여 효과적인 것을 볼 수 있다. 그러나 사전계산을 위한 메모리가 증가하게 된다. 이는 사전연산을 하지 않는 알고리듬에 비해 8개 정도의 점(point)을 저장할 레지스터를 더 사용한다.

4.2.4 제안된 SPA/DPA에 대응하는 알고리듬의 MESD 공격 분석

SPA의 대응방법은 두배 연산과 덧셈 연산이 서로 다른 소모전력 파형을 가진다면 연산 과정을 항상 동일하게 실행하여 공격자가 비밀키의 비트열을 추측하지 못하도록 하는 것이다. 그러나 이러한 대응책은 DPA 공격에 대응할 수 없다. 따라서 DPA에 대응방안으로 스마트카드의 암호 알고리듬이 실행될 때마다 제안된 리코딩 알고리듬을 사용하여 소모전력 파형을 랜덤하게 분포시키는 것이다. [그림 8]은 $GF(p)$ 상에서 비밀키가 랜덤하게 재배열되어질 때 공격자가 측정회수 h=300으로 설정하고 MESD 공격을 실시한 파형이다. 실험 조건은 비밀키를 제외하고 2.2절에서 언급한 내용과 같다. 제안된 SPA/DPA에 대응하는 스칼라 곱셈 알고리듬과 리코딩 알고리듬을 적용할 때 두 번째 비트를 추측해도 [그림 3]의 차분 소모전력 파형과 차이가 있음을 알 수 있다. 즉, 공



〔그림 8〕 제안된 SPA/DPA에 대응하는 스칼라 곱셈 알고리듬의 소모전력 파형

격자가 예상하는 비트의 위치 외에서도 피크가 생성되지 않아 공격자는 정확한 비밀키의 비트를 추측할 수 없으며 다시 소모전력을 재구성하여 분류해도 연관성이 없는 소모전력 파형이 차분되어 정확하게 비밀키를 추측할 수 없다. 이는 배열쌍이 랜덤하게 생성되고 비트열 '0'에서 '111'까지 항상 "DDDA"의 연산을 수행하기 때문이다.

V. 결 론

본 논문에서는 부-채널 공격 중에서 가장 핵심이 되는 전력분석 공격의 대응방법으로 랜덤 윈도우를 사용하여 비밀키를 새로운 배열쌍으로 구성하는 리코딩 알고리듬과 항상 규칙적인 스칼라 곱셈 연산을 실시하는 방법을 제안하였다. 기존의 이진 방법을 사용하는 대응방법과 비교할 때, 사전 계산을 위한 메모리가 약간 증가하지만 연산량은 감소하며 규칙적인 연산 과정을 통해 SPA를 방어하도록 설계하였고 연산되는 중간값을 랜덤화하여 DPA에 효과적으로 대응하도록 하였다. 또한 실험을 통하여 SPA/DPA 공격의 대응방법이 될 수 있음을 검증하였다. 그러므로 기존의 대응방법보다 효과적인 방법이 될 수 있다.

참 고 문 헌

[1] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", *in Proceedings of Advances in Cryptology- CRYPTO '96, LNCS 1109,* Springer-Verlag, pp.104~113, 1996.

[2] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", *in Proceedings of Advances in Cryptology-CRYPTO'97,* LNCS 1294, Springer-Verlag, pp.513~525, 1997.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *in Proceedings of Advances in Cryptology-CRYPTO'99, LNCS 1666,* Springer-Verlag, pp.388~397, 1999.

[4] Josyula R. Rao and Pankaj Rohatgi, "EMpowering Side-Channel Attacks", Available at http://eprint.iacr.org/complete/

[5] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power Analysis Attacks on Moular Exponentiation in Smart cards", *in Proceedings of Workshop on Cryptographic Hardware and Embedded Systems-CHES'99, LNCS,* Springer-Verlag, LNCS 1717, pp.144~157, 1999.

[6] R. L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM,* 21, pp.120~126, 1978.

[7] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation,* vol. 48, pp.203~209, 1987.

[8] J. S. Coron, "Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems", *in Proceedings of Workshop on Cryptographic Hardware and Embedded Systems-CHES'99,* LNCS 1717, Springer-Verlag, pp.292~302, 1999.

[9] Y. Hitchcock and P. Montague, "A new elliptic curve scalar multiplication algorithm to resistant simple power analysis", in *Proceedings of Information Security and Privacy-ACISP'02,* 7th Australian Conference, LNCS 2384, pp.214~225, Springer-Verlag, 2002.

[10] JaeCheol Ha and SangJae Moon, "Randomized signed-scalar multip- lication of ECC to resist power attacks", in *Pre-Proceedings of Workshop on Cryptographic Hardware and Embedded Systems-CHES'02,* pp.553~565, Springer-Verlag, 2002.

[11] B. Möller, "Securing elliptic curve point multiplication against side-channel attacks", in *Information Security : 4th International Conference, Proceedings-ISC'01,* LNCS 2200, pp.324~334, Springer-Verlag, 2001.

[12] P. Y. Liardet and N. P. Smart, "Preventing SPA/DPA in ECC systems using the Jacobi Form", in *Proceedings of Workshop on Cryptographic Hardware and Embedded Systems-CHES'01,* LNCS 2162, pp.401~411, Springer-Verlag, 2001.

[13] I. F. Blake, G. Seroussi, and N. P. Smart, "*Elliptic Curves in Cryptography*", *London Mathematical Society Lecture Note Series.* 265, pp.66~72, 1999.

[14] K. Itoh, J. Yajima, M. Takenaka, and N. Torii, 'DPA countermeasure by improving the window method", in *Pre-Proceedings of Workshop on Cryptographic Hardware and Embedded Systems-CHES'02,* pp.304~319, Springer-Verlag, 2002.

[15] K. Okeya and K. Sakurai, "Power analysis breaks elliptic curve cryptosystems even secure against the timing attack", in *Proceedings of Cryptology-INDOCRYPT'00,* LNCS 1977, pp.475~486, Springer-Verlag, 2000.

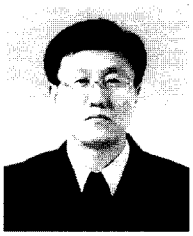
[16] K. Okeya and K. Sakurai, "A second-order DPA attack breaks a window-method based countermeasure against side channel attacks", in *Pre-Proceedings of Information Security Conference-ISC' 02,* Springer-Verlag, 2002.

[17] C. H. Lim and P. J. Lee, "More flexible exponentiation with precomputation", *CRYPTO'94, LNCS 2200,* pp.324~334, Springer-Verlag, 1994.

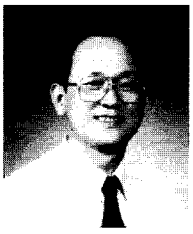
〈著者紹介〉

안 만 기 (MahnKi Ahn) 정회원
2000년 2월 : 경북대학교 전자전기공학부 졸업(학사)
2000년 1월~2001년 1월 : 삼성전자 프린터사업부 C-LBP 연구원
2003년 2월 : 경북대학교 대학원 전자공학과 졸업(정보통신공학, 석사)
2003년 4월~현재 : 국방품질관리소 연구원
<관심분야> 정보보호, 스마트카드 보안, 정보통신

하 재 철 (JaeCheol Ha) 종신회원
1989년 2월 : 경북대학교 전자공학과 졸업(학사)
1993년 8월 : 경북대학교 대학원 전자공학과 졸업(석사)
1998년 2월 : 경북대학교 대학원 전자공학과 졸업(박사)
1998년 3월~2000년 2월 : 나사렛대학교 전자계산소장
1998년 9월~2002년 2월 : 나사렛대학교 학술정보관장
1998년 3월~현재 : 나사렛대학교 정보통신학과 조교수
<관심분야> 정보 보호, 네트워크 보안, 스마트 카드 보안



이 훈 재 (HoonJae Lee) 정회원
1985년 2월 : 경북대학교 전자공학과 졸업(공학사)
1987년 2월 : 경북대학교 전자공학과 졸업(정보통신공학, 공학석사)
1998년 2월 : 경북대학교 전자공학과 졸업(정보통신공학, 공박사)
1987년 2월~1998년 1월 : 국방과학연구소 선임연구원
1998년 2월~2002년 1월 : 경운대학교 컴퓨터전자정보공학부 조교수
2002년 2월~현재 : 동서대학교 인터넷공학부 조교수
<관심분야> 암호이론, 네트워크보안, 디지털 통신



문 상 재 (SangJae Moon) 종신회원
1972년 2월 : 서울대학교 공업교육(전자)과 졸업(학사)
1974년 2월 : 서울대학교 대학원 전자공학과 졸업(석사)
1984년 6월 : 미국 UCLA 전자공학과 졸업(박사)
1984년 7월~1985년 6월 : UCLA Postdoctoral 근무
1984년 7월~1985년 6월 : 미국 OMNET 컨설턴트
1974년 12월~현재 : 경북대학교 공과대학 전자전기컴퓨터학부 교수
2000년 8월~현재 : 경북대학교 이동네트워크 정보보호기술 연구센터 소장
2002년 2월~현재 : 한국정보보호학회 명예회장
<관심분야> 정보보호, 디지털 통신, 이동 네트워크