情報保護學會論文誌
第14券 第6號, 2004. 12

부채널 공격에 안전한 타원곡선 스칼라 곱셈 알고리즘*

김 태 현[a)†], 장 상 운[b)], 김 웅 희[b)], 박 영 호[c)‡]
고려대학교 정보보호대학원[a)], 국가보안기술연구소[b)], 세종사이버대학교[c)]

Elliptic Curve Scalar Multiplication Resistant against Side Channel Attacks

Tae Hyun Kim[a)†], Sang-Woon Jang[b)], Woong Hee Kim[b)], Young-Ho Park[c)‡]
Graduate School of Information Security(GSIS) Korea University[a)], NSRI[b)], Sejong Cyber University[c)]

요 약

스마트카드와 같이 계산능력이나 메모리가 제한된 암호학적인 장치를 구현할 때 부채널 공격을 고려해야 할뿐만 아니라 장치내에 내장되어 있는 암호학적인 알고리즘은 적은 메모리를 이용하여 효율적인 연산을 수행해야 한다. 이러한 목적으로 부채널 공격에 대한 윈도우 기반의 대응방법으로 Möller 방법, Okeya-Takagi 방법, Overlapping window 방법등이 제안되었다. 하지만 Möller 방법과 Okeya-Takagi 방법은 SPA에 안전한 대응방법이기 때문에 다른 공격들(DPA, Second-Order DPA, Address-DPA)을 방어하기 위하여 추가적인 연산이 요구되며 Overlapping window 방법은 많은 저장 공간을 요구하는 단점이 있다. 본 논문에서는 기존의 대응방법들에 대하여 장단점을 분석하고 각각의 대응방법들의 장점을 이용하여 기존의 모든 부채널 공격에 안전하면서 효율적인 대응방법을 제안한다. 더욱이 제안하는 대응방법은 혼합 좌표계를 이용하여 효율성을 더욱더 높일 수 있다.

ABSTRACT

When cryptosystem designers implement devices that computing power or memory is limited such as smart cards, PDAs and so on, not only he/she has to be careful side channel attacks(SCA) but also the cryptographic algorithms within the device has to be efficient using small memory. For this purpose, countermeasures such as Möller's method, Okeya-Takagi's one and overlapping window method, based on window method to prevent SCA were proposed. However, Möller's method and Okeya-Takagi's one require additional cost to prevent other SCA such as DPA, Second-Order DPA, Address-DPA, and so on since they are immune to only SPA. Also, overlapping window method has a drawback that requires big memory. In this paper, we analyze existing countermeasures and propose an efficient and secure countermeasure that is immune to all existing SCA using advantages of each countermeasure. Moreover, the proposed countermeasure can enhance the efficiency using mixed coordinate systems.

Keywords : *Elliptic Curve Cryptosystems, Side Channel Attacks, Countermeasures, SPA, DPA, Second-Order DPA*

접수일 : 2004년 10월 19일 ; 채택일 : 2004년 12월 6일
* 본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구결과로 수행 되었습니다.
† 주저자 : thkim@cist.korea.ac.kr
‡ 교신저자 : youngho@cybersejong.ac.kr

Ⅰ. 서 론

타원곡선 암호시스템은 RSA 암호시스템에 비하여 키 사이즈가 작기 때문에 스마트카드와 같이 계산능력이나 메모리가 제한된 환경에서 구현하기 적합하다. 그렇지만 타원곡선 암호시스템의 구현에 주의를 기울이지 않으면 공격자는 알고리즘 실행시간, 전력소모량 같은 부채널 정보를 이용하여 비밀정보를 찾아낼 수 있다. [5,8-9] 이러한 공격을 부채널 공격 (Side Channel Attack, SCA)이라 한다.

계산능력이나 메모리 같은 자원이 제한된 환경에서 타원곡선 암호시스템을 구현하기 위하여 효율성을 증대시키기 위한 많은 방법들이 제안되었다. 만약 추가적인 메모리를 이용할 수 있다면 윈도우 기반의 방법들은 타원곡선 암호시스템의 속도를 향상시킬 수 있다. 그러나 대부분의 윈도우 기반의 방법들은 [12]에서 Okeya-Sakurai에 의해서 제안된 Second-Order DPA에 의해서 분석되기 때문에 Second-Order DPA 공격을 효율적으로 막는 문제가 현재 가장 큰 쟁점이 되고 있다. 지금까지 제안된 Second-Order DPA를 막는 대응방법들의 단점은 연산의 효율성을 떨어뜨린다는 것이다.

부채널 공격에 대한 여러 가지 윈도우 기반의 대응방법이 제안되었다. Okeya-Takagi는 윈도우 기반의 대응방법을 두 가지 형태로 분류하였다. [13] 첫 번째 형태는 고정된 윈도우를 활용하는 방법이다. 이 방법은 Möller에 의해서 처음 제안되었고 Okeya-Takagi는 w-NAF 방법을 이용하여 저장 공간을 줄이는 방법을 제안하였다. [11,13] 두 번째 방법은 윈도우 기반의 덧셈체인 (addition chains)을 랜덤하게 변화시킴으로써 부채널 공격에 안전하게 하는 것이다. 이 방법은 Liardet-Smart에 의해 [10]에서 처음 제안되었고 [7]에서 Itoh 외 3인에 의해서 더욱 개선된 방법들이 제안되었다.

Ⅱ. 윈도우 기반의 대응방법

본 장에서는 부채널 공격에 대한 윈도우 기반의 대응방법으로 Möller 방법, Okeya-Takagi 방법 그리고 Overlapping window 방법에 대하여 살펴본다. [7,11,13]

Möller [11] 방법과 Okeya-Takagi [13] 방법은 모두 고정된 형태로 비밀키를 재구성함으로써 SPA에 안전하게 된다. 특히 [13]에서 제안된 Okeya-Takagi 방법은 w-NAF 방법을 이용하여 저장 공간을 줄이는데 중점을 두고 있다. 마지막으로 Itoh 외 3인은 [7]에서 Overlapping window 방법을 포함한 윈도우 기반의 DPA 대응방법을 제안하였다.

2.1 Möller의 대응방법

비밀키를 $k=k_e 2^{we}+\cdots+k_1 2^w+k_0$, $k_i \in \{0,1,\cdots,2^w-1\}$ 라 하면 일반적인 타원곡선 스칼라 곱셈 연산에서 $k_i=0$인 경우 덧셈 연산을 수행하지 않게 되며 이러한 성질로 인하여 SPA에 의해서 비밀키가 노출된다. 따라서 Möller는 $k_i=0$인 경우가 발생하지 않도록 비밀키를 다음과 같은 새로운 표현으로 변환하는 방법을 제안하였다. [11]

$$d=k=d_{e+1}2^{w(e+1)}+\cdots+d_1 2^w+d_0.$$

여기서 $d_i \in \{-2^w,1,\cdots,2^w-1\}$ 레코딩 방법은 $0 \le c_i \le 2$와 $0 \le t_i \le 2^w+1$을 만족하는 임시변수 c_i와 t_i를 이용하여 다음과 같이 표현할 수 있다.

$$c_0=0$$

라 하자. 그리고 $i=0,\cdots,e+1$ 에 대하여

$$t_i=k_i+c_i$$

라 하면

$$(c_{i+1},d_i)=\begin{cases}(1,-2^w), & \text{if } t_i=0;\\ (0,t_i), & \text{if } 0<t_i<2^w;\\ (2,-2^w), & \text{if } t_i=2^w;\\ (1,1), & \text{if } t_i=2^w+1.\end{cases}$$

여기서 항상 식 $c_{i+1}\cdot 2^w+d_i=t_i$ 을 만족하고 출

력 정수의 길이는 많아야 한 자리 증가한다.

2.2 Okeya-Takagi의 대응방법

스마트카드와 같이 메모리가 제한된 환경에서 암호학적인 알고리즘은 적은 저장 공간을 이용하면서 효율적이어야 한다. Möller에 의하여 제안된 SPA에 안전한 대응방법은 2^w-ary 방법에 기반을 하고 있기 때문에 2^w개의 저장 공간을 필요로 한다. 적은 저장 공간을 이용하는 윈도우 방법은 w-NAF 방법이다. w-NAF 방법에서 요구되는 저장 공간의 수는 2^{w-2}이다. Okeya-Takagi는 w-NAF 방법을 이용하여 부채널 공격에 안전한 방법을 제안하였다.[13]

Okeya-Takagi 방법은 w-NAF 방법을 SPA에 안전한 덧셈체인(addition chain)으로 변형하는 것이다. 이 덧셈체인은 다음과 같은 고정된 형식으로 스칼라를 생성한다.

$$|0\cdots 0x|0\cdots 0x|\cdots|0\cdots 0x|$$

여기서, $x < 2^w$는 홀수인 정수이다. Okeya-Takagi의 방법에서 요구되는 저장 공간의 수는 2^{w-1}이다. 그러므로 Möller 방법보다 적은 저장 공간을 요구한다.[1] Okeya-Takagi 방법은 모든 짝수 자리를 홀수 자리로 변환함으로써 저장 공간의 수를 줄인다. 예를 들어 $w=4$인 경우에 홀수 자리는 0001,0011,0101,0111,1001,1011,1101 그리고 1111이다. 그러면 짝수 자리 |0100| 을 홀수 자리로 변환하는 방법은 다음과 같다.

$$|0100|01{**}| \rightarrow |0101|\bar{1}\bar{1}{**}|,$$
$$|0100|0\bar{1}{**}| \rightarrow |010\bar{1}|11{**}|.$$

그러면 |0000|을 제외한 나머지 짝수 자리는 위와 같이 과정으로 변환시킬 수 있고 |0000|을 변환하는 방법은 다음과 같다.

$$|{***}0|0000|01{**}| \rightarrow |{***}1|\bar{1}\bar{1}\bar{1}\bar{1}|\bar{1}\bar{1}{**}|$$
$$|{***}0|0000|0\bar{1}{**}| \rightarrow |{***}\bar{1}|1111|11{**}|$$

위의 변형을 이용하여 모든 짝수 자리는 다음과 같이 모든 홀수 자리로 변환할 수 있다.

$$|0000|* \rightarrow |1111|* \text{ or } |\bar{1}\bar{1}\bar{1}\bar{1}|*,$$
$$|0010|* \rightarrow |0011|* \text{ or } |00\bar{1}\bar{1}|*,$$
$$|0100|* \rightarrow |0101|* \text{ or } |01\bar{0}\bar{1}|*,$$
$$|0110|* \rightarrow |0111|* \text{ or } |01\bar{1}\bar{1}|*,$$
$$|1000|* \rightarrow |1001|* \text{ or } |1\,00\bar{1}|*,$$
$$|1010|* \rightarrow |1011|* \text{ or } |1\,01\bar{1}|*,$$
$$|1100|* \rightarrow |1101|* \text{ or } |1\,10\bar{1}|*,$$
$$|1110|* \rightarrow |1111|* \text{ or } |1\,11\bar{1}|*.$$

위의 성질을 이용하여 홀수인 스칼라에 대하여 w-NAF로 변환하는 알고리즘은 다음과 같다.

마지막으로 스칼라가 짝수인 경우를 고려하자. 만일 스칼라 k가 짝수이면 $d=k+1$으로 변환하고 홀수이면 $d=k+2$으로 변환한다. 그러면 d는 항상 홀수이다. 그러므로 d를 위의 알고리즘에 적용하여 스칼라 곱셈을 수행한 후에 $dP-P$ 또는 $dP-2P$을 계산함으로써 kP가 계산된다.

SPA_Protected_w-NAF_Algorithm_with_Odd_Scalar

INPUT An odd n bit integer d and $t=\lceil n/w \rceil$

OUTPUT $d_w[n], d_w[n-1], \cdots, d_w[0]$

1. $u[0] \leftarrow d \bmod 2^w$
2. $d \leftarrow d-u[0]$
3. $d \leftarrow d/2^w$
4. for $i=1$ to t do
   - 4.1 $u[i] \leftarrow d \bmod 2^w$
   - 4.2 if $u[i]$ is even, $b \leftarrow sign(u[i-1])$, $u[i] \leftarrow u[i]+b$, $u[i-1] \leftarrow u[i-1]-b2^w$
   - 4.3 $d_w[(i-1)w] \leftarrow u[i-1]$, $d_w[(i-1)w+1] \leftarrow 0, \cdots, d_w[(i-1)w+w-1] \leftarrow 0$
   - 4.4 $d \leftarrow d-u[i]$, $d \leftarrow d/2^w$
5. $d_w[tw] \leftarrow u[t]$, $d_w[tw+1] \leftarrow 0, \cdots, d_w[tw+w-1] \leftarrow 0$
6. Return $d_w[n], d_w[n-1], \cdots, d_w[0]$

2.3 Overlapping window 방법

Itoh 외 3인은 DPA에 안전한 세 가지 알고리

1) Möller 방법에서 부호화표현을 이용할 때 요구되는 저장 공간은 $2^{w-1}+1$이지만 Okeya-Takagi 방법에서 요구되는 저장 공간은 2^{w-1}이다. 그러므로 부호화 표현을 이용하여도 Okeya-Takagi 방법이 효율적이다.

즘을 제안하였다. 본 절에서는 세 가지 알고리즘 중에 하나인 중첩 윈도우 (overlapping window, O-WM) 방법에 대하여 살펴본다. [7]

먼저 대응방법을 설명하기에 앞서 표기법을 기술한다.

- $d=\sum_{i=0}^{n-1} d_i \cdot 2^i$ 는 비밀키 값이다. r은 윈도우의 크기이고 w_i는 사전 계산된 테이블에 대한 인덱스 값이다. 즉 w_i는 윈도우이다. q는 $w_i(i=0, 1, \cdots, q-1)$의 개수이다. h_i는 w_i와 w_{i+1}사이의 중첩된 비트의 길이를 나타낸다. 즉 $0 < h_i < r$.
- $bit(a, x, \cdots, y)$는 a의 이진 표현에서 x번째 비트부터 y번째 비트까지의 비트열을 나타낸다. ($x=0, 1, \cdots; y=0, 1, \cdots; x \geq y$). 최상위 비트보다 큰 비트는 0으로 간주한다. 예를 들어 $a=6=(110)_2$이면 $bit(a,0)=0$, $bit(a,1)=1$이고 $bit(a,4,3,2,1)=(0011)_2=3$이다.
- ECADD와 ECDBL은 각각 타원곡선 덧셈 연산과 두 배 연산을 나타낸다.

중첩 윈도우 방법의 특징은 윈도우 w_i와 w_{i+1}의 겹침을 허용하는 것이다. 고정된 비밀키 값 d로부터 생성되는 $w_0, \cdots, w_{q-1}$는 스칼라 곱셈을 할 때마다 랜덤하게 변하게 된다. 그러므로 공격자가 중간 결과값을 예상하는 것은 매우 어렵기 때문에 DPA에 안전하게 된다.

$$q \times r - (h_0 + h_1 + \cdots + h_{q-2}) = n, \quad 0 < h_i < \neq 1)$$

중첩 윈도우 방법은 위의 식을 항상 만족하고 중첩 윈도우 방법으로부터 표현되어지는 비밀키는 다음과 같다.

$$d = (\cdots(w_0 \times 2^{r-h_0} + w_1) \times 2^{r-h_1} \cdots) \times 2^{r-h_{q-2}} + w_{q-1}$$

Remark 1. 중첩 윈도우 방법이 SPA에 안전하기 위해서는 h_i를 고정된 값 h으로 사용해야 한다.

중첩 윈도우 방법은 k-ary 방법과 비교하여

Overlapping-Window_Method (OW-M)

INPUT An n bit integer d
OUTPUT $w_0, w_1, \cdots, w_{q-1}$

1. Randomly choose q and $h_0, \cdots, h_{q-2}$ satisfied with equation (1)
2. $n' \leftarrow n-r$, $dt_0 = bit(d, n-1, \cdots, n')$
3. for $i=0$ to $q-2$ do
   3.1 Randomly choose w_i such that
   $\max(0, dt_i - 2^{h_i}+1) \leq w_i \leq dt_i$
   3.2 $dt_{i+1} \leftarrow (dt_i - w_i) \times 2^{r-h_i}$
   $+ bit(d, n'-1, \cdots, n'-(r-h_i))$
   3.3 $n' \leftarrow n' - (r-h_i)$
4. $w_{q-1} \leftarrow dt_{q-1}$
5. Return $w_0, w_1, \cdots, w_{q-1}$

테이블을 만드는 연산량은 같지만 테이블을 참조하는 연산수가 더 많다. 즉 ECADD 연산을 더 많이 수행하게 된다.

Ⅲ. 제안 방법 (w-NAF를 이용한 Overlapping window 방법)

본 장에서는 w-NAF를 이용하여 중첩 윈도우 방법의 저장 공간을 줄이는 방법을 제안한다. 제안하는 방법은 Second-Order DPA 뿐만 아니라 기존의 공격에 안전하고 효율적임을 보일 것이다.

우선 제안하는 방법을 설명하기 위하여 사용할 표기법을 간단히 살펴보자. 윈도우의 크기가 w일 때, SPA 에 안전하기 위하여 윈도우의 중첩된 길이는 고정된 값 h를 이용한다. 그러면 윈도우 개수 s는 $\lceil (n-h)/(w-h) \rceil$이고 비밀키의 길이는 $t=s \cdot (w-h)+h$로 조정된다.

Example 1. 7-비트 비밀키 $d=(1001101)_2$ 에 대하여 $w=4$이고 $h=2$이면 $s=3$이고 $t=3 \cdot (4-2)+2=8$이다. 그러므로 비밀키는 8-비트 정수 $d=(01001101)_2$으로 나타낼 수 있다. 또 d 로부터 생성된 수열의 예는 $d_w[4]=(0011)_2$, $d_w[3]=0, d_w[2]=(1000)_2, d_w[1]=0, d_w[0]=(00\overline{11})_2$ 이다.

비밀키는 다음과 같이 표현된다.

$$d=\sum_{i=0}^{(w-h)\cdot(s-1)} d_w[i]\cdot 2^i=\sum_{i=0}^{(s-1)} d_w[(w-h)i]\cdot 2^{i(w-h)}$$

제안하는 대응방법은 SPA를 막기 위하여 고정된 h를 이용하기 때문에 다음과 같은 형식으로 스칼라 곱셈을 수행한다.

$$\underbrace{|0\cdots 00x}_{w-h-1}|\underbrace{0\cdots 00x}_{w-h-1}|\cdots|\underbrace{0\cdots 00x}_{w-h-1}|$$

여기서, x는 $|x|<2^{w-h-1}$인 홀수이다. 그러면 사전 계산되는 점의 수는 2^{w-h-1}이고 0이 아닌 비트의 조밀도는 $1/(w-h)$이다.

OW-Method using Width w-NAF with Odd Scalar

INPUT An odd n bit integer d, a width w and an overlapping width h

OUTPUT $d_w[(w-h)\cdot(s-1)],\cdots,d_w[0]$

1. $s \leftarrow \lceil (n-h)/(w-h) \rceil$, $t \leftarrow s\cdot(w-h)+h$
2. $dt \leftarrow bit(d, t-1, \cdots, t-w)$, $t \leftarrow t-w$
3. for $i=s-1$ downto 1 do
   3.1 Randomly choose odd integer $u[i]$ such that
   $\max(-2^w+1, dt-2^h+1) \le u[i] \le \min(2^w-1, dt+2^h-1)$
   3.2 $d_w[(w-h)\cdot i] \leftarrow u[i]$, $d_w[(w-h)\cdot i-1] \leftarrow 0$, $\cdots, d_w[(w-h)\cdot(i-1)+1] \leftarrow 0$,
   3.3 $dt \leftarrow (dt-d_w[i])\times 2^{w-h}+bit(d, t-1, \cdots, t-h)$
   3.4 $t \leftarrow t-(w-h)$
4. $d_w[0] \leftarrow dt$
5. Return $d_w[(w-h)\cdot(s-1)],\cdots,d_w[0]$

위의 알고리즘의 step 3.1에서 $u[i]$가 될 수 있는 경우의 수는 일반적으로 2^h이다. 그러므로 각각의 스칼라 곱셈에서 서로 다른 $u[i]$가 선택되기 때문에 DPA에 안전하게 된다. 제안하는 알고리즘과 기존의 중첩 윈도우 알고리즘의 차이는 step 3.1 이다. 기존의 방법은 양의 정수를 선택했지만 제안하는 방법은 홀수인 정수를 선택한다. 그러므로 테이블에 홀수인 점들만 저장하면 되기 때문에 저장 공간이 반으로 줄어든다. 만약 짝수 정수가 위의 알고리즘에 입력된다면 문제가 되는 부분은 최하위 자리이다. 즉 step 4에서 $d_w[0]$는 짝수가 된다. 이제 짝수 스칼라를 홀수로 바꿀 수 있는 두 가지 방법을 설명한다.

첫 번째 방법은 [13]에서 Okeya-Takagi가 설명한 방법이다. 만약 d가 짝수이면 $d'=d+1$으로, 홀수이면 $d'=d+2$으로 변환한다. 그러면 d'은 항상 홀수이고 스칼라 곱셈 후에 $d'P-P$ 또는 $d'P-2P$을 계산함으로써 dP는 복원된다.

두 번째 방법은 Assumption 1에서부터 시작한다.

Assumption 1. 타원곡선 스칼라 곱셈에 입력되는 점은 위수는 큰 소수라고 가정하자. 다시 말해서 위수가 작은 점은 스칼라 곱셈이 수행되기 전에 제거된다고 가정한다.

ECIES, single-pass ECDH와 single-pass ECMQV 등의 프로토콜에 대한 여러 가지 표준문서에는 작은 위수 공격(small subgroup attacks)을 막기 위하여 스칼라 곱셈이 수행되기 전에 작은 위수(cofactor)를 사용하여 위수가 작은 점을 알아 낼 수 있다.[14] 그러므로 위의 가정은 정당하다.

위수가 큰 소수 (p : large prime)인 점 P는 $pP=O$이다. 만약 d가 짝수이면 $d'=d+p$, 홀수이면 $d'=d$로 변환한다. 그러면 d'은 항상 홀수이고 $d'P=dP$이다.

Ⅳ. 안전성과 효율성 분석

본 장에서는 제안하는 대응방법에 대한 안전성과 효율성을 분석하고 2장에서 설명한 세 가지 대응방법과 안전성, 연산량 그리고 저장 공간을 비교한다.

4.1 안전성 분석

본 절에서는 제안하는 대응방법의 안전성을 분석한다. SPA, DPA, Second-Order DPA 및 Address-DPA에 대하여 고려하고 AR(Attenuation Ratio)을 이용하여 안전성을 분석한다.[2] AR은 대응방법이 있는 경우와 없는 경우에 발생

2) AR은 [7]에서 Itoh 외 3인에 의해 처음 제안되었다.

하는 피크의 비율에 의해서 평가된다. 0과 1사이의 값을 갖는 AR은 작을수록 안전하다. 만약 AR=0이면 공격자는 피크를 확인할 수 없고 대응방법은 안전하다. 만약 AR=1이면 공격자는 항상 피크를 확인할 수 있고 대응방법은 안전하지 못하다. SPA, DPA, Second-Order DPA와 Address-DPA에 대한 AR은 각각 AR_S, AR_D, $AR_{S\text{-}D}$와 $AR_{A\text{-}D}$로 표기한다.

4.1.1 SPA

제안하는 알고리즘은 $x\neq 0$에 대하여 고정된 형식으로 스칼라 곱셈을 계산한다. 공격자는 전력소모량의 측정을 통해서 ECADD와 ECDBL을 구별할 수 있다. 그러나 모든 스칼라에 대하여 나타나는 AD 수열은 모두 다음과 같은 고정된 형태로 발생한다.

$$\cdots DDA\ |\underbrace{D\cdots DDA}_{w-h}|\underbrace{D\cdots DDA}_{w-h}|\cdots\ |\underbrace{D\cdots DDA}_{w-h}|DD\cdots,$$

여기서 A와 D는 각각 ECADD와 ECDBL을 의미한다.

비밀키에 관계없이 항상 $w-h$번의 ECDBL과 1번의 ECADD 연산을 수행하게 되므로 공격자가 SPA에 의해서 얻을 수 있는 정보는 없다. 그러므로 $AR_S=0$. 즉, 제안 알고리즘은 SPA에 안전하다.

4.1.2 DPA

제안하는 대응방법은 각각의 스칼라 곱셈에서 서로 다른 수열을 사용하기 때문에 공격자는 스칼라 곱셈이 수행되는 동안에 나타나는 중간값을 예상하는 것이 매우 어렵다.

Remark 2. 사영 좌표계에서 같은 점에 대한 데이터 표현은 다르다. 예를 들면 $A=(X,Y,Z)$에 대한 $7A$가 $(X_1,Y_1,Z_1)=2((11)_2(X,Y,Z))+((01)_2(X,Y,Z))$ 또는 $(X_2,Y_2,Z_2)=2((10)_2(X,Y,Z))+((11)_2(X,Y,Z))$에 의해서 계산될 때, 이 두 표현은 아핀 좌표계에서는 같지만 사영 좌표계에서는 다르다. 즉, 높은 확률로 $X_1\neq X_2$, $Y_1\neq Y_2$, $Z_1\neq Z_2$이다.

제안 알고리즘 OW-Method_using_Width_w-NAF_with_Odd_Scalar에서 $u[i]$가 될 수 있는 경우의 수는 2^h이고 0이 아닌 윈도우의 개수는 s이므로

$$AR_D=\frac{1}{2^{h\times s}}.$$

예를 들어, 160비트 스칼라에 대하여 $w=4$이고 $h=1$인 경우에 $AR_D=2^{-53}$이다. [7]에서 Itoh 외 3인은 대응방법이 적용된 장치에서 발생하는 피크의 크기가 대응방법 없이 구현된 장치에서 발생하는 피크의 크기보다 100배 정도 줄어들면 충분한 안전성을 줄 수 있음을 언급하였다. 그러므로 이 수치는 충분히 DPA에 안전함을 보여준다.

4.1.3 Second-Order DPA (SO-DPA)

Okeya-Sakurai는 [12]에서 Möller의 윈도우 방법에 대한 Second-Order DPA 공격을 제안하였다. Second-Order DPA 공격은 비밀키의 자리가 같을 경우 같은 테이블 값을 참조하는 성질을 이용한다. 그러므로 Second-Order DPA를 막기 위해 테이블을 참조할 때마다 테이블을 랜덤화하는 방법을 사용해야 한다.

그러나 제안 대응방법은 각각의 스칼라 곱셈에서 서로 다른 수열을 사용하기 때문에 매번 테이블을 참조하는 값이 달라진다. 그러므로 Second-Order DPA에 의해서 비밀키를 찾는 것이 어렵다. 더욱이 제안 대응방법은 Second-Order DPA를 막기 위하여 추가적인 연산이 필요하지 않으므로 연산의 효율성을 높일 수 있는 장점이 있다.

4.1.4 Address-DPA (A-DPA)

Itoh-Izu-Takenaka는 [6]에서 비밀키를 랜덤화하는 형태의 대응책은 A-DPA에 안전하다고 언급하였다. 비밀키를 랜덤화하는 방법은 스칼라 곱셈을 할 때마다 비밀키의 표현을 랜덤하게 변화시키기 때문에 레지스터의 주소 또한 랜덤하게 변하게 된다. 그러므로 제안 대응방법은 A-DPA에 안전하다.

표 1. 원도우 기반의 대응방법에 대한 안전성 비교

Method	SPA	DPA	SO-DPA	A-DPA
Möller	Immune	Vulnerable	Vulnerable	Vulnerable
Okeya-Takagi	Immune	Vulnerable	Vulnerable	Vulnerable
Overlapping-Window	Immune	Immune	Immune	Immune
Proposed	Immune	Immune	Immune	Immune

표 2. 원도우 기반의 대응방법에 대한 효율성과 저장 공간 비교

Method	ECDBL	ECADD	Table Size
Möller	n	n/w	2^w
Okeya-Takagi	n	n/w	2^{w-1}
Overlapping-Window (h : 고정)	n	$\approx n/(w-h)$	2^w
Proposed	n	$\approx n/(w-h)$	2^{w-1}

표 1은 윈도우 기반의 대응방법에 대하여 SPA, DPA, Second-Order DPA 및 Address-DPA에 대한 안전성을 보여준다.

4.2 효율성 분석

스칼라 곱셈의 연산량은 각각 ECDBL과 ECADD의 수에 의해서 평가된다. 그리고 저장 공간의 수는 테이블에 저장되는 점의 개수로 평가한다.

표 2에서의 ECDBL과 ECADD의 수는 테이블을 만들 때 요구되는 연산량은 포함되어 있지 않고 스칼라 곱셈을 수행할 때 필요한 연산량을 나타낸다. 제안하는 방법은 Möller 방법과 Okeya-Takagi 방법보다 더 많은 ECADD 연산을 요구함을 표 2에서 보여주기 때문에 제안하는 방법의 연산속도가 Möller 방법과 Okeya-Takagi 방법의 연산속도보다 더 느린 것처럼 보인다. 그러나 실제로 Second-Order DPA를 막기 위하여 대응방법을 추가되면 제안하는 방법이 더 효율적임을 보일 것이다.

Möller의 대응방법과 Okeya-Takagi의 대응방법은 모두 SPA를 막기 위해서 제안된 알고리즘이기 때문에 DPA와 Second-Order DPA에 안전하지 못하다. 그러므로 이 대응방법들은 DPA를 막기 위하여 랜덤 사영좌표계 또는 랜덤 타원곡선 동형사상등의 DPA 대응방법을 함께 사용해야 하고 Second-Order DPA를 막기 위하여 ECADD 연산을 수행할 때마다 테이블을 랜덤화하는 방법을 사용해야 한다. 예를 들면 Jacobian 좌표계에서 테이블을 랜덤화하기 위하여 ECADD 과정마다 (r^2X, r^3Y, rZ) 연산을 수행해야 한다. 즉 ECADD을 수행할 때마다 4번의 유한체 곱셈과 1번의 유한체 제곱 연산이 추가적으로 요구된다. 표 3은 모든 대응방법들이 SPA, DPA 및 Second-Order DPA에 모두 안전하게 구성되었을 때, 160 비트 스칼라와 윈도우가 4이고 제안 알고리즘에서 $h=1$인 경우에 대한 연산량과 저장 공간을 보여준다. 표 3에서 Additional Mul.은 Jacobian 좌표계에서 ECADD 연산 전에 (r^2X, r^3Y, rZ)을 수행할 때 필요한 연산량이

표 3. 기존의 부채널 공격에 안전한 대응방법들의 안전성, 효율성과 저장 공간 비교 (160-bit key, w=4, h=1)

Method	AR_S	AR_D	AR_{SD}	AR_{AD}	ECDBL	ECADD	Additional Mul.	Table Size
Möller + DPA + SO-DPA	0	2^{160}	0	0	160	40	160M+40S	16
Okeya-Takagi + DPA + SO-DPA	0	2^{160}	0	0	160	40	160M+40S	8
Overlapping-Window (h : 고정)	0	2^{53}	0	0	160	53	-	16
Proposed	0	2^{53}	0	0	160	53	-	8

표 4. 사영 좌표계(projective coordinate)(P), Jacobian 좌표계(J), Chudnovsky Jacobian 좌표계(J^C) 그리고 변형된 Jacobian 좌표계(J^m)에서 대응방법들의 연산량 비교 (160-bit key, w=4, h=1)

Method	좌표계	Computing Time	
		Z≠1	Z=1
Möller + DPA + SO-DPA 또는 Okeya-Takagi + DPA + SO-DPA	P	1720M+880S=2424M	-
	J	1280M+1160S=2208M	-
	J^C	1480M+1120S=2376M	-
	J^m	1360M+960S=2128M	-
Proposed	P	1756M+906S=2480.8M	1597M+906S=2321.8M
	J	1276M+1172S=2213.6M	1064M+1119S=1959.2M
	J^C	1383M+1119S=2278.2M	1224M+1119S=2119.2M
	J^m	1329M+958S=2095.4M	1117M+905S=1841M

다. Address-DPA는 [6]에서 제안된 레지스터의 주소를 랜덤화하는 방법에 의해서 추가적인 연산 없이 쉽게 막을 수 있다.

표 4는 SPA, DPA 및 Second-Order DPA를 모두 막는 대응방법에 대하여 160 비트 스칼라와 윈도우가 4이고 $h=1$인 경우에 $GF(p)$에서 정의되는 여러 좌표계에서의 연산량을 보여준다. $GF(p)$에서 제곱 연산량은 일반적으로 곱셈 연산량의 80% 정도 소비된다(S=0.8M). 제안하는 알고리즘은 ECADD를 계산할 때 $Z=1$로 놓고 계산할 수 있어 연산이 효율적임을 알 수 있다. 더욱이 제안하는 알고리즘은 좌표계의 선택이 자유롭지만 Möller 방법과 Okeya-Takagi 방법은 그렇지 못하다. Möller 방법과 Okeya-Takagi 방법이 Jacobian Chundonovsky 좌표계에서 ECADD 연산을 수행한다면 Second-Order DPA를 막기 위하여 다섯 개의 좌표를 랜덤화해야 하므로 연산량이 늘어나게 된다. 그러므로 제안 알고리즘은 ECADD와 ECDBL 연산에 대하여 각각의 연산이 효율적인 좌표를 자유롭게 선택하여 연산의 효율성을 높일 수 있다.

V. 결 론

본 논문에서는 SPA, DPA, A-DPA 및 Second-Order DPA에 안전한 대응방법을 제안하였다. 제안하는 대응방법은 중첩 윈도우 방법에 w-NAF 방법을 적용함으로써 기존의 부채널 공격에 안전하면서 저장 공간을 줄이고 연산의 효율성을 높이는데 중점을 두었다. 다른 대응방법과의 단순한 효율성 비교는 제안하는 대응방법과 기존의 방법들과 차이점이 없어 보이지만 효율성을 높이기 위하여 혼합좌표계를 이용하면 기존의 대응방법들은 좌표계에 따라서 추가적인 연산이 요구되지만 제안하는 방법은 추가적인 연산 없이 자유롭게 좌표계를 선택함으로써 효율성을 높일 수 있다. 그러므로 제안하는 대응방법은 기존의 부채널 공격에 안전하면서 적은 저장 공간을 이용하여 효율적인 연산을 할 수 있다.

참 고 문 헌

[1] 안만기, 하재철, 이훈재, 문상재, "타원곡선 암호시스템에서 랜덤 m-ary 방법을 사용한 전력분석 공격의 대응방법," 정보보호학회논문지, 13권 3호, 35-43, 2003.

[2] 장상운, 정석원, 박영호, "전력분석공격을 효율적으로 방어하는 타원곡선 비밀키의 랜덤화," 정보보호학회논문지, 13권 5호, pp. 169-177, 2003.

[3] 하재철, 곽동진, 문상재, "Folding 기법을 이용한 전력분석 공격에 대응하는 고속 스칼라 곱셈," 정보보호학회논문지, 13권 3호, pp. 57-64, 2003.

[4] 한동국, 장남수, 장상운, 임종인, "랜덤한 덧셈-뺄셈 체인에 대한 부채널 공격," 정보보호학회논문지, 14권 5호, pp. 121-133, 2004.

[5] J. S. Coron, "Resistance against Differential Power Analysis for Elliptic

Curve Cryptosystems," CHES 1999, LNCS 1717, pp. 292-302, Springer-Verlag, 1999.

[6] K. Itoh, T. Izu, M. Takenaka, "A Practical Countermeasure against Address-bit Differential Power Analysis," CHES 2003, LNCS 2779, pp. 382-396, Springer-Verlag, 2003.

[7] K. Itoh, J. Yajima, M. Takenaka, N. Torrii, "DPA Countermeasures by Improving the Window Method," CHES 2002, LNCS 2523, pp. 303-317, Springer-Verlag, 2003.

[8] P. Kocher, "Timing attacks on implmentation of Diffie-Hellman, RSA, DSS, and other systems," CRYPTO 1996, LNCS 1109, pp.104-113, Springer-Verlag, 1996.

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

[10] P. Y. Liardet, N. P. Smart, "Preventing SPA/DPA in ECC systems using the Jacobi form," CHES 2001, LNCS 2162, pp. 391-401, Springer-Verlag, 2001.

[11] B. Möller, "Securing Elliptic Curve Point Multiplication against Side-Channel Attacks," ISC 2001, LNCS 2200, pp. 324-334, Springer-Verlag, 2001.

[12] K. Okeya, K. Sakurai, "A Second-Order DPA Attack Breaks a Window-Method Based Counterm- easure against Side Channel Attack," ISC 2002, LNCS 2433, pp. 389-401, Springer-Verlag, 2002.

[13] K. Okeya, T. Takagi, "The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks," CT-RSA 2003, LNCS 2612, pp.328-342, Springer-Verlag, 2003.

[14] Standards for Efficient Cryptography Group (SECG). Specification of Standards for Efficient Cryptography.

〈著者紹介〉

김 태 현 (Tae Hyun Kim) 학생회원
2002년 2월 : 서울시립대학교 수학과 이학사
2004년 8월 : 고려대학교 정보보호대학원 공학석사
〈관심분야〉 공개키 암호 알고리즘, 부채널 공격, 암호칩 설계 기술

장 상 운 (Sang-Woon Jang) 정회원
2002년 2월 : 고려대학교 수학과 이학사
2004년 2월 : 고려대학교 정보보호대학원 공학석사
2004년 3월~현재 : 국가보안기술연구소 연구원
〈관심분야〉 암호이론, 정보보호

김 웅 희 (Woong Hee Kim) 정회원
2000년 2월 : 고려대학교 전기전자전파공학부 공학사
2002년 2월 : 한국과학기술원 전자전산학과 석사
2002년 3월~현재 : 국가보안기술연구소 연구원
〈관심분야〉 정보보호, 전자공학, 통신공학

박 영 호 (Young-Ho Park) 정회원
1990년 2월 : 고려대학교 수학과 이학사
1993년 2월 : 고려대학교 수학과 이학석사
1997년 2월 : 고려대학교 수학과 이학박사
2002년 3월~현재 : 세종 사이버 대학교 조교수
〈관심분야〉 정수론, 공개키 암호, 암호 프로토콜, 부채널 공격