랜덤 부호화 스칼라 곱 알고리즘 분석

한동국*, 김태현*, 장상운*, 박영호**

*고려대학교, 정보보호학과, **세종사이버대학교, 정보보호시스템공학과

Cryptanalysis of the Randomized Signed-Scalar Multiplication

Dong-Guk Han*, Tae-Hyun Kim*, Sang-Woon Jang*, Young-Ho Park**

*Center for Information and Security Technologies(CIST), Korea Univ.

**Dept. of Information Security, Sejong Cyber Univ.

요 약

부채널 공격(side channel attack)을 막는 새로운 접근방법으로 생각되는 랜덤 부호화 스칼라 곱 알고리즘은 Ha와 Moon에 의해서 제안되었다. 그러나 이 방법은 여전히 논쟁의 여지가 있다. 본 논문에서는 Ha-Moon 알고리즘이 기존의 세 가지 단순 전력 소모량 분석(simple power analysis, SPA)에 안전함을 보인다. 그리고 정수론의 성질을 이용하여 두 가지 중요한 정리를 제시하고 이 정리들을 이용하여 Ha-Moon 알고리즘에 적용할 수 있는 공격 알고리즘을 개발한다. 예를 들면, 163-비트 키들에 대하여 제안 알고리즘은 20개의 전력 소모량을 이용하여 키 복잡도 $O(2^8)$를 가지고 공격할 수 있다.

Ⅰ. 서론

최근에, Kocher[7]는 암호시스템의 스마트 카드 구현에서 부채널 공격을 기술하였다. 암호시스템의 구현에서 부채널 공격은 실행시간[6] 또는 전력 소비량 등을 사용한다. 타원곡선 암호시스템에 전력 소비량 분석 공격을 적용하는 기본적인 원리는 [1]에서 Coron에 의해서 논의되었다. [1]에서 Coron은 전력 소비량 공격에 대한 세 가지 대응책을 제시하였다; 비밀키의 랜덤화, 기본점의 Blinding 그리고 사영 좌표계의 랜덤화. 그러나 [11]에서 Okeya와 Sakurai는 Coron의 첫 번째와 두 번째 DPA(differential power analysis) 대응책의 약점을 설명하였다. Joye와 Tymen[4]는 두 가지 DPA 대응책을 제안하였다; 랜덤 타원곡선 동형사상과 랜덤 유한체 동형사상. 그러므로 DPA를 막기 위해서는 세 가지 DPA 대응책(사영 좌표계의 랜덤화, 랜덤 타원곡선 동형사상, 랜덤 유한체 동형사상)중에 하나를 주로 사용하였다. 그러나 Goubin [2]은 위의 세 가지 대응책은 타원곡선에서 특별한 점의 성질이 유지되는 것을 이용하여 공격됨을 보였다. SPA를 막는 접근방식은 두 종류가 있다. 첫 번째는 스칼라 곱에서 덧셈과 두 배 연산을 구별할 수 없게 하는 방법이다. 예를 들면, Hesse와 Jacobi 형태의 타원곡선들은 덧셈과 두 배를 같은 공식을 사용하여 구별할 수 없게 만든다[5,8]. 두 번째는 덧셈과 두 배를 항상 연산하는 방법이다. 예를 들면, 불필요한 연산(dummy operation)을 추가한 Coron의 방법[1]과 몽고메리 방법이다[10].

■본 논문의 기여

Ha와 Moon은 DPA를 막기 위하여 non-adjacent form(NAF) 부호화 알고리즘과 랜덤화의 개념을 이용해서 효과적이고 새로운 랜덤 재부호화(recording) 알고리즘을 제안하였다. SPA를 막기 위해서는 추가적으로 SPA에 안전한 덧셈-뺄셈 스칼라 곱 알고리즘을 써야한다. 본 논문에서는 우선 Ha-Moon의 알고리즘이 기존의 세 가지 SPA 공격에 공격되지 않음을 보일 것이다.

본 논문은 두 가지 중요한 정리와 이 성질들에 대한 보조정리를 제시하고 Ha-Moon 알고리즘에

적용할 수 있는 SPA 공격 알고리즘을 개발한다. 예를 들면, 163-비트 키들에 대하여 제안 알고리즘은 20개의 전력 소모량을 이용하여 키 복잡도 $O(2^8)$를 가지고 공격할 수 있다.

더욱이, 제안 공격 알고리즘은 다음과 같은 성질을 만족하는 랜덤 부호화 스칼라 곱 알고리즘에 적용할 수 있다.

Property 1 : 랜덤하게 재부호화하여 0이 아닌 비트가 발생할 필요 충분 조건은 덧셈 또는 두 배 연산이 수행되는 것과 동치이다. 이것이 의미하는 것은 랜덤하게 재부호화하여 0인 비트가 발생할 필요 충분 조건은 두 배 연산이 수행되는 것과 동치이다.

Ⅱ. 랜덤 부호화 스칼라 곱

Okeya-Takagi[15]는 부채널 공격에 대한 대응책을 4가지 종류로 분류하였다; 고정된 프로시저(procedure) 형태, 랜덤화된 addition-chains 형태, 연산의 구별 불가능한 형태, 데이터 랜덤화 형태. 고정된 프로시저 형태는 SPA와 시간공격(timing attack)을 막기 위하여 연산의 순서가 미리 결정된 고정된 프로시저를 사용하여 스칼라 곱을 계산한다. 이에 반하여 랜덤화된 addition-chains 형태는 실행을 할 때마다 시간공격, SPA와 DPA를 막기 위하여 서로 다른 addition-chains을 사용한다. 그렇지만 랜덤화된 addition-chains 형태의 안전성은

표 1 : 랜덤 재부호화 방법

입 력				출 력	
k_{i+1}	k_i	c_i	r_i	c_{i+1}	d_i
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	1	-1
0	1	0	0	0	1
0	1	0	1	1	-1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	-1
1	0	1	1	0	1
1	1	0	0	1	-1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	0

표 2 : k=(1001011010)$_2$ 일 때 랜덤하게 재부호화된 수 $d(i)$를 찾는 예제

Index	10	9	8	7	6	5	4	3	2	1	0
	MSB										LSB
키 k	1	0	0	0	0	1	1	1	1	0	0
캐리 c	1	0	0	0	0	1	1	1	1	0	0
난수 r		1	0	1	0	1	0	1	0	1	0
$d(1)$	1	-1	0	0	1	1	0	0	-1	-1	0
캐리 c	0	0	0	1	1	1	1	0	0	0	0
난수 r		0	1	0	1	0	1	0	1	0	1
$d(2)$		1	0	1	0	-1	0	-1	0	1	0
캐리 c	1	1	1	1	0	0	0	0	0	0	0
난수 r		0	0	1	1	0	0	1	1	0	0
$d(3)$	1	0	-1	-1	-1	0	1	1	0	1	0
캐리 c	1	0	0	1	0	0	0	0	0	0	0
난수 r		1	0	0	1	0	0	1	0	0	0
$d(4)$	1	-1	0	1	-1	0	1	1	0	1	0
캐리 c	1	0	0	0	0	1	1	0	0	0	0
난수 r		1	0	1	0	1	1	0	0	0	0
$d(5)$	1	-1	0	0	1	1	0	-1	0	1	0

논쟁의 여지가 남아 있다. 왜냐하면 모든 가능한 addition-chains의 수는 모든 가능한 스칼라의 수와 비교하면 작다. 그리고 앞선 계산에서 사용된 난수는 다음 계산에 영향을 주지 않는다. 실제로 Okeya-Sakurai는 Oswald의 랜덤화된 addition-subtraction chains 방법[14]이 덧셈과 두 배 연산의 구별이 가능한 상황에서 SPA에 공격당하기 쉬움을 증명하였다[12]. Walter는 Oswald 방법에 대한 다른 SPA 공격을 제안하였다. 이 두 가지 공격은 다음 장에서 살펴본다.

이에 반해서 Ha와 Moon[3]은 DPA를 막기 위하여 NAF 알고리즘과 랜덤화의 개념을 모두 사용하여 랜덤 addition-chains 형태(즉, 랜덤 스칼라 재부호화 알고리즘)의 대응책을 제시하였다. k는 다음과 같은 n-bit 정수라 하자. $k=\sum_{i=0}^{n-1} k2^i$, $d_i \in \{1,0\}$, 여기서 c_i는 $c_0=0$인 i번째 임시 캐리 비트이고, d_i는 재부호화된 i번째 랜덤 비트이다. 랜덤하게 재부호화된 d_i와 다음 임시 캐리 비트 $c_{i+1}(0 \le i \le n)$는 표 1에서와 같이 순차적으로 생성된다. 표 2는 $k=(1001011010)_2$가 주어졌을 때 난수 r에 따라서 랜덤하게 재부호화된 수 d가 어떻게 생성되는지 보여주는 예제이다. 이 예제가 4장에서 사용되는 것에 유념하자.

Remark 1. 랜덤하게 재부호화된 수 d에서 0이

아닌 비트의 수는 $n/2$이다[3]. 그러므로 랜덤 스칼라 재부호화 알고리즘에 의해서 요구되는 덧셈(뺄셈)연산의 평균적인 수는 $n/2$이다.

Ⅲ. 기존의 SPA 분석 방법

이 장에서는 Ha-Moon의 랜덤 부호화 스칼라 곱 알고리즘은 기존의 세 가지 SPA 공격법에 공격되지 않음을 주장한다. 첫 번째 방법은 Oswald [14]에 의해 제안된 랜덤화된 오토마타 1에서 Okeya와 Sakurai[12]의 공격방법이다. 두 번째 방법은 Morain과 Oliver[9]에 의해서 제안된 오토마타 1을 분석하기 위하여 Oswald[13]에 의해 제안된 마코프 체인 모델(Markov chain model)이다. 세 번째는 일반적인 랜덤화된 오토마타에서 Walter의 공격방법이다[17].

A와 **D**를 각각 덧셈(뺄셈)과 두 배 연산이라 하자. **A**와 **D**는 왼쪽부터 오른쪽으로 순차적으로 쓴다.

3.1. Okeya-Sakurai 분석 방법

[14]에서 랜덤화된 오토마타 1은 Ha-Moon의 랜덤 부호화 스칼라 곱 알고리즘과 비교하여 다른 특성들을 가지고 있다. 표 3에서 보는 것과 같이 랜덤화된 오토마타 1은 비트 패턴과 AD 수열 사이에 일대다(one-to-many) 대응이 존재한다. 예를 들면, Okeya와 Sakurai는 랜덤화된 오토마타 1의 다음과 같은 특성를 이용하여 비트 패턴과 AD 수열사이의 일대일 대응관계를 부여하였다. 만약 어떤 AD 수열이 (ADDD)을 포함하고 또 다른 어떤 AD 수열이 (ADADA 또는 ADDAA)을 포함하고 있으면 이들 AD 수열에 해당하는 비트 패턴을 111으로 설정한다. 반면, Ha-Moon의 부호화 스칼라 곱 알고리즘은 비트 패턴과 AD 수열사이에 다

표 3 : 랜덤화된 오토마타 1 과 Ha-Moon 알고리즘의 특징 비교.

[14]의 랜덤화된 오토마타 1		HA-Moon's Algorithm	
비트 패턴	AD 수열	비트 패턴	AD 수열
11	**ADAD,ADD,ADDA**	11, 1-1, -11,-1-1	**ADAD**
10	**ADAD,ADD**	10, 10	**ADD**
01	**DAD**	01, 01	**DAD**
00	**DD**	00	**DD**

표 4 : 2단계 상태 전이 조건부 확률

Pr(00|**DD**)=1	Pr(10|**ADD**)=1/2 Pr(-10|**ADD**)=1/2
Pr(01|**DAD**)=1/2 Pr(0-1|**DAD**)=1/2	Pr(11|**ADAD**)=1/4 Pr(-11|**ADAD**)=1/4 Pr(1-1|**ADAD**)=1/4 Pr(-1-1|**ADAD**)=1/4

대일(many-to-one)대응이 존재한다. 공격자가 단지 AD 수열로부터 알아낼 수 있는 정보는 AD 수열에 해당하는 비트 패턴의 해밍웨이트 뿐이다. 그러므로 Okeya-Sakurai 분석 방법은 Ha-Moon에 의해 제안된 알고리즘에 적용할 수 없다.

3.2. Makov chains 모델

마코프 체인 모델(Makov chains model)을 이용하여 Ha-Moon 방법을 분석하면 마코프 체인 모델의 장점이 없다는 것을 알 수 있다. [3]에서의 부호화 알고리즘에서 마코프 체인 모델은 알고리즘의 안전성 분석에 적용할 수 없다. 단지 부호화된 스칼라의 해밍웨이트(Hamming weight)를 결정할 수 있다. 표 4는 2단계 상태전이 조건부 확률 계산 결과를 분석한 간단한 예제이다.

다음은 Ha- Moon의 알고리즘은 마코프 모델에 의한 분석이 불가능한 이유이다.

1. 비트 패턴
   (a) Oswald[13] : 고정된 상태전이의 수에 대하여 3가지 비트 패턴만이 가능하고, 또는 나타나지 않는 경우가 있다.
   (b) Ha-Moon : 연산 **AD**는 비트 1 또는 -1 에 해당된다. 주어진 AD 수열에 대하여, 가능한 비트 패턴의 개수는 2^l 개인데, 여기서 l 은 AD 수열내의 연산 **AD** 의 개수를 말한다. 비트 0 의 위치는 AD 부분수열에서 결정된다.
2. 확률 분포
   (a) Oswald[13] : 3가지 가능한 비트 패턴에 해당하는 확률은 1/2, 1/4, 1/4 으로 나타난다.
   (b) Ha-Moon : 2^l 개의 가능한 비트 패턴에 대한 확률은 모두 $1/2^l$ 으로서 균등하다.

순수한 SPA를 이용한 Ha-Moon 알고리즘의 분석은 옳은 키를 찾기 위하여 **Remark 1**과 Pr(di=1) = Pr(di=-1) = 1/2 때문에 편중된 확률이 없이 $n/2$ 키 전수조사를 해야한다. 그러므로 마코프 체인 모델의 분석 복잡도는 정확하게 비밀키를

찾는 SPA의 복잡도와 같게 된다.

3.3. Walter 분석 방법

S는 타원곡선 뎃셈 연산이라 하자. Ha-Moon 알고리즘에서 비밀키 비트와 타원곡선 연산과의 대응관계는 다음과 같다:

$$1 \leftrightarrow \mathbf{AD}, \quad -1 \leftrightarrow \mathbf{SD}, \quad 0 \leftrightarrow \mathbf{D}.$$

Oswald-Aigner 알고리즘[17]의 일반화된 버전에서 그것들의 대응관계는 다음과 같다:

$$1 \leftrightarrow \mathbf{AD}, \ -1 \leftrightarrow \mathbf{SD}, \ 2 \leftrightarrow \mathbf{DA}, \ 0 \leftrightarrow \mathbf{D}.$$

만약 **DA** 의 사건이 제거된다면 마치 Ha-Moon 알고리즘은 Oswald-Aigner 알고리즘의 일반화된 버전처럼 보인다. 그러나 두 알고리즘의 차이점은 확연히 드러난다. 예를 들면, -1 비트는 Ha-Moon 알고리즘에서 연속적으로 발생할 수 있다. 그러나 일반적인 버전의 경우에는 그렇지 않다. 이것이 Walter의 분석 방법은 Ha-Moon 알고리즘에 직접적으로 적용할 수 없는 이유이다.

Ⅳ. 랜덤 부호화 스칼라 곱 알고리즘 분석

Assumption 1 : 덧셈과 두 배 연산은 한번의 전력 소비량 측정으로 구별할 수 있다. 그러나 덧셈과 뎃셈 연산은 구별할 수 없다.

Assumption 2 : 왼쪽부터 오른쪽으로 실행되는 덧셈-뎃셈 스칼라 곱 알고리즘([3]의 그림 3.)은 [3]에서 제안된 랜덤 부호화 스칼라 곱 알고리즘을 계산하기 위하여 사용한다.

랜덤 스칼라 재부호화 알고리즘[3]은 **Property 1** 을 만족한다.

그래서 공격은 덧셈 패턴의 발생에 대한 전력 소비량을 관찰하는 것에 의해서 매우 직관적인 방법으로 할 수 있다. 랜덤 재부호화 방법의 $n/2$은 **Remark 1** 으로부터 0이 아닌 비트이기 때문에 n 비트 비밀키를 찾기 위하여 평균적으로 약 $2^{n/2}$ 개의 키를 조사해야 한다.

Assumption 3 : 공격자는 랜덤 부호화 스칼라 곱 알고리즘이 있는 암호학적인 장치에 타원곡선 점을 입력할 수 있고 덧셈과 두 배의 수열(AD 수열)을 얻을 수 있다. 이러한 과정을 s번 반복하여 s개의 AD 수열을 얻는다. S_i 는 i번째 AD 수열이라 하자 ($1 \le i \le s$).

Notation : k는 n비트 비밀키 값이고 d는 k로부터 생성된 랜덤하게 재부호화된 $n+1$비트 수라고 하자. i번째 랜덤하게 재부호화된 수를 다음과 같이 표시한다.

$$d(i) \triangleq \sum_{j=0}^{n} d_{i,j} 2^j \text{ 여기서. } d_{i,j} \in \{-1, 0, 1\}$$

Assumption 2에 의해서 공격자는 AD 수열 S_i로 부터 다음과 같은 방법에 의해서 부호화된 스칼라 $d(i)$로 변환할 수 있다.

1. AD 수열은 **D**와 **DA**사이를 기호 | 에 의해서 분리할 수 있다.
2. **D** ⇔ 랜덤하게 재부호화된 비트는 0이다.
3. **DA** ⇔ 랜덤하게 재부호화된 비트는 1 또는 -1이다.

Example 1 : 공격자가 **DADADDDADDADD**와 같은 AD 수열 S를 얻었다고 가정하자. 그러면 **DADADDDADDADD**는 다음과 같이 분리된다. **DA | DA | D | D | DA | D | DA | D | D** 그러므로 $d_0 = d_1 = d_3 = d_5 = d_6 = d_9 = d_4 = d_7 = d_8 =$? 기호 ? 는 비트가 1인지 -1인지 아직 결정하지 못한 것을 의미한다.

Attacker's Goal : s개의 AD 수열 S_i로부터 공격자는 AD 수열 S_i를 부호화된 스칼라 $d(i)$ $(0 \le i \le s)$로 변환한다. s개의 부호화된 스칼라 $d(i)$를 이용해서 공격자는 비밀키 값을 찾기 위한 조사 회수를 줄이기 원한다. 즉, n비트 비밀키 값을 찾기 위한 조사 회수가 $2^{n/2}$보다 작게 만든다.

$i \in N$에 대하여 $J_i \subseteq \{0, \cdots, n\}$라 하자. $d_i(J_i, n) \triangleq (d_{i,0}, \cdots, d_{i,n})$와 $d_i(J_i, l, m) \triangleq (d_{i,l}, \cdots, d_{i,m})$ (여기서, 만약 $j \in J_i$이면 $d_{i,j} = 0$이고 그렇지 않으면 $d_{i,j} = \pm 1$로 정의한다. $\langle d_i(J_i, n) \rangle \triangleq \sum_{j=0}^{n} d_{i,j} 2^j$ 와 $\langle d_i(J_i, l, m) \rangle \triangleq \sum_{j=l}^{m} d_{i,j} 2^j$ (여기서, $0 < l < m < n$)로 정의한다.

Theorem 1. 임의의 $J_i \subseteq \{0, \cdots, n\}$에 대하여 $i = 1, 2$일 때 $\langle d_1(J_1, n) \rangle = \langle d_2(J_2, n) \rangle$라고 가정하자. 만약 $t \in J_1 \cap J_2$ 이고 $t \notin \{0, n\}$라면 $\langle d_1(J_1, t-1) \rangle = \langle d_2(J_2, t-1) \rangle$이고 $\langle d_1(J_1, t+1, n) \rangle = \langle d_2(J_2, t+1, n) \rangle$이다.

4.1. 여러 개의 AD 수열을 이용한 강력한 공격 알고리즘

이 절에서는 여러 개의 AD 수열을 이용한 강력

한 공격 알고리즘을 제안한다. 이 알고리즘은 다음과 같은 일반화된 theorem과 corollary를 필요로 한다.

Corollary 1 : $i=1,\cdots,m$에 대하여 $J_i \subseteq \{0,\cdots,n\}$라 하자. $\langle d_1(J_1,n)\rangle = \cdots = \langle d_m(J_m,n)\rangle$라고 가정하자. 만약 $t_1, t_2 \in \bigcap_{i=0}^{m} J_i$ $(0<t_1<t_2<n)$라면 $\langle d_1(J_1,t_1+1, t_2-1)\rangle = \cdots = \langle d_m(J_m,t_1+1,t_2-1)\rangle$이다.

Theorem 2 : $i=1,\cdots,m$에 대하여 $J_i \subseteq \{0,\cdots,n\}$라 하자. 만약 $\bigcap_{i=1}^{m} J_i = 0$이면 $|\{ \langle d_1(J_1,n)\rangle, \cdots, \langle d_m(J_m,n)\rangle \mid \langle d_1(J_1,n)\rangle = \cdots = \langle d_m(J_m,n)\rangle \}| = 2^{n+1-|\bigcup_0^n J_i|} = 2^x$여기서, $x = |\{j \mid D_{1,j},\cdots,d_{m,j} \neq 0,\ 0 \le j \le n\}|$

■ **Attack Algorithm**

1. AD 수열 수집
2. 데이터 변환
3. 데이터 분석
   - 3.1 m 선택 $(2<m<s)$.
   - 3.2 s개의 부호화된 스칼라 $\{\langle d_i(J_i,n)\rangle \mid 1 \le i \le s\}$ 중에 m개의 부호화된 스칼라 선택 $\{\langle d_{j_1}(J_{j_1},n)\rangle, \cdots, \langle d_{j_m}(J_{j_m},n)\rangle\}$.
   - 3.3 선택된 m개의 스칼라를 Corollary와 같이 부분 부호화된 스칼라 $\{\langle d_{j_1}(J_{j_1},t_1,t_2)\rangle, \cdots, \langle d_{j_m}(J_{j_m},t_1,t_2)\rangle\}$로 분리한다.
   - 3.4 Theorem 2를 이용하여 $\langle d_{j_1}(J_{j_1},t_1,t_2)\rangle = \cdots = \langle d_{j_m}(J_{j_m},t_m,t_2)\rangle$와 같은 경우의 수를 센다.
   - 3.5 세어진 수를 합하고 ${}_sC_m$ 개의 조합 중에서 조사해야할 수가 가장 적은 두 개의 스칼라를 찾는다.
4. 키 조사 : 알고 있는 평문 암호문 쌍을 가지고 3.5과정으로부터 얻어진 비트 패턴들의 모든 조합을 조사한다. 그러면 비밀키 값을 찾을 수 있다.

다음은 간단하게 3 AD 수열을 이용한 예제이다.

Example 2. 예를 들어 표 2를 사용하자. 가능한 경우의 수는 $10(={}_5C_3)$ 가지이다. 세 개의 스칼라로 키를 조사해야 될 수를 어떻게 결정하는지 보여준다. 먼저, $d(1)$, $d(2)$와 $d(3)$을 생각하자.

■ $d_{1,0}=d_{2,0}=d_{3,0}=0$이고 $1 \le j \le 3$에 대하여 $d_{i,j}=0$인 i의 수는 9이다. Theorem 2로부터 $\sum_{i=1}^{10} d_{1,i}2^i = \sum_{i=1}^{10} d_{2,i}2^i = \sum_{i=1}^{10} d_{3,i}2^i$ 만족하는 경우의 수는 $2(=2^{10-9})$이다. 그러나 최상위 비트가 1인 것을 알기 때문에 $d(1)$, $d(2)$와 $d(3)$를 유일하게 결정할 수 있다. 다른 스칼라의 조합에 대하여 위의 과정을 적용한 결과는 다음과 같다.

- $\{d(1),d(2),d(4)\}$와 $\{d(2),d(3),d(5)\}$의 경우, 2개의 경우의 수가 있다.
- $\{d(1),d(2),d(5)\}$ $\{d(1),d(3),d(4)\}$ $\{d(1),d(3),d(5)\}$, $\{d(2),d(3),d(4)\}$와 $\{d(2),d(4),d(5)\}$의 경우, 2^2개의 경우의 수가 있다.
- $\{d(1),d(4),d(5)\}$와 $\{d(3),d(4),d(5)\}$의 경우, 2^3개의 경우의 수가 있다.

그러므로, 얻어진 5개의 AD 수열을 이용하여 10비트 비밀값을 알아내기 위해 조사해야될 최소의 수는 $d(1)$, $d(2)$와 $d(3)$의 경우 1개이다. 즉 $d(1)$, $d(2)$와 $d(3)$를 가지고 10비트 비밀값을 유일하게 결정할 수 있다.

Corollary 1과 Theorem 2를 이용한 구현 결과 :

표 5에서 #(Testing)은 조사해야할 키의 수이고 Imp.은 키 길이가 n비트일 때 키를 찾기 위해 SPA로부터 얻은 $2^{n/2}$개의 경우의 수와 비교하여 향상된 효율성을 의미한다.

공격자가 20개의 AD 수열을 얻었을 때 예를 들면 163비트 키 경우에는 평균적으로 2^8개의 가능한 키들을 조사하면 비밀키를 찾을 수 있다는 것을 알 수 있다. 193비트 경우에는 평균적으로 2^8개의 가능한 키들을 조사해야하고, 2^{10}비트 경우에는 평균적으로 2^{13}개의 가능한 키들을 조사해야

표 5 : 공격자가 얻은 AD 수열의 수가 20일 때 3개부터 10개까지 AD 수열을 이용하여 표준의 163, 193, 233 비트에 대한 구현 결과.

m	s	비밀키 값의 길이					
		163 비트		193 비트		233 비트	
		#(Testing)	Imp.	#(Testing)	Imp.	#(Testing)	Imp.
3	20	2^{32}	2^{50}	2^{38}	2^{59}	247	2^{70}
4	20	2^{23}	2^{59}	2^{28}	2^{69}	235	2^{82}
5	20	2^{18}	2^{64}	2^{22}	2^{75}	227	2^{90}
6	20	2^{14}	2^{68}	2^{18}	2^{79}	222	2^{95}
7	20	2^{12}	2^{70}	2^{15}	2^{82}	219	2^{98}
8	20	2^{10}	2^{72}	2^{13}	2^{84}	216	2^{101}
9	20	2^{9}	2^{73}	2^{11}	2^{86}	214	2^{103}
10	20	2^{8}	2^{74}	2^{10}	2^{87}	213	2^{104}

한다.

4.2. 안전성 분석

표 1-5로부터 공격 알고리즘은 더 많은 AD 수열을 이용하는 것이 좀더 강력함을 알 수 있다. 본 논문에서 제안한 공격 알고리즘은 많은 AD 수열 없이도 2개의 AD 수열만을 이용해서 공격 가능한 장점이 있다.

본 논문에서 제안한 공격 알고리즘은 Property 1을 만족하는 일반적인 랜덤 부호화 스칼라 곱 알고리즘에 적용할 수 있다.

참고문헌

[1] J. S. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems", *In Workshop on Cryptographic Hardware and Embedded Systems (CHES' 99)*, LNCS1717, (1999),292-302.

[2] Louis Goubin, "A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems", *Public Key Cryptography (PKC 2000)*, LNCS2567, (2003), 199-211.

[3] J.C. Ha and S.J. Moon, "Randomized Signed -Scalar Multiplication of ECC to Resist Power Attacks", *In Workshop on Cryptographic Hardware and Embedded Systems (CHES'02)*, LNCS2523, (2002),551-563.

[4] M. Joye, C. Tymen, "Protections against differential analysis for elliptic curve cryptography: An algebraic approach", *In Workshop on Cryptographic Hardware and Embedded Systems(CHES'01)*, LNCS2162,(2001), 377- 390.

[5] M. Joye and J. Quisquater, "Hessian ellitpic curves and side-channel attacks", *In Workshop on Cryptographic Hardware and Embedded Systems(CHES'01)*, LNCS2162,(2001), 402-410.

[6] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", *In Advances in Cryptology-CRYPTO'96*, LNCS1109,(1996),104-113.

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis", *In Advances in Cryptology -CRYPTO'99*, LNCS1666,(1999),388-397.

[8] P. Liardet and N. Smart, "Preventing SPA/ DPA in ECC systems using the Jacobi form", *In Workshop on Cryptographic Hardware and Embedded Systems(CHES'01)*, LNCS2162, (2001),391-401.

[9] F. Morain and J. Olivos, "Speeding up the computation on an elliptic curve using addition-subtraction chains, Inform Theory Appl., vol 24,(1990),

[10] K. Okeya, H. Kurumatani and K. Sakurai, "Elliptic curves with the Montgomery form and their cryptographic applications", *Public Key Cryptography (PKC 2000)*, LNCS1751, (2000), 446-465.

[11] K. Okeya, K. Sakurai, "Power analysis brea- ks elliptic curve cryptosystems even secure against the timing attack", *Indocrypt 2000*, LNCS1977, (2000),178-190.

[12] K. Okeya, K. Sakurai, "On Insecurity of the Side Channel Attack Countermeasure Using Addition-Subtraction Chains under Distinguishability between Addition and Doubling", *Information Security and Privacy(ACISP'02)* , LNCS2384, (2002), 420-435.

[13] E. Oswald, "Enhancing simple Power-Analysis Attacks on Elliptic Curve Cryptosystems", In *Workshop on Cryptographic Hardware and Embedded Systems(CHES' 02)*, LNCS2523, (2002),82-97.

[14] E. Oswald, M. Aigner, "Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks", *In Workshop on Cryptographic Hardware and Embedded Systems(CHES'01)*, LNCS2162,(2001),39-50.

[15] K. Okeya, T. Takagi, "The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks", *Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2003 (CT-RSA 2003)*, LNCS2612, (2003), 328-342.

[16] C.D. Walter, "Breaking the Liardet-Smart Randomized Exponentiation Algorithm", *Proceedings of CARDIS'02*, USENIX Assoc, (2002), 59-68.

[17] C.D. Walter, "Security Constraints on the Oswald-Aigner Exponentiation Algorithm", *Cryptology ePrint Archive*, Report 2003/013, (2003). http//eprint.iacr.org/.