情報保護學會論文誌
第13券 第3號, 2003. 6

Folding 기법을 이용한 전력분석 공격에 대응하는 고속 스칼라 곱셈

하 재 철*, 곽 동 진**, 문 상 재**

A Fast Scalar Multiplication to Resist against Power Attacks by Folding the Scalar in Half

JaeCheol Ha*, DongJin Kwak**, SangJae Moon**

요 약

최근 스마트 카드와 같은 정보보호 장치에 대한 물리적 공격 중 소비 전력 분석을 통해 비밀키를 알아내는 전력분석 공격이 위협적이다. 본 논문에서는 전력분석 공격 중 단순 전력분석(SPA) 공격 및 차분 전력분석(DPA) 공격에 대응할 수 있는 방안을 분석하고 타원곡선 암호시스템에 대해 스칼라 곱셈의 연산량을 줄일 수 있는 방안을 제안한다. 제안하는 스칼라 곱셈 알고리듬은 DPA 공격을 방어하기 위해 비밀키에 랜덤성을 부여하였으며 SPA 공격에 대응하면서 계산 효율을 높이기 위해 비밀키를 절반으로 folding하는 기법을 사용하였다. 제안 알고리듬은 하나의 사전 계산정보를 이용하여 SPA와 DPA 공격을 방어하면서도 스칼라 곱셈 연산량을 이진 방식에 비해 약 33%정도 개선하였다.

ABSTRACT

Recently, it has been shown that cryptographic devices such as smart cards are vulnerable to power attacks. In this paper, by mixing the randomization concept and the folding in half for secret scalar integer on ECCs, we propose an efficient and fast scalar multiplication algorithm to resist against simple power analysis(SPA) and differential power analysis(DPA) attacks. Our proposed algorithm as a countermeasure against SPA and DPA is estimated as a 33% speedup compared to the binary scalar multiplication.

Keyword : *스마트 카드, ECC, 전력분석 공격, SPA, DPA, Scalar-folding method*

Ⅰ. 서 론

지금까지의 많은 암호시스템은 이론적으로 이산대수 문제나 소인수분해 문제에 그 안전성을 기반하여 개발되었다. 그러나 이러한 암호시스템을 실제 구현할 때 사전에 고려하지 못했던 부분에서 부가적으로 정보의 누출이 있다는 사실이 최근 밝혀지고 있다. 특히 스마트 카드와 같은 정보보호 장치의 경우, 비밀키와 관련된 연산과정에서 다양한 종류의 비밀정보의 누출이 있을 수 있다. 이와 같이 부가 정보를 이용한 공격 방법을 부-채널공격(side-channel attack)[1]이라 하며 크게 능동적 공격과 수동적 공격으로 나누어진다. 능동적 공격에는 결함주입 공격(fault insertion attack)[2,3]이 있으며 수동적 분석 공격에는 시간 공격(timing attack)과 전력분석 공격(power analysis attack)[4-6]이 대표적이다. 특히 전력분석 공격은 적은

** 나사렛대학교 정보통신학과(jcha@kornu.ac.kr)
*** 경북대학교 대학원 전자공학과(neverdid@palgong.knu.ac.kr, sjmoon@knu.ac.kr)

비용과 노력으로도 공격에 성공할 수 있어 현실적인 공격법으로 주목받고 있다. 전력분석 공격은 단순히 전력신호를 관찰함으로써 사용된 비밀키를 알아내는 단순 전력분석(Simple Power Analysis, SPA) 공격과 SPA에 통계학적 방법과 에러 정정 기법을 사용하는 차분 전력분석(Differential Power Analysis, DPA) 공격으로 나누어진다.

본 논문에서는 스마트 카드 등에 사용되는 공개키 암호 시스템 중 짧은 키 길이를 이용하면서도 처리 속도와 대역폭 그리고 설계용량 면에서 우수한 타원곡선 암호시스템(Elliptic Curve Cryptosystem, ECC)[7,8]에서의 전력분석 공격을 살펴본다. 지금까지 타원곡선 암호시스템에 대해 SPA와 DPA에 대응하기 위한 대처 방법들이 많이 제시되었다[9-13]. 그러나 제안된 대응방법들의 특징 중 하나는 SPA에 대응하도록 하기 위해서 불필요한 연산(dummy operation)을 추가했다는 것인데 이는 연산 속도를 저하시키는 결과를 가져왔다.

본 논문에서는 타원곡선 암호 시스템에서 전력분석 공격에 대응하면서 고속으로 스칼라 곱셈을 구현할 수 있는 방법을 제안하고자 한다. 제안 방식의 주된 내용은 비밀키에 랜덤성을 부여하여 DPA 공격에 대응하면서 비밀키를 절반으로 접는 방법(scalar-folding method)을 사용하여 고속으로 동작할 뿐만 아니라 SPA 공격에 대응하도록 한 것이다.

논문의 제 2장에서는 ECC 시스템의 연산에 대해 간단히 알아보고, 3장에서는 전력분석 공격과 기존의 대응책을 살펴본다. 제 4장에서 전력분석 공격에 대한 새로운 대처 방안을 제시하며 5장에서 기존의 방식과 비교 분석한 후, 마지막으로 결론을 맺는다.

II. 타원곡선 암호시스템에서 스칼라 곱셈

타원곡선 암호시스템은 1985년 Koblitz와 Miller에 의해 각각 독립적으로 제안되었다. ECC의 장점으로는 현재 가장 많이 사용되고 있는 RSA와 동일한 암호학적 안전도를 유지하기 위한 키 길이가 현저히 작고 연산이 효율적이라는 점이다. ECC 시스템에서 사용되는 비밀키 160비트는 RSA시스템에서 사용하는 1024 비트의 비밀키와 비슷한 안전도를 제공하는 것으로 알려져 있다. 따라서 ECC는 저장 용량 및 대역폭(bandwidth)의 제한이 있는 스마트 카드나 무선 통신 등에 유용하게 사용될 수 있다.

타원곡선 상에서의 중요한 연산은 타원곡선상에 주어진 한 점 P를 k번 더하는 스칼라 곱셈(scalar multiplication)이며 kP로 표기한다. 본 논문에서는 스칼라 정수 k를 비밀키로 한정하고 공격자가 공격하고자 하는 최종적인 정보라 가정한다.

스칼라 곱셈 kP의 계산은 정수 k의 이진 표현법을 이용한 Binary 방법을 통하여 얻을 수 있는데 이를 구체적으로 나타낸 것이 [그림 1]이다.

```
Algorithm 1 : Binary(Left-to-Right)
Output : Q = kP
1.1 Q = O
1.2 for i = n-1 to 0 by -1 do {
1.3     Q = 2Q
1.4     if ( k_i == 1) then Q = Q + P }
1.5 Return Q
```

[그림 1] Binary 스칼라 곱셈 알고리즘

여기서 O는 타원곡선상의 무한 원점을 나타낸다. 또 n비트의 k를 이진수 표현으로 나타내면 다음과 같이 쓸 수 있다.

$$k = \sum_{i=0}^{n-1} k_i 2^i, \quad k_i \in \{0, 1\}$$

위의 알고리즘에서 $2Q$와 $Q+P$의 계산을 각각 두배(doubling) 연산과 덧셈(addition)연산이라 한다. 이와 같은 스칼라 곱셈의 속도를 개선하기 위한 여러 가지 방법들이 제시되었는데 두 점의 덧셈 계산량은 뺄셈과 거의 동일한 연산량을 가진다는 점에 착안되어 addition-subtraction 방법이 많이 사용되고 있다. 이 알고리듬은 비밀키 k를 부호화된 이진수(signed binary) d로 확장하여 사용한다. 즉, d는 다음과 같이 표현할 수 있다.

$$d = \sum_{i=0}^{n} d_i 2^i, \qquad d_i \in \{\bar{1}, 0, 1\}$$

여기서 $\bar{1}$은 -1을 의미한다. 특히, 일반적인 이진수를 부호화된 이진수로 바꾼 수열 중에서 모든 i에 대해 인접한 두 비트 중 적어도 한 비트가 0이 되는 형태를 NAF(Non-Adjacent Form)이라 한다. 즉, $d_i\, d_{i+1} = 0$가 되는 형태를 의미하는데 모든 양의 정수는

유일한 NAF를 갖는다. 또한, 어떤 정수의 NAF는 0이 아닌 1이나 $\bar{1}$의 개수가 가정 적은 부호화된 이진 표현 방법이 되는데 '0'이 아닌 비트 수를 일반적인 이진 표현 방법에 비해 약 33%까지 줄일 수 있다[14,15]. 스칼라 곱셈에서 부호화된 이진수 혹은 NAF 이진수를 이용한 addition-subtraction 알고리듬을 나타낸 것이 [그림 2]이다.

Algorithm 2 : Addition-Subtraction
Output : $Q = dP$
2.1 $Q = O$
2.2 for $i = n$ to 0 by -1 {
2.3 $Q = 2Q$
2.4 if ($d_i == 1$) then $Q = Q + P$
2.5 if ($d_i == \bar{1}$) then $Q = Q - P$}
2.6 Return Q

〔그림 2〕 Addition-Subtraction 알고리즘

Ⅲ. 전력분석 공격 및 대응 방법

전력분석 공격은 스마트 카드와 같은 정보보호 장치가 구동시 누출되는 부-채널 정보 중 소비 전력을 측정하고 이를 분석하여 비밀정보를 알아내는 공격법이다. 전력분석 공격은 내부의 비밀키와 관련한 연산시 직접 소비전력 신호의 특성을 파악하여 비밀키에 대한 정보를 알아내는 SPA와 SPA에 통계적인 분석방법과 에러 정정 기술을 첨가한 DPA로 나누어질 수 있다.

ECC의 경우를 예를 들면, [그림 1]과 같은 Binary 방법을 이용하여 스칼라 곱셈을 할 경우 한 점을 두 배하는 연산($2Q$)과 두 점을 더하는 연산($P+Q$)의 소비전력이 다르게 나타날 것이다. 따라서 이러한 연산 명령의 수행시 소비되는 전력의 특성을 파악하여 [그림 1]이나 [그림 2]와 같은 조건문(if 문)에 따른 알고리듬의 수행과정을 추적하는 공격이다. 이를 방어하기 위해서는 스마트 카드 시스템 개발자에 의해 소비전력 누출을 막는 기술을 사용하거나 비밀키 연산 시 소비 전력이 비밀 정보에 의존하지 않도록 조건문 등이 없는 알고리듬을 구현함으로써 방어할 수 있다. [그림 3]은 [그림 1]의 Binary 방법을 SPA에 대응하도록 변형한 것이다[10]. 그러나 이 알고리듬은 불필요한 덧셈 연산이 약 $0.5n$번 추가되므로 계산량이 많아진다는 단점을 가지고 있다.

Algorithm 3 : SPA resistant Binary
Output : $Q[0] = kP$
3.1 $Q[0] = O$
3.2 for $i = n-1$ to 0 by -1 do {
3.3 $Q[0] = 2Q[0]$
3.4 $Q[1] = Q[0] + P$
3.5 $Q[0] = Q[k_i]$ }
3.6 Return $Q[0]$

〔그림 3〕 SPA 대응하는 Binary 알고리즘

DPA는 SPA보다 방어하기 더 어려운 강력한 분석 방법이다. DPA는 기존의 SPA의 소비 전력을 관찰하는 것에 더하여 비밀키와 정확히 상관관계(correlation)를 가지는 정보를 추출하고 이를 분석함으로써 비밀 키를 공격하는 방법이다. DPA의 구현은 다음의 두 단계로 나눌 수 있다. 먼저 데이터 수집 단계로 스마트 카드가 암호학적 연산을 실행 시에 소비되는 전력을 표본화(sampling)하여 그 데이터를 수집한다. 두번째는 데이터 수집 후 실시하는 데이터 분석 단계로서 표본화한 데이터의 잡음신호(noise signal)를 감소시키고 차분신호(differential signal)의 명확성을 높이기 위해 디지털 신호 해석과 통계적인 방법을 이용한다.

DPA 공격의 대응방법으로 Coron은 3가지의 대응책을 제안하였다[10]. Coron의 대응책의 핵심은 $Q = kP$ 계산시 다양한 형태의 랜덤 기법을 도입하여 DPA 공격에 결정적인 역할을 하는 분류함수(partitioning function)가 비밀키 정보와 어떠한 상관 관계도 없게 만드는 것이다. 이 대응책들은 알고리듬 구현시 약간의 비용증가와 덧붙여지는 잉여 정보가 있으나 DPA 공격을 효과적으로 방어할 수 있게 하는 방법들이다.

DPA 공격에 대응하기 위한 다른 방법 중 하나는 비밀키가 연산에 사용될 때 동일한 비밀키 연산과정을 거치는 것이 아니라 매번 다른 연산 과정을 진행하도록 비밀키를 재부호화(recoding)하여 사용하는 것이다[13,16]. 이렇게 연산과정을 달리함으로써 연산에 사용되는 중간값들이 매번 달라지게 만들어 DPA 공격을 방어한다. 특히, 문헌 [13]에서는 NAF으로 재부호화하는 과정에 랜덤 성분을 삽입하여 비밀키가 사용되는 연산은 매번의 스칼라 곱셈시 다른 연산과정

을 거치도록 하였다. 문헌에서 사용되었던 재부호화 방법은 [표 1]과 같다. 표에서 k는 계수가 0과 1만 있는 이진수이며 검색은 k의 하위 비트부터 두 비트씩을 고려하여 수행한다. 여기서 r_i는 랜덤 비트이고 t_i는 임시 기억 값이다. 이 과정을 거쳐 계수가 0, 1, 그리고 $\bar{1}$로 구성된 d가 재부호화된 결과값이다. 이와 같은 재부호화 방법을 사용하여 생성된 부호화된 비밀키 d를 이용하면 매번 스칼라 곱셈 연산 과정이 달라져 DPA 공격을 효과적으로 방어할 수 있다.

물론 문헌 [13]에서 사용된 최종 알고리듬은 비밀키의 연산과정에서 측정되는 소비 전력이 비밀 정보에 의존하지 않도록 구현하여 SPA에 대응하도록 하였다. 그러나 여기에서도 SPA에 대응하도록 알고리듬을 구현하기 위해 부가적인 연산을 필요로 하게 되고 n비트의 비밀키를 사용할 경우 각각 $n+1$번의 덧셈연산과 두배 연산이 필요하게 된다. 이 결과는 기존의 SPA나 DPA를 대응하기 전의 계산량보다 많아져서 결국 전력 분석공격에 대응하기 위해서는 계산량의 손실을 수반할 수 밖에 없었다. 따라서 본 논문의 핵심은 비밀키 k가 d로 재부호화된 경우를 가정하고 계산량을 더 줄이는 방법을 제안하고자 하는 것이다.

〔표 1〕 비밀키의 랜덤 재부호화 방법[13]

입 력				출 력		비 고
k_{i+1}	k_i	t_i	r_i	t_{i+1}	d_i	
0	0	0	0	0	0	×
			1	0	0	×
0	0	1	0	0	1	NAF
			1	1	$\bar{1}$	AF
0	1	0	0	0	1	NAF
			1	1	$\bar{1}$	AF
0	1	1	0	1	0	×
			1	1	0	×
1	0	0	0	0	0	×
			1	0	0	×
1	0	1	0	1	$\bar{1}$	AF
			1	0	1	NAF
1	1	0	0	1	$\bar{1}$	AF
			1	0	1	NAF
1	1	1	0	1	0	×
			1	1	0	×

Ⅳ. 고속 스칼라 곱셈 방법 제안

4.1 Scalar-folding을 이용한 곱셈 알고리듬

비밀키 k가 n비트일 경우 [표 1]에 의해 재부호화된 비밀키 d는 최대 $n+1$비트가 된다. 랜덤수에 의해 재부호화된 비밀키를 다시 표현하면 아래와 같다.

$$d = d_n 2^n + d_{n-1}2^{n-1} + \ldots\ldots\ldots + d_1 2^1 + d_0 2^0$$

이 경우 $d_i \in \{\bar{1}, 0, 1\}$이다. 이와 같이 표현된 비밀키 d를 이용하여 다음과 같이 두 번째 부호화 과정을 거치게 된다. 제안하는 고속 스칼라 곱셈 알고리듬은 비밀키 d를 절반으로 분할하여 상·하위 두 부분으로 나누며 다음과 같이 쓸 수 있다. 이 경우 $h = \lceil (n+1)/2 \rceil$ 이다.

$$d = 2^h(e_{h-1}2^{h-1} + e_{h-2}2^{h-2} + \ldots + e_0 2^0) + (f_{h-1}2^{h-1} + f_{h-2}2^{h-2} + \ldots + f_0 2^0)$$

여기서 $e_i,\ f_i \in \{\bar{1}, 0, 1\}$이며 $i = 0,1,\ldots,h-1$일 경우는 $f_i = d_i,\ e_i = d_{i+h}$이다.

제안하는 스칼라 곱셈에서는 d를 다음과 같이 표현한다.

$$Q = dP = \sum_{i=0}^{h-1} (2^i)(e_i(2^h P) + f_i P)$$

그리고 제안 방식에서는 계산 효율을 높이기 위해 $2^h P$를 미리 계산해 두기로 한다. 이는 고정된 한 점 P와 비밀키의 길이가 주어지면 사전계산이 가능하다.

이러한 개념은 문헌 [17]에서 처음 도입되었으며 이후 멱승(exponentiation) 연산에서 비밀키 접는 방식으로 비슷하게 적용되어 왔는데 기존 방식들은 이진 표현법에 기초하여 RSA와 같은 연산에 이용되었다. 그러나 본 논문에서와 같이 비밀키가 랜덤 부호화된 이진수(random signed binary)인 경우에 적용된 예는 없다.

이와 같이 비밀키 d를 상위와 하위로 분할한 후에는 다음과 같이 재부호화된다.

$$g_i' = e_i 3^1 + f_i 3^0,$$
$$g_i' \in \{\bar{4}, \bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3, 4\},$$
$$g_i = |g_i'|,\ g_i \in \{0, 1, 2, 3, 4\}$$

또한 $g_i'=0$이면 $s_i=0$로, $g_i'>0$이면 $s_i=1$로 $g_i'<0$이면 $s_i=\bar{1}$로 둔다. 여기서 주의할 것은 g_i'가 비밀키 값 d와 동일한 의미를 가지는 부호화가 아니라 단지 연산에 필요한 정보가 표시된 배열의 위치만을 나타내는 값이다는 점이다. 연산에 필요한 정보가 저장된 배열 내용은 다음과 같다.

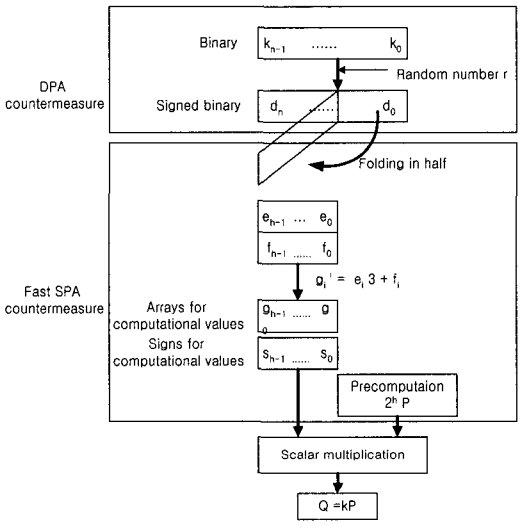
$P[0]=P$, dummy 정보:($e_i=f_i=0$일때 덧셈)
$P[1]=P$:($e_i=0$이고 $f_i=1$ 일때 덧셈)
$P[\bar{1}]=-P$:($e_i=0$이고 $f_i=\bar{1}$ 일때 덧셈)
$P[2]=(2^h-1)P$:($e_i=1$이고 $f_i=\bar{1}$ 일때 덧셈)
$P[\bar{2}]=(-2^h+1)P$:($e_i=\bar{1}$이고 $f_i=1$일때 덧셈)
$P[3]=2^hP$:($e_i=1$이고 $f_i=0$ 일때 덧셈)
$P[\bar{3}]=-2^hP$:($e_i=\bar{1}$이고 $f_i=0$ 일때 덧셈)
$P[4]=(2^h+1)P$:($e_i=1$이고 $f_i=1$ 일때 덧셈)
$P[\bar{4}]=(-2^h-1)P$:($e_i=\bar{1}$이고 $f_i=\bar{1}$일때 덧셈)

비밀키를 부호화하여 g_i와 s_i가 구해지면 이를 통해 $Q=dP$를 수행하는 알고리듬은 [그림 4]와 같다. 알고리듬 4의 내용을 보면 먼저 사전 계산을 통해 2^hP를 계산하여 $P[3]$에 저장해 두고 연산에 필요한 정보를 각 배열에 저장한다. 단계 4.2에서부터 4.7까지는 g_i와 s_i에 따라 스칼라 곱셈을 실시하는데 총 h번의 loop를 반복한다.

Algorithm 4: SPA/DPA resistant Scalar-folding
Output : $Q=dP$
----- Precomputation Phase($P[3]=2^hP$)----
P.1 $P[3]=P$
P.2 for $i=h-1$ to 0 by -1 do {
P.3 $P[3]=2P[3]$ }
----- Evaluation Phase ----------
4.1 $P[0]=P$, $P[1]=P$,
$P[2]=P[3]-P[1]$, $P[4]=P[3]+P[1]$
4.2 for $i=h-1$ to 0 by -1 do {
4.3 $Q[0]=2Q[0]$ // Doubling
4.4. $R[0]=R[1]=P[g_i]$ // Positive or 0
4.5 $R[\bar{1}]=-P[g_i]$ // Negative
4.6 $Q[1]=Q[\bar{1}]=Q[0]+R[s_i]$ //Addition
4.7 $Q[0]=Q[s_i]$ } // Selection
4.8 Return $Q[0]$

〔그림 4〕 제안하는 Scalar-folding을 이용한 스칼라 곱셈

[그림 5]는 지금까지 설명한 folding 기법을 이용한 전력분석 공격 대응 알고리듬을 단계적으로 도시한 것이다. 그림에서 이진 비밀키 k를 랜덤 수를 이용하여 부호화된 이진수로 바꾸는 과정은 DPA 공격에 대응하기 위해 [표 1]의 알고리듬을 이용한 것이다.



〔그림 5〕 새로운 스칼라 곱셈 알고리듬 수행단계

이 부호화된 이진수는 상위 부분과 하위부분으로 나누어져 계산에 필요한 값을 저장하는 배열 위치 정보 g_i와 부호 정보 s_i로 재부호화된다. 마지막으로 사전 계산된 값 2^hP와 [그림 4]의 알고리듬을 이용하여 스칼라 곱셈을 수행하면 최종 결과인 $Q=kP$ 값이 계산된다.

4.2 Numerical Example

본 절에서는 상기한 SPA/DPA에 대응하는 스칼라 곱셈 알고리듬을 수치적인 예를 통해 설명한다. 여기서 비밀키 $k=(111011110)$라 하고 랜덤 수 $r=(101010011)$이라 하여 [표 1]에 의해 재부호화를 수행할 때 d는 아래와 같이 표현된다.

$k=(111011110)=2^8+2^7+2^6+2^4+2^3+2^2+2^1$
$=478$
$r=(101010011)$
$d=(1000\bar{1}00\bar{1}10)=2^9-2^5-2^2+2^1=478$
여기서 $n=9$이므로 $h=5$이다.

따라서 랜덤 비밀키 d에 의해 e, f, g 그리고

〔표 2〕 연산량 비교 분석(○ : 제공, ×: 제공 않음, △: 추가 연산이 필요하거나 별도의 DPA 대응책과의 혼용이 필요)

Algorithm	SPA	DPA	Additions	Doublings	A=D	D=0.7A	Point 임시 저장장소수
Binary[18]	×	×	n/2(avg)	n	1.5n(avg)	1.2n	2
NAF[14]	×	×	n/3(avg)	n+1	1.33n(avg)	1.033n	2
Coron's[10]	○	△	n	n	2n	1.7n	3
Möller's(min. window)[11]	○	△	n/2	n	1.5n	1.2n	5
Hichcock-Montague's[12]	○	△	5n/9	10n/9	1.667n	1.333n	4
Ha-Moon's[13]	○	○	n+1	n+1	2n+2	1.7n	6
Proposed method	○	○	$\lceil (n+1)/2 \rceil + 2$	$\lceil (n+1)/2 \rceil$	n+3	0.85n	11

s는 아래와 같다.

$$e=(1000\bar{1})$$
$$f=(00\bar{1}10)$$
$$g=(30113)$$
$$s=(10\bar{1}1\bar{1})$$

[그림 4]에 주어진 알고리듬에 의해 스칼라 곱셈을 수행할 경우 사전 계산 값은 $P[3]=2^5P$이 되고 나머지 사전 계산 값은 아래와 같다.

$$P[0]=P,\ P[1]=P,\ P[2]=(2^5-1)P=31P,$$
$$P[3]=2^5P,\ P[4]=(2^5+1)P=33P$$

또한 [그림 4]의 "for"문에서 각 loop에서의 계산 값은 다음과 같다.

loop $i=4$: $Q[0]=2^5P=32P$
loop $i=3$: $Q[0]=2(32)P=64P$
loop $i=2$: $Q[0]=(2(64)-1)P=127P$
loop $i=1$: $Q[0]=(2(127)+1)P=255P$
loop $i=0$: $Q[0]=(2(255)-32)P=478P$

따라서 최종 결과는 $Q=kP$와 같으며 랜덤 수 r에 따라 d값이 바뀌므로 연산과정은 매번 달라지게 된다. 이러한 결과는 DPA 공격시 필요한 [그림 4]의 중간 값 $Q[0]$가 고정적이지 않기 때문에 분류함수를 통하여 비밀키와 전력 측정 결과가 연관성을 갖지 못하도록 했기 때문이다. 물론 [그림 4]에서 보는 바와 같이 조건문이 없으므로 비밀키와 연산 수행 과정이 독립적임을 알 수 있다. 따라서 SPA 공격에도 대응할 수 있다.

V. 비교 분석

본 장에서는 제안된 스칼라 곱셈 방법과 기존의 방법을 비교 분석한다. [표 2]는 지금까지의 여러 알고리듬의 계산량을 기술하였다. 여기서 비밀키를 2차에 걸쳐 재부호화를 하지만 스칼라 곱셈이 진행되기 전 한번만 수행하고 소비되는 시간도 스칼라 곱셈 전체에 비해 극히 작은 부분이므로 비교 대상에서는 제외하였다. 먼저 SPA와 DPA의 방어대책은 아니지만 이진(Binary) 방식은 평균 $n/2$의 덧셈과 n번의 두배 연산이 요구된다. 이를 개선하여 비밀키를 NAF 형태로 재부호화한 경우에는 "0"이 아닌 비트 수가 전체의 약 1/3로 줄어 덧셈 수를 $n/3$으로 줄일 수 있다.

[표 2]에서 Coron의 방법, Möller의 방법 그리고 Hichcock-Montague 방법은 SPA에 대한 방어 대책으로 제안되었는데 DPA에 대한 방어대책을 세우기 위해서는 별도의 추가적인 연산이 필요하거나 특별한 언급이 없어 다른 DPA 방어 대책과 혼용하여 사용할 수 밖에 없다. 위의 3가지 방법 중에서는 Möller 방법이 Binary 방법과 거의 비슷한 계산 속도를 내는 것으로 알려져 있는데 고정 윈도우 방법을 사용하므로 최대 5개의 점을 저장할 수 있는 메모리가 필요하다.

지금까지 제안된 방법 중 SPA와 DPA를 동시에 방어할 수 있는 방법인 Ha-Moon 방법은 각각 $n+1$번의 덧셈과 두배 연산이 필요하게 되어 비교적 연산 시간이 늦다는 단점이 있다.

제안하는 스칼라 곱셈 방법은 SPA와 DPA를 방어할 수 있는 방법으로서 2^hP는 사전에 계산되었다

고 가정한다. 이 경우 $\lceil (n+1)/2 \rceil + 2$번의 덧셈과 $\lceil (n+1)/2 \rceil$번의 두배 연산이 필요하게되어 가장 빠른 속도로 스칼라 곱셈을 구현할 수 있다. 만약 2^hP를 사전에 계산하여 저장하지 않는다 하더라도 $h = \lceil (n+1)/2 \rceil$번의 두배 연산만이 추가적으로 필요하다. 이 경우 계산량은 Möller 방법과 비슷한 계산량을 가진다. 그러나 제안 방식은 2^hP를 저장하기 위한 하나의 고정된 저장 공간과 연산과정에서 약 11개의 점을 저장하기 위한 임시 메모리가 필요하여 타 방식에 비해 메모리가 좀 더 필요하다는 것이 단점이다.

Ⅵ. 결 론

본 논문에서는 부-채널 공격 중 전력분석 공격에 대응할 수 있는 원리와 제안된 여러 가지 알고리듬을 분석하였다. 전력분석 공격 중 SPA와 DPA공격을 동시에 방어할 수 있으면서 연산 효율을 높이는 방법을 찾는 것이 본 논문의 핵심적인 내용이다. 특히 SPA를 방어하기 위해서는 비밀키 연산시 소비되는 전력이 비밀 정보에 의존하지 않도록 조건문 등이 없도록 알고리듬을 구현해야 한다. 이를 위해서는 실제 연산과 관련이 없는 불필요한 추가 연산이 필요하게 되고 이는 전체적인 연산속도를 저하시키는 요인이 되어 왔다.

본 논문에서는 이를 개선하기 위하여 사전 계산된 하나의 점(2^hP)을 이용하여 계산 효율을 높이고자 하였다. DPA 공격을 방어하기 위해 1차적인 재부호화 과정을 거쳐 비밀키를 랜덤화하여 계산 경로를 매번 다르게 하였다면 2차 부호화 과정은 하나의 사전 계산 값을 이용하여 계산 효율을 높이면서 SPA 공격에 대응하도록 하였다.

위에서 분석된 바와 같이 제안 알고리듬은 전력분석공격에 대응하면서 계산량을 Binary 방법보다 약 2/3수준까지 줄일 수 있으며 SPA/DPA 공격에 대한 방어 방법으로 알려진 Ha-Moon 방법보다 약 1/2정도까지 줄일 수 있다.

참 고 문 헌

[1] J. Keley, B. Schneier, D. Wagner, and C. Hall, "Side channel cryptanalysis of product cipher", *in Proceedings of ESORICS'98*, pp.97~110, Springer-Verlag, Sep. 1998.

[2] Bellcore Press Release, "New threat model breaks crypto codes", Sep. 1996 or D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," *In Advances in Cryptology-EUROCRYPT '97, LNCS 1233*, pp.37~51, Springer-Verlag, 1997.

[3] S. M. Yen, S. J. Kim, S. G. Lim, and S. J. Moon, "A countermeasure against one physical cryptanalysis May Benefit Another Attack", *In Proc. of the ICISC 2001*, Korea. Dec. 2001

[4] P. Kocher, J. Jaffe, and B. Jun, "Introduction to differential power analysis and related attacks", *http://www.cryptography.com/dpa/technical*, 1998.

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis", *in Advances in Cryptology-CRYPTO'99*, pp.388~397, Springer-Verlag, 1999.

[6] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks on modular exponentiation in Smart cards", *in Proc. of Workshop on Cryptographic Hardware and Embedded Systems*, pp.144~157, Springer-Verlag, 1999.

[7] N. Koblitz, "Elliptic curve crypto-systems", *Math. of Computation*, vol.48, pp.203~209, 1987.

[8] V. Miller, "Uses of elliptic curves in crypto in cryptography", *in Proc. of Advances in Cryptology-CRYPTO' 85*, pp.417~426, Springer-Verlag, 1985.

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

[10] J. S. Coron, "Resistance against differential power analysis for Elliptic Curve Cryptosystems", i*n Proc. of Workshop on Cryptographic Hardware and Embedded Systems*, pp.292~302, Springer-Verlag, 1999.

[11] B. Möller, "Securing elliptic curve point multiplication against side- channel attacks", *In Information Security Conference-ISC'01, LNCS 2200*, pp.324~334, 2001.

[12] Y. Hitchcock and P. Montague, "A new elliptic curve scalar multiplica- tion algorithm to resistant simple power analysis", *in Proc. of Information Security and Privacy-ACISP2002, LNCS 2384*, pp.214~225, Springer-Verlag, 2002.

[13] J. C. Ha and S. J. Moon, "Random ized signed-scalar multiplication of ECC to resist power attacks", *in Proc. of Workshop on Cryptographic Hardware and Embedded Systems-CHES 2002*, pp.553~565, Springer-Verlag, 2002.

[14] C. N. Zhang, "An improved binary algorithm for RSA", *Computers Math. Application*, Vol. 25, No. 6, pp.15~24, 1993.

[15] O. Egecioglu and C. K. Koc, "Exponentiation using canonical recording", *Theoretical Computer Science*, Vol. 129, No. 2, pp.407~417, 1994.

[16] E. Oswald and M. Aigner, "Randomized addition-subtraction chains as a countermeasure against power attacks", i*n Workshop on Cryptographic Hardware and Embedded Systems*, pp.40-52, Springer-Verlag, 2001.

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

[18] D. E. Knuth, *The art of computer programming*, Vol 2: Seminumerical algorithms, Reading, MA: Addison- Wesley, 2nd Edition, 1981.

〈著者紹介〉

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

곽 동 진 (DongJin Kwak) 학생회원
1998년 2월 : 경북대학교 전자공학과 졸업(학사)
2000년 2월 : 경북대학교 대학원 전자공학과 졸업(석사)
2000년 3월~현재 : 경북대학교 대학원 전자공학과 박사과정
<관심분야> 공개키 암호 프로토콜, 이동네트워크 정보보호

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