Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.3, Jun. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.3.587

부채널 공격에 대응하는 새로운 스칼라 레코딩 방법*

유 효 명,[1†] 조 성 민,[1] 김 태 원,[1] 김 창 한,[2] 홍 석 희[1‡]
[1]고려대학교, [2]세명대학교

A New Scalar Recoding Method against Side Channel Attacks*

Hyo Myoung Ryu,[1†] Sung Min Cho,[1] TaeWon Kim,[1] Chang han Kim,[2]
Seokhie Hong[1‡]
[1]Korea University, [2]Semyung University

요 약

본 논문에서는 SPA와 DPA 모두에 안전한 스칼라 레코딩 방법을 제안한다. 제안하는 방법은 스칼라 레코딩을 이용한 전력분석공격의 대응방법으로써 음수 표현을 사용한 스칼라 레코딩 방법이다. 제안하는 방법은 각 digit에 대해 모두 동일한 패턴의 연산을 수행하도록 레코딩하여 SPA에 안전하다. 또한 랜덤수에 따라 랜덤한 레코딩 결과를 생성하도록 하여 DPA에 안전하다. 그리고 타원곡선 덧셈 연산이 digit의 부호에 대한 SPA에 안전하도록 사전연산 테이블과 변형한 타원곡선 덧셈 알고리즘을 적용한다. 제안하는 방법은 단독 사용으로 SPA와 DPA 모두에 안전하므로 보다 효율적인 안전성을 제공한다. 제안하는 방법을 사용하면 SPA와 DPA에 안전한 기존의 스칼라 레코딩에 비해 연산효율이 11% 이상 향상된다.

ABSTRACT

In this paper we suggest method for scalar recoding which is both secure against SPA and DPA. Suggested method is countermeasure to power analysis attack through scalar recoding using negative expression. Suggested method ensures safety of SPA by recoding the operation to apply same pattern to each digit. Also, by generating the random recoding output according to random number, safety of DPA is ensured. We also implement precomputation table and modified scalar addition algorithm for addition to protect against SPA that targets digit's sign. Since suggested method itself can ensure safety to both SPA and DPA, it is more effective and efficient. Through suggested method, compared to previous scalar recoding that ensures safety to SPA and DPA, operation efficiency is increased by 11%.

Keywords: Elliptic curve cryptosystem, Power analysis attack, SPA, DPA, Scalar recoding

I. 서 론

최근 사회는 생활의 모든 것들이 연결되는 초연결사회로 나아가고 있다. 이러한 초연결사회의 기반이 되는 것이 사물인터넷(Internet of Things : IoT)이다. 사물인터넷은 각종 사물(전자제품, 모바일 장비, 웨어러블 컴퓨터 등)에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술을 의미한다. 사물인터넷 기술을 통해 생활의 각종 사물들이 모두 인터넷으로 연결되어 사용자 중심의 지능형 서비스를 제공하게 된다. 이러한 연결 관계 속에서 사물들은 거대한 정보를 생성, 수집, 공유하게 된다. 거대한 정보들은 사물에 저장되고 사물 간의 인터넷 통신을 통하여 공유되기 때문에 해킹의 대상이 될 수 있다. 게

Received(03. 08. 2016), Modified(04. 08. 2016), Accepted(04. 14. 2016)
* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었습니다.(IITP-2016-R0992-16-1017)
† 주저자, ryu7242@naver.com
‡ 교신저자, hsh@cist.korea.ac.kr(Corresponding author)

다가 사물인터넷 기술이 발달하면서 인터넷을 통해 연결되는 사물의 수가 기하급수적으로 증가하고 있다. 즉, 보호해야 할 대상의 수가 증가하고, 대상의 특성들이 다양해져 보안의 위협이 가중되고 있다. 그러므로 사물인터넷의 사물들에 대한 보안 기술이 필수적이다. 특히, 사물들이 저장된 데이터를 서로 공유하기 위해서는 사물 간의 인증 및 키 교환은 반드시 필요한 보안 기능이다. 인증 및 키 교환에는 공개키 암호시스템이 적합하다. 대표적인 공개키 암호시스템은 RSA 암호시스템과 타원곡선 암호시스템이 있다[17][18]. 타원곡선 암호시스템은 RSA 암호시스템에 비해 짧은 키 길이로 RSA 암호시스템과 동일한 안전성을 제공한다. 그러므로 저전력, 저성능의 기기환경을 필요로 하는 사물인터넷에는 타원곡선 암호시스템이 적합하다.

1996년 Paul Kocher에 의해 부채널 분석이 소개되었다[1]. 부채널 분석은 암호알고리즘의 취약점이 아닌 암호알고리즘이 장치에서 동작할 때 발생하는 시간, 전력소모량, 전자기파 등의 물리적 정보를 이용한 공격방법이다[1][2][19]. 수학적으로 공격이 어렵다고 알려진 암호시스템도 부채널 정보를 이용하여 공격이 가능해졌다. 그러면서 최근에는 스마트카드와 같은 저전력 장치에 대한 부채널 분석 연구가 활발히 진행되고 있다. 사물인터넷의 기기들은 외부로 노출되어 있기 때문에 부채널 분석은 사물인터넷의 보안 위협요소가 될 수 있다. 따라서 부채널 분석에 대한 안전성은 반드시 고려되어야 한다. 부채널 분석 중에서도 전력분석을 통한 대표적인 공격방법은 단순전력분석(Simple Power Analysis : SPA)과 차분전력분석(Differential Power Analysis : DPA)이 있다[2]. SPA는 하나의 전력소비파형을 이용하여 분석하는 방법이다. SPA는 단순히 전력소비파형을 관찰하여 수행 연산의 구분으로부터 비밀값을 추출하는 방법이다. DPA는 다수의 전력소비파형을 이용하여 분석하는 방법이다. DPA는 연산의 중간 계산 값의 추측과 통계적인 분석법을 이용하여 비밀 값을 알아내는 방법이다.

타원곡선 암호시스템의 안전성과 속도에 가장 큰 영향을 미치는 연산은 타원곡선 스칼라 곱셈 연산이다. 이러한 타원곡선 스칼라 곱셈 연산이 스칼라와 수행 연산 사이에 관계가 있어 타원곡선 암호시스템은 전력분석공격의 대상이 될 수 있다. 타원곡선 스칼라 곱셈 연산은 타원곡선 덧셈 연산과 타원곡선 두 배 연산으로 이루어져있어 두 연산을 구분하는 것으로 SPA 공격이 가능하다. 타원곡선 암호시스템에 대한 SPA에 대응하기 위해 연산 패턴을 구분하더라도 비밀 값의 정보를 알 수 없도록 비밀 값과 관계없이 항상 같은 패턴의 연산을 수행하는 다양한 대응방법들이 연구되었다. 대표적으로 Montgomery ladder 방법, Double and add always 방법, Protected odd-only recoding방법 등이 있다[3][4][15][16]. 타원곡선 암호시스템에 대한 DPA는 비밀 값의 일부를 추측하여 분석한다. 해당 데이터와 추측한 비밀 값의 연산 결과로 중간 계산 값을 예측한 뒤 통계치를 계산한다. DPA에 대한 대응방법은 연산의 중간 계산 값을 랜덤화하여 통계치가 부정확하도록 하는 방법이 있다. 중간 계산 값을 랜덤화하기 위해 비밀 값 또는 좌표 값 등을 랜덤화하는 대응방법들이 연구되었다[3][5]. 그 중에도 coordinate를 랜덤화하는 Random projective coordinates를 사용하는 방법이 가장 일반적이다. 최근에는 스칼라 레코딩을 이용하여 SPA 또는 DPA에 대응하는 방법들이 제안되고 있다[4][6][9][10][11][22].

본 논문에서는 NAF 방법을 기반으로 하는 새로운 스칼라 레코딩 방법을 제안한다. 제안하는 방법은 스칼라를 올림수 표현방법을 사용하여 랜덤한 형태로 레코딩하는 방법이다. 제안하는 방법을 적용하면 일정한 연산패턴의 반복으로 연산을 수행하게 된다. 제안하는 방법은 스칼라 레코딩을 이용하여 보다 효율적으로 부채널 공격에 대응하는 방법이다. 따라서 경량화를 요구하는 사물인터넷 환경에 적합하며 SPA와 DPA 모두에 안전한 대응방법이다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 타원곡선 암호와 부채널 공격, 그리고 부채널 공격에 대한 대응에 대해 설명한다. Ⅲ장에서 제안하는 방법에 대해 구체적으로 기술하고, Ⅳ장에서 실험결과를 제시한다.

Ⅱ. 기존 연구

2.1 타원곡선 암호

2.1.1 타원곡선 암호

타원곡선 군은 유한체 $GF(p)$ 상에서의 Weierstrass 방정식 식(1)을 만족하는 타원곡선 위의 점들과 무한원점 O를 원소로 가지는 집합이다.

$$E: y^2 = x^3 + ax + b,\ a, b \in GF(p). \quad (1)$$

이 때, $4a^3 + 27b^2 \neq 0$이다.

타원곡선 군은 타원곡선 덧셈 연산에 대해 아벨군을 이룬다. 여기서 무한원점 O는 타원곡선 군의 항등원이고, $P=(x,y)\in E$의 덧셈에 대한 역원은 $-P=(x,-y)$로 정의된다. 그리고 타원곡선 덧셈 연산(Elliptic Curve Addition : ECADD)과 타원곡선 두 배 연산(Elliptic Curve Doubling : ECDBL)은 다음과 같이 정의된다[1].

○ ECADD : $P_1 \neq \pm P_2$인 임의의 두 점 $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$에 대해 두 점의 합 $P_3=P_1+P_2=(x_3,y_3)$은 다음과 같이 계산된다.

$$\begin{aligned} x_3 &= \lambda^2 - (x_1 + x_2), \\ y_3 &= \lambda(x_1 - x_3) - y. \end{aligned} \quad (2)$$

이 때, $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$이다.

○ ECDBL : 임의의 점 $P_1=(x_1,y_1)$의 두 배 $P_3=2P_1=(x_3,y_3)$ 은 다음과 같이 계산된다.

$$\begin{aligned} x_3 &= \lambda^2 - 2x_1, \\ y_3 &= \lambda(x_1 - x_3) - y_1. \end{aligned} \quad (3)$$

이 때, $\lambda = \dfrac{3x_1^2 + a}{2y_1}$이다.

위와 같은 연산식을 이용할 경우 연산 과정 중에 유한체 곱셈에 대한 역원을 계산해야 한다. 역원 계산은 곱셈에 비해 큰 연산량이 필요하기 때문에 최소화 하는 것이 효율적이다. 이에 따라 역원 연산이 필요하지 않은 가장 효율적인 Jacobian coordinates를 사용하는 것이 일반적이다. 다음을 만족하는 $(X:Y:Z)$을 Jacobian point이라고 한다.

$$(X:Y:Z) = \{(\lambda^2 X, \lambda^3 Y, \lambda Z) : \lambda \in K^*\}. \quad (4)$$

이 때, $(x,y)=(X/Z^2, Y/Z^3)$, $Z\neq 0$이다. 그리고 $Z=0$인 $(X:Y:Z)$는 무한원점 O와 같다. Jacobian coordinates를 사용하여 나타낸 Weierstrass 방정식은 식(5)와 같다.

$$E': Y^2 = X^3 + aXZ^4 + bZ^6. \quad (5)$$

Jacobian coordinates에서 임의의 점 $P=(X,Y,Z)$에 대한 역원은 $-P=(X,-Y,Z)$이고 무한원점 O는 $(1:1:0)$이다. Jacobian coordinates에서의 ECADD 및 ECDBL 연산은 아래와 같이 계산된다.

○ ECADD : $P_1 \neq \pm P_2$인 임의의 두 점 $P_1=(X_1,Y_1,Z_1)$, $P_2=(X_2,Y_2,Z_2)$에 대해 두 점의 합 $P_3=P_1+P_2=(X_3,Y_3,Z_3)$은 다음과 같이 계산된다.

$$\begin{aligned} X_3 &= (3X_1^2 + aZ_1^4)^2 - 8X_1Y_1^2, \\ Y_3 &= (3X_1^2 + aZ_1^4)(4X_1Y_1^2 - X_3) - 8Y_1^4, \\ Z_3 &= 2Y_1Z_1. \end{aligned} \quad (6)$$

○ ECDBL : 임의의 점 $P_1=(X_1,Y_1,Z_1)$의 두 배 $P_3=2P_1=(X_3,Y_3,Z_3)$ 은 다음과 같이 계산된다.

$$\begin{aligned} X_3 &= (Y_2Z_1^3 - Y_1)^2 - (X_2Z_1^2 - X_1)^2(X_1 + X_2Z_1^2), \\ Y_3 &= (Y_2Z_1^3 - Y_1)(X_1(X_2Z_1^2 - Z_1)^2 - X_3) \\ &\quad - Y_1(X_2Z_1^2 - X_1)^3, \\ Z_3 &= (X_2Z_1^2 - X_1)Z_1. \end{aligned} \quad (7)$$

2.1.2 타원곡선 스칼라 곱셈

타원곡선 암호시스템의 가장 중요한 연산은 타원곡선 스칼라 곱셈이다. 타원곡선 스칼라 곱셈은 k가 양의 정수이고 P가 타원곡선 상의 한 점일 때, P를 k번 더하는 연산을 말한다.

$$Q = kP = P + P + \cdots + P \quad (k \text{ times}) \quad (8)$$

타원곡선 스칼라 곱셈은 타원곡선 암호시스템에서 키 쌍 생성 및 전자서명 생성 등에 사용되며 타원곡

선 암호시스템의 안전성 및 효율성에 큰 영향을 주는 연산이다. 타원곡선 스칼라 곱셈의 안전성은 Q와 P를 알 때, k를 찾기 어려움을 이용하는 타원곡선 이산대수 문제에 기반 한다.

Algorithm 1은 타원곡선 전자서명(ECDSA : Elliptic Curve Digital Signature Algorithm)을 위한 키 쌍을 생성하는 알고리즘이다. step 2의 타원곡선 스칼라 곱셈 연산이 주 연산이며 비밀키 d

Algorithm 1. Key pair generation

INPUT : Domain parameters
$D=(q, FR, S, a, b, P, n, h)$.
OUTPUT : Public key Q, private key d.
1. Select $d \in [1, n-1]$.
2. Compute $Q=dP$.
3. Return (Q, d).

Algorithm 2. ECDSA signature generation

INPUT : Domain parameters
$D=(q, FR, S, a, b, P, n, h)$,
private key d, message m.
OUTPUT : Signature (r, s).
1. Select $k \in [1, n-1]$.
2. Compute $kP=(x_1, y_1)$ and convert x_1 to an integer $\overline{x_1}$.
3. Compute $r=\overline{x_1} \bmod n$. If $r=0$ then go to step 1.
4. Compute $e=H(m)$.
5. Compute $s=k^{-1}(e+dr) \bmod n$. If $s=0$ then go to step 1.
6. Return (r, s).

Algorithm 3. Left-to-right binary method

INPUT : Positive integer
$k=(k_{l-1}, \cdots, k_0), P \in E(F_q)$.
OUTPUT : kP.
1. $Q \leftarrow O$.
2. For i from $l-1$ to 0 do
2.1 $Q \leftarrow 2Q$.
2.2 if $(k_i=1)$ then $Q \leftarrow Q+P$.
3. Return (Q).

를 이용한 연산이기 때문에 연산 중 비밀키 d의 노출 위험이 있다. Algorithm 2는 ECDSA 의 서명 생성 알고리즘으로 step 2에서 난수 k에 대한 타원곡선 스칼라 곱셈 연산을 수행한다. 이 때, step 5의 r과 s는 서명 값이고 e는 메시지의 해시 값이며 n은 타원곡선 군의 위수로 모두 공개된 값이다. 따라서 step 2에서 k 값이 노출되면 비밀키 d의 값은 식(9)와 같이 쉽게 계산된다.

$$d=r^{-1}(ks-e) \bmod n \tag{9}$$

만약, 비밀키 d가 노출되면 서명의 위변조가 가능하다. 따라서 타원곡선 스칼라 곱셈 연산은 스칼라 값이 노출되지 않도록 안전해야한다. 공격자가 비밀키 d을 알기 위해서는 ECDLP 문제를 해결해야한다. 하지만 부채널 분석 방법이 소개되면서 우회적인 방법으로 비밀키 d을 알아내는 것이 가능해졌다. 따라서 부채널 분석에 대한 대응책들이 연구되고 있다.

타원곡선 스칼라 곱셈 연산은 일반적으로 Left-to-right binary 방법을 사용한다. 이 방법은 스칼라를 이진표현으로 나타내고 Algorithm 3과 같이 계산한다. 이러한 Left-to-right binary 방법은 비밀 값의 비트 값과 수행 연산 간에 밀접한 관계가 있기 때문에 전력분석공격에 안전하지 않다. 따라서 전력분석공격을 막기 위한 별도의 대응 기법들이 혼합 적용되어야 한다.

2.2 전력분석공격

전력분석공격은 암호장비에 물리적인 공격을 가하지 않고 연산 시 사용되는 전력량만을 이용하는 공격 방법이다. 전력분석공격은 연산을 수행하는 동안의 전력소비량과 해당 연산에 대한 데이터 사이에 밀접한 관계가 있다는 사실을 기반으로 한다. 스칼라 곱셈 알고리즘에 대한 전력분석공격은 크게 SPA와 DPA가 있다.

2.2.1 SPA(Simple Power Analysis)

SPA는 단순히 하나의 전력소비파형으로 연산 패턴을 관찰하여 비밀 값의 정보를 알아내는 공격이다. 비밀 값과 수행하는 연산 사이에 밀접한 관계가 있는 경우 연산 패턴의 구분만으로 비밀 값의 정보를 얻을 수 있다. 그리고 암호시스템을 구성하는 각 연산들은

수행되는 동안 소비하는 전력량이 다르기 때문에 전력소비파형을 통해 쉽게 구분이 가능하다. SPA는 이를 이용하여 하나의 전력소비파형으로부터 연산 패턴을 분석하는 공격 방법이다.

타원곡선 스칼라 곱셈 연산의 기반이 되는 ECADD 연산과 ECDBL 연산은 서로 다른 연산 패턴과 연산량을 가지고 있기 때문에 전력소비량의 차이를 통하여 연산의 구분이 가능하다. 만약 타원곡선 스칼라 곱셈이 비밀 값에 의존하여 수행되는 경우 SPA를 이용하여 쉽게 공격이 가능하다. 예를 들어, Algorithm 3은 k_i 값이 1인 경우에만 ECADD 연산을 수행하여 연산의 구분을 통해 쉽게 비트정보를 알 수 있기 때문에 SPA에 안전하지 않다.

기존에 연구된 대표적인 SPA 대응방법은 비밀 값의 비트 값에 관계없이 항상 일정한 연산 패턴으로 스칼라 곱셈을 수행하는 방법이다. 이 방법을 적용하면 수행 연산이 비밀 값에 의존하지 않기 때문에 전력소비파형을 통해 연산 패턴이 관찰되더라도 비밀 값의 정보가 노출되지 않는다. 대표적으로 Montgomery ladder 방법과 Double-and-add always 방법 등이 있고, 각각 Algorithm 4, Algorithm 5와 같다[13][14][15].

Algorithm 4. Montgomery ladder

INPUT : Positive integer
$k=(k_{n-1}, k_{n-2}, ...k_0)_2, P \in E(F_q)$.
OUTPUT : kP
1. $R_0 \leftarrow O$, $R_1 \leftarrow P$
2. for i from $n-1$ downto 0 do
  2.1 $b \leftarrow k_j$.
  2.2 $R_{1-b} \leftarrow R_0 + R_1$; $R_b \leftarrow 2R_b$
3. Return (R_0)

Algorithm 5. Double-and-add always

INPUT : Positive integer
$k=(k_{n-1}, k_{n-2}, ...k_0)_2, P \in E(F_q)$.
OUTPUT : kP
1. $R_0 \leftarrow O$
2. for i from $n-1$ downto 0 do
  2.1 $R_0 \leftarrow 2R_0$; $R_1 \leftarrow R_0 + P$.
  2.2 $b \leftarrow k_j$; $R_0 \leftarrow R_b$
3. Return (R_0)

Algorithm 4는 step 2.2에서 k_i 값과 관계없이 항상 ECADD와 ECDBL 연산을 수행하여 SPA에 안전하다. 마찬가지로 Algorithm 5도 step 2.1에서 항상 동일하게 ECADD와 ECDBL 연산을 수행하기 때문에 SPA에 안전하다. 하지만 위 방법들은 매 루프마다 ECADD 연산을 수행하기 때문에 Left-to-right binary 방법에 비해서 추가적인 연산이 많이 필요하다는 단점이 있다.

2.2.2 DPA(Differential Power Analysis)

DPA는 다수의 전력소비파형을 이용한 통계적인 분석을 통해 비밀 값과 관계가 있는 정보를 추출하는 공격이다. 전력소비파형은 고정된 비밀 값을 여러 개의 데이터와 연산하여 얻으며 각 데이터에 대한 특정 연산에서의 중간 계산 값을 추측하여 분석을 수행한다. 중간 계산 값을 추측한 후에는 해당 연산에 대한 전력소비량의 통계치를 계산하여 추측이 옳은지 판단한다. DPA는 이에 따라 비밀 값에 대한 정보를 얻는 공격방법이다. DPA는 통계적인 분석법을 적용하여 육안으로 분석하기 어려운 파형에 대해서도 공격이 가능하므로 SPA보다 강력한 공격이라고 할 수 있다.

DPA에 대한 대응방법으로는 내부적으로 비밀 값 또는 데이터 등을 랜덤화하여 중간 계산 값에 대한 추측이 불가능하도록 하는 방법이 있다. 첫 번째 방법은 비밀 값인 스칼라를 랜덤화하는 방법이다[5]. 비밀 값의 랜덤화는 스칼라 k를 랜덤한 수 r과 타원곡선 군의 위수 n을 이용하여 식 (10)과 같이 나타낸다.

$$k' = k + r \cdot n \tag{10}$$

이 때, $k'P$는 식 (11)과 같이 kP와 같은 결과 값을 갖는다.

$$k'P = kP + r \cdot n \cdot P = kP + O = kP \tag{11}$$

이 방법을 적용하면 같은 스칼라 k로 여러 번 연산을 수행하더라도 랜덤 값 r에 의해 변형된 형태의 스칼라를 사용하게 된다. 따라서 공격자는 연산의 중간 계산 값을 추측할 수가 없어 DPA 공격이 불가능하다. 두 번째 방법은 kP 연산에서 점 P을 랜덤화하는 방법이다[5]. 비밀의 랜덤한 점 R을 선택하여

$k(R+P)-S$ 연산을 수행한다. 이 때, $S=kR$이다. $k(R+P)-S$는 식 (12)과 같이 kP와 같은 결과 값을 갖는다.

$$k(R+P)-S=S+kP-S=kP \tag{12}$$

R에 따라 다른 점에 대한 k배 연산을 수행하기 때문에 중간 계산 값은 랜덤하게 나타난다. 따라서 공격자는 중간 계산 값을 추측할 수 없어 DPA 공격이 불가능하다. 세 번째 방법은 가장 일반적으로 사용하는 방법으로 좌표 값을 랜덤화하는 Random projective coordinates 방법이다[5]. $GF(p)$ 상에서 연산 상의 효율이 가장 좋은 Jacobian coordinates는 랜덤한 $r(r\neq 0)$ 을 사용하여 식 (13)에 따라 랜덤화할 수 있다.

$$\begin{aligned}(r^2X:\ r^3Y:\ rZ) &= \left(\frac{r^2X}{(rZ)^2}, \frac{r^3Y}{(rZ)^3}\right)\\ &= \left(\frac{X}{Z^2}, \frac{Y}{Z^3}\right)=(X:\ Y:\ Z)\end{aligned} \tag{13}$$

위와 같이 랜덤화하면 r 값에 따라 연산의 중간 계산 값이 랜덤하게 나타난다. 따라서 r을 모르는 공격자는 중간 계산 값을 추측할 수 없기 때문에 중간 값 추측을 이용하는 DPA 공격이 불가능하다. Random projective coordinates는 랜덤화를 위한 연산 외에는 추가적인 연산이 들지 않아 효율적으로 DPA에 대응하는 방법이다.

2.3 스칼라 레코딩 방법

스칼라 레코딩은 스칼라 k을 이진수 표현이 아닌 음수 표현, 2^b진수 표현 등의 표현 방식을 적용하여 레코딩하는 것이다. 기존의 스칼라 레코딩은 타원곡선 스칼라 곱셈 연산의 효율 향상을 목적으로 연구되었다. 연산의 효율 향상을 위한 대표적인 스칼라 레코딩 방법으로는 Non-adjacent form(NAF)이 있다[3]. NAF 방법은 Algorithm 6과 같고 이를 이용한 타원곡선 스칼라 곱셈은 Algorithm 7과 같다.

NAF 방법은 음수 표현을 사용하여 k의 비트열의 Hamming weight를 줄이는 방법이다. Hamming weight는 약 $\frac{l}{2}$에서 NAF 방법 수행 후 약 $\frac{t}{3}$로 감소한다. 이 때, $l\leq t\leq l+1$이다. 타원곡선 스칼라 곱셈 연산은 Algorithm 7과 같이 k_i가 1인 경우 P를 더하고 k_i가 -1인 경우에는 $-P$을 더하여 수행된다. 타원곡선 암호는 점의 역원을 구하는 연산이 매우 간단하다는 장점이 있어 음수 digit에 대한 연산은 어렵지 않게 수행할 수 있다.

Algorithm 6. Computing the NAF

INPUT : Positive integer
$k=(k_{l-1}, k_{l-2}, ..., k_0)_2$.
OUTPUT : $NAF(k)=(k'_{t-1}, k'_{t-2}, ..., k'_0)$
1. $i\leftarrow 0$
2. while $k\geq 1$ do
   2.1 if k is odd then: $k'_i\leftarrow 2-(k \bmod 4)$, $k\leftarrow k-k'_i$.
   2.2 Else: $k'_i\leftarrow 0$.
   2.3 $k\leftarrow k/2, i\leftarrow i+1$.
3. Return $(k'_{t-1}, ..., k'_1, k'_0)$

Algorithm 7. Binary NAF method for point multiplication

INPUT : Positive integer
$k=(k_{n-1}, k_{n-2}, ..., k_0)_2$, $P\in E(F_q)$.
OUTPUT : kP.
1. Use Algorithm 6 to compute $NAF(k)=\sum_{i=0}^{l-1} k_i 2^i$.
2. $Q\leftarrow O$.
3. For i from $l-1$ downto 0 do
   3.1 $Q\leftarrow 2Q$.
   3.2 If $k_i=1$ then $Q\leftarrow Q+P$.
   3.3 If $k_i=-1$ then $Q\leftarrow Q-P$.
4. Return (Q).

최근에는 부채널 공격에 대응하기 위한 새로운 스칼라 레코딩 방법들이 소개되고 있다[10][11]. 스칼라 레코딩을 이용하면 기존의 대응방법들과 같이 SPA와 DPA 대응책을 혼합사용하지 않고도 모두에 안전할 수 있어 보다 효율적으로 부채널 공격에 대응할 수 있다. 제안된 방법들은 기존의 대응책들과 달리 SPA와 DPA 모두에 안전하게 하는 방법들이다. 대표적인 SPA와 DPA에 안전한 스칼라 레코딩 방법에는 Random signed-scalar recoding(Ha-Moon's method)과 Random m-ary 방법이 있다[1

0][11]. Random signed-scalar recoding은 NAF 방법에 랜덤성을 추가한 것으로 랜덤수를 생성하여 랜덤수의 비트 값에 따라 변환형태를 선택하여 레코딩하는 방법이다. 스칼라 $k=(k_{l-1},\cdots,k_0)$, 올림수 $c=(c_l,\cdots,c_0)$, 랜덤 값 $r=(r_{l-1},\cdots,r_0)$, 레코딩 결과 $d=(d_l,\cdots,d_0)$에 대해 레코딩 방법은 Table 1과 같고, 이 방법을 적용한 타원곡선 스칼라 곱셈은 Algorithm 8과 같다.

Random m-ary 방법은 랜덤수를 생성하여 윈도우 사이즈를 선택하는 방법이다. Random m-ary 방법을 적용한 타원곡선 스칼라 곱셈 방법은

Table 1. Random signed-scalar recoding

Input				Output		
k_{i+1}	k_i	c_i	r_i	c_{i+1}	d_i	Remark
0	0	0	0	0	0	NAF
0	0	0	1	0	0	NAF
0	0	1	0	0	1	NAF
0	0	1	1	1	$\bar{1}$	AF
0	1	0	0	0	1	NAF
0	1	0	1	1	$\bar{1}$	AF
0	1	1	0	1	0	NAF
0	1	1	1	1	0	NAF
1	0	0	0	0	0	NAF
1	0	0	1	0	0	NAF
1	0	1	0	1	$\bar{1}$	NAF
1	0	1	1	0	1	AF
1	1	0	0	1	$\bar{1}$	NAF
1	1	0	1	0	1	AF
1	1	1	0	1	0	NAF
1	1	1	1	1	0	NAF

Algorithm 8. SPA resistant algorithm using Random signed-scalar recoding

INPUT : $P\in E(F_q)$, $d=(d_{n-1},d_{n-2},...,d_0), d_i\in\{\bar{1},0,1\}$.
OUTPUT : dP
1. $Q[0]=O$
2. $P[0]=P, P[1]=P, P[\bar{1}]=-P$
2. for i from $n-1$ to 0 by -1 do
  2.1 $Q[0]=2Q[0]$
  2.2 $Q[1]=Q[0]+P[d_i]$
  2.3 $Q[\bar{1}]=Q[1]$
  2.4 $Q[0]=Q[d_i]$
Return ($Q[0]$)

Algorithm 9. Random m-ary method for point multiplication

INPUT : $P\in E(F_q)$, c, $array[l]$, $r=(r_{n-1},r_{n-2},...,r_0),(r_i=1,2,3)$.
OUTPUT : kP
{Precomputation phase}
1. $P[1]=P$
2. for i from 2 to 6 by 2 do
  2.1 $P[i]=2P[i/2]$
  2.2 $P[i+1]=P[i-1]+P[2]$
{Evaluation phase}
3. $Q[0]=O$
4. for i from 0 to $l-1$ do
  4.1 $Q[1]=2Q[0]$
  4.2 $Q[2]=2Q[1]$
  4.3 $Q[3]=2Q[2]$
  4.4 $Q[1+r_i]=Q[r_i]+P[array[i]]$
  4.5 $Q[0]=Q[c_i+r_i]$
Return ($Q[0]$)

Algorithm 9와 같다.

두 방법은 모두 랜덤수에 의존하여 레코딩된 스칼라를 사용하기 때문에 랜덤한 중간 계산 값을 가진다. 따라서 중간 값 추측이 불가능해 DPA에 안전하다. 또한 연산 과정에서 모든 digit에 대해 동일한 루프의 연산을 고정적으로 수행하기 때문에 SPA에 안전하다.

III. 제안하는 방법

본 논문에서는 SPA와 DPA 모두에 안전한 새로운 스칼라 레코딩 방법인 Carry Random Recoding(CRR) 방법을 제안한다.

3.1 제안하는 레코딩 방법

CRR 방법은 올림수 표현을 이용하여 digit을 변환하는 방식이다. 예를 들어 $01_{(2)}$은 $1\bar{1}\bar{1}_{(2)}$와 같이 올림수 표현을 사용한 digit으로 변환이 가능하다. CRR 방법은 이러한 올림수를 이용한 변환을 기본으로 하며 변환여부를 랜덤하게 선택하여 수행한다. 구체적인 변환은 Table 2와 같다.

이 때, 올림수 변환은 랜덤 값을 생성하여 랜덤 값의 비트 값에 따라 각 digit에 선택적으로 수행하며 레코딩 방법은 Algorithm 10과 같다.

Table 2. Carry Random Recoding

digit	Carry	Recoded digit
00	1	$0\bar{4}$
01	0	01
	1	$0\bar{3}$
10	0	02
	1	$0\bar{2}$
11	0	03
	1	$0\bar{1}$

CRR 방법은 SPA에 대응하기 위해 모든 digit에서 같은 패턴의 연산이 수행되도록 레코딩한다. 즉, $00_{(2)}$의 경우에는 항상 $0\bar{4}$로 레코딩한 후 올림수를 1로 하여 레코딩함으로써 k_i'이 00인 경우가 발생하지 않도록 한다.

레코딩 결과는 생성한 랜덤 값에 의존하여 변환한다. 게다가 앞 digit에서의 올림수 발생 여부가 다음 digit에 영향을 준다. 올림수 발생이 연쇄적으로 영향을 미쳤을 때 한 digit은 최대 2까지의 올림수를 받게 된다. 한 digit은 최대 일곱 가지의 경우로 레코딩 될 수 있어 아주 랜덤한 레코딩 결과가 생성된다. 또한 레코딩 결과는 digit의 길이가 증가하더라도 한 자리를 넘어가지 않는다. 다음 정리 1은 레코딩 결과의 digit의 길이가 증가하더라도 한 자리만을 증가함을 보여줌으로써 Algorithm 10의 step 4가 정당함을 보여준다.

Algorithm 10. Carry Random Recoding (CRR) method

INPUT : Positive integer
$k=(k_{l-1},...,k_1,k_0)_4,\ k_i\in\{0,1,2,3\}$.

OUTPUT : Recoded scalar
$k'=(k'_{t-1},...,k'_1,k'_0)_4,$
$k'_i\in\{-4,-3,-2,-1,1,2,3\}$

1. $i\leftarrow 0$
2. For i from 0 to $l-1$ do
   2.1 Judge the hit bit R of random number
   2.2 If $k \bmod 4=0$, then $R\leftarrow 1$.
   2.3 $k'_i\leftarrow(k \bmod 4)-4\cdot R$.
   2.4 $k\leftarrow k/4+R$.
3. $k'_l\leftarrow k$.
4. If $k'_l=0$, then $t=l$ else $t=l+1$.
5. Return $(k'_{t-1},k'_{t-2},...,k'_1,k'_0)_4$

정리 1.

l은 스칼라의 digit 수, t는 레코딩 후 스칼라의 digit 수라 하고, $k_i(i=0,...,l-1)$는 스칼라 k의 i번째 digit이라고 하자. CRR 방법을 이용하여 스칼라 k를 레코딩 한 결과 k'은 자리 수가 증가하더라도 최대 한 자리까지밖에 증가하지 않는다.

증명)

수학적 귀납법을 이용하여 쉽게 증명할 수 있다.

$l=1$일 때, 스칼라는 $00\le k\le 11$이다. 올림수 변환을 하면 $10\bar{4}\le k'\le 10\bar{1}$ 로 t가 $l+1$로 1 증가함을 확인할 수 있다.

이제 $l=n$일 때, 레코딩 결과 값인 $k'_{n-1}4^n+k'_{n-2}4^{n-1}+\cdots+k'_0$의 길이가 $n+1$이라고 가정한다. 그러면 $l=n+1$일 때 $n+1$번째 digit k_n는 n 자리까지 변환하여 발생한 올림수 c가 더해진다. 이 때, c는 $0\le c<4$ 이므로 k_n는 $00\le k_n\le 110$ 을 만족하게 된다. 이는 올림수 변환을 하면 $10\bar{4}\le k'_n\le 20\bar{2}$ 이므로 $t=n+2$로 길이가 최대 1 증가함을 보인다. 따라서 CRR 방법을 적용했을 때 digit의 길이는 증가하지 않거나 증가하더라도 1 증가한다.

3.2 제안한 레코딩을 적용한 타원곡선 스칼라 곱셈

본 절에서는 CRR 방법을 적용한 타원곡선 스칼라 곱셈 연산에 대해 기술하고 추가적으로 음수 digit에 대한 ECADD 연산의 SPA 대응책을 기술한다.

제안하는 레코딩이 적용된 타원곡선 스칼라 곱셈 알고리즘은 Algorithm 11과 같다.

- step 1 : Algorithm 10을 이용하여 스칼라 k에 대한 CRR 결과 값 k'을 생성한다.
- step 2 : 점 P와 양수 digit 값들의 타원곡선 스칼라 곱셈 결과인 $P, 2P, 3P, 4P$을 사전 연산하여 테이블에 저장한다. CRR 방법은 자체적으로 DPA에 대응하기 때문에 DPA 대응책인 Random projective coordinates를 사용할 필요가 없다. 따라서 사전연산 테이블은 Jacobian coordinates를 사용하지 않아도 된다. 사전연산 테이블은 affine coordinates로 저장하여 Jacobian-Jacobian coordinates

Algorithm 11. CRR for point multiplication

INPUT : Positive integer $k, P \in E(F_q)$.

OUTPUT : kP.

1. Use Algorithm 10 to compute $k' = \sum_{i=0}^{t-1} k_i, 4^i$.
2. Compute $P_i = iP$ for $i \in 1, 2, 3, 4$.
3. $Q \leftarrow O$.
4. For i from $t-1$ to 0 do
   4.1 $Q \leftarrow 4Q$.
   4.2 $Q \leftarrow Q + P_{k'_i}$.
5. Return (Q).

ECADD보다 연산 효율이 좋은 Jacobian-affine coordinates ECADD 방법을 사용함으로써 다른 대응책들을 사용하는 것에 비해 연산 속도를 향상시킬 수 있다[3].

최근 음수 digit에 대한 ECADD 연산 수행 시 $-Y$ 좌표를 계산하는 추가연산이 발생함에 따라 부채널 정보를 통해 digit의 부호 정보가 노출됨이 연구되었다[8].

따라서 제안하는 방법은 사전연산 시 부호가 음인 digit에 대한 연산까지 고려하여 $(x, y, -y)$와 같이 세 개의 좌표를 저장하고 음수 digit에 대한 부채널 공격에 안전하도록 하는 ECADD 연산을 사용한다. 제안하는 ECADD 연산은 step 4에서 설명한다.

- step 3 : 점 Q를 무한원점 O로 설정한다.
- step 4 : k'의 상위 digit에서부터 각 digit에 해당하는 타원곡선 네 배 연산(ECQPL : Elliptic Curve Quadrupling)과 ECADD 연산을 수행한다. 이 때, ECADD 연산은 $P+Q$ 연산과 $P-Q$ 연산에 대한 SPA에 안전하기 위해 변형한 알고리즘을 사용한다. 제안하는 ECADD 연산은 Algorithm 12와 같다.

변형한 알고리즘은 $P+Q$ 또는 $P-Q$ 연산을 수행할 때, 두 점 P와 Q, Q의 부호 정보 s을 입력받아 Algorithm 12에 따라 연산한다. 제안하는 알고리즘은 입력으로 받은 부호 값에 따라 연산에 y 또는 $-y$을 적용하도록 step 6과 같이 연산한다. 이에 따라 Algorithm 12는 digit의 부호에 관계없이 같은 과정의 연산을 수행한다.

Algorithm 12. Signed point addition using affine-Jacobian coordinates on Weierstrass curves.

INPUT : $P, Q \in E(F_q)$ such that $P = (X_1, Y_1, Z_1)$ is in Jacobian coordinates, $Q = (x_2, y_{2,0}, y_{2,1}), (y_{2,1} = -y_{2,0})$ is in affine coordinates, $s = 1(negative)$ or $0(positive)$ is sign bit of operand Q.

OUTPUT : $P+Q = (X_3, Y_3, Z_3) \in E(F_q)$ in Jacobian coordinates.

1. if $P = O$ then return Q
2. if $Q = O$ then return P
3. $t_1 = Z_1^2$
4. $t_2 = t_1 \times Z_1$
5. $t_1 = t_1 \times x_2$
6. $t_2 = y_{2,s} \times t_2$
7. $t_1 = t_1 - X_1$
8. $t_2 = t_2 - Y_1$
9. if $t_1 = 0$ then
10. If $t_2 = 0$ then
11. Use point doubling algorithm
12. else Return (O)
13. $Z_3 = Z_1 \times t_1$
14. $t_3 = t_1^2$
15. $t_4 = t_3 \times t_1$
16. $t_3 = t_3 \times X_1$
17. $t_1 = 2t_3$
18. $X_3 = t_2^2$
19. $X_3 = X_3 - t_1$
20. $X_3 = X_3 - t_4$
21. $t_3 = t_3 - X_3$
22. $t_3 = t_3 \times t_2$
23. $t_4 = t_4 \times Y_1$
24. $Y_3 = t_3 - t_4$
25. Return $P+Q = (X_3, Y_3, Z_3)$

- step 5: kP 연산의 결과 점 Q를 반환한다.

3.3 비교분석

본 절에서는 CRR 방법 적용에 따른 연산량과 CRR 방법의 기반이 되는 NAF 방법 적용에 따른 연산량을 비교분석하고, 기존에 연구된 SPA와 DPA 모두에 안전한 대응방법들과 비교한다.

CRR 방법과 NAF 방법은 모두 변환 시의 길이 증가가 최대 1이라는 점과 음수 표현을 이용한 변환이라는 점에서 일치한다. 하지만 NAF 방법은 부채널 공격에 취약한 반면에 CRR 방법은 SPA 및 DPA에 안전하다. NAF 방법과 CRR 방법을 사용하여 변환 한 스칼라를 이진표현 했을 때, 그 길이가 n라고 하자. ECADD 연산량에 영향을 미치는 Hamming weight는 Table 3과 같다. 그리고 NAF 방법의 경우 P에 대한 ECADD 연산만 수행하므로 사전연산이 불필요한 반면에 CRR 방법의 경우 $P, 2P, 3P, 4P$에 대한 사전연산이 필요하다. 연산량에 대한 비교는 Table 4와 같다.

CRR 방법을 적용하게 되면 NAF 방법을 적용하는 것에 비해 14%의 ECADD 연산을 더 수행하게 된다. 즉, 256비트의 스칼라의 경우 CRR 방법을 적용하면 전체 연산에 있어서 NAF 방법을 적용했을 때 보다 약 44번의 ECADD 연산과 두 번의 ECDBL 연산이 더 발생한다. 하지만 NAF 방법에 SPA와 DPA에 대한 대응책을 추가적용 하여 고려하면 CRR 방법이 보다 효율적인 방법이라고 할 수 있다. 또한 SPA와 DPA에 대응하는 이전 연구 결과들과 비교해 보아도 CRR 방법이 연산 효율성이 매우 우수한 대응방법이라는 것을 알 수 있다. 비교한 기존 연구들은 모두 CRR 방법과 같이 스칼라 레코딩을 이용한 방법들이며 SPA와 DPA 모두에 대응하는 방법들이다. 비교분석한 연산량은 Table 5와 같으며 ECDBL의 연산량은 ECADD의 0.7배로 계산하였다[20].

Table 5에서 볼 수 있듯이 SPA와 DPA에 안전한 타원곡선 스칼라 곱셈 연산은 CRR 방법을 적용한 것이 가장 적은 연산량을 가진다. 제안하는 방법은 256비트 스칼라에 대한 타원곡선 스칼라 곱셈 연산에 있어서 Random m-ary 방법보다 약 11% 정도의 연산량 감소를 가져온다.

Table 3. Hamming weight comparison with NAF.

NAF	CRR
$\frac{n}{3}$	$\frac{n}{2}$

Table 4. Computational cost comparison with NAF.

7	NAF	CRR
Pre-computation	0	$[1]ECADD$ $+[2]ECDBL$
Computation	$\left[\frac{n}{3}\right]ECADD$ $+[n]ECDBL$	$\left[\frac{n}{2}\right]ECADD$ $+[n]ECDBL$

Table 5. Computational cost comparison with existing countermeasure against SPA and DPA

Algorithm	ECADD	ECDBL	Total
Ha-Moon's method [10]	$n+1$	$n+1$	$1.7n+1.7$
Random m-ary method [11] (2, 4, 8-ary)	$0.43n+3$	$1.29n+3$	$1.333n+5.1$
CRR	$0.5n+1$	$n+2$	$1.2n+2.4$

IV. 실험 결과

본 장에서는 NAF 방법과 CRR 방법에 대한 이론적 분석을 실험을 통해 검증한다. NAF 방법과 CRR 방법 각각에 대한 SPA를 수행하였으며 중간 값 추측을 이용한 DPA에 대한 실험은 중간 값 추측이 불가능 하므로 생략한다. 실험은 KLA-Scarf 시스템의 ARM 부채널 테스트 보드를 이용하여 진행하였다[21]. 테스트 보드에 탑재된 ARM 칩의 사양은 아래와 같다.

- S3C2410(ARM920 T)
- 16/32-bit RISC microprocessor
- 동작주파수 200MHz

파형은 Lecroy WaveRunner 204Xi-A 2GHz 오실로스코프를 사용하여 샘플링속도 100MS/sec로 수집하였다. 실험에 사용된 스칼라 파라미터는 다음과 같으며 해당 스칼라에 대해서만 실험을 수행하였다.

$$
\begin{aligned}
k = \\
&11000011101001010101101010100101 \\
&10000000000000011000000000000001 \\
&10001000100010001000100010001000 \\
&11011101110111011101110111011101 \\
&11001100110011001100110011001100 \\
&10101010101010101010101010101010 \\
&00000000000000000000000000000000 \\
&11111111111111111111111111111111
\end{aligned} \tag{14}
$$

4.1 파형 분석

타원곡선 스칼라 곱셈에 대한 SPA는 주요 연산인 ECDBL과 ECADD 연산의 구분으로 수행된다. 따라서 ECDBL과 ECADD 연산의 전력파형 차이를 분석하였다. ECDBL과 ECADD 연산은 유한체를 기반으로 한 곱셈, 제곱, 덧셈, 뺄셈 연산으로 구성되어 있다. 이러한 기본적인 연산을 이용하여 ECDBL는 네 번의 곱셈, 네 번의 제곱, 열 번의 덧셈 및 뺄셈 연산이 필요하고, ECADD 연산을 수행하기 위해서는 열 두 번의 곱셈, 네 번의 제곱, 일곱 번의 덧셈 및 뺄셈 연산이 필요하다. 따라서 ECADD 연산이 ECDBL 연산보다 더 많은 연산을 수행하므로, 이러한 사실은 전력소모량을 통해 나타날 것이다. Fig. 1.은 수집한 파형에서 ECDBL과 ECADD를 나타내고 있다. 이론적인 분석대로 ECDBL은 ECADD보다 상대적으로 적은 연산수행이 관찰됨을 확인할 수 있었다.

두 연산이 전력파형에서 구별될 수 있음을 이용하여 NAF 방법과 CRR 방법의 SPA 안전성을 분석한다.

4.2 NAF 방법

Fig. 2.는 $NAF(k)$의 상위 세 digit $10\bar{1}$에 대한 연산에 따른 전력소비파형이다. 파형에서 보이는 첫 번째 ECDBL 연산은 두 번째 digit인 0에 해당하는 연산이며, 세 번째 digit $\bar{1}$에 대해서는 ECDBL 연산과 ECADD 연산이 수행되었다. 즉, 0인 digit에서는 ECDBL 연산만 수행하고, 0이 아닌 digit $1, \bar{1}$에서는 ECDBL, ECADD 연산이 수행된다(digit 1도 $\bar{1}$과 마찬가지로 ECDBL, ECADD 연산이 수행되므로 digit $\bar{1}$에 대해서만 언급하기로 한다.). 따라서 digit이 0은 digit이 1에 비해 ECADD 연산이 수행되지 않으므로 수집된 파형에서 ECADD 연산 부분을 찾아 0인 스칼라 값을 알아낼 수 있다.

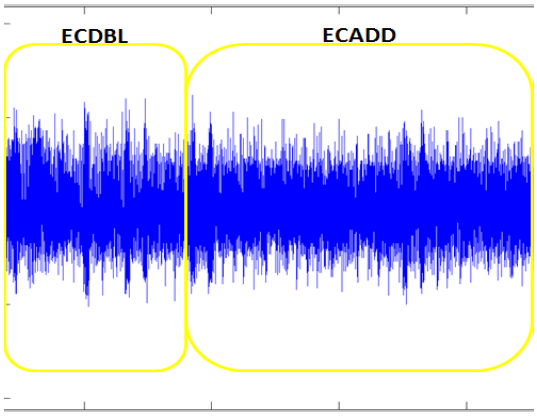

Fig. 1. Power trace of ECDBL and ECADD

1과 $\bar{1}$을 구별하기 위하여 1인 digit에서는 $P+Q$ 연산이 수행되고, $\bar{1}$인 digit에서는 $P-Q$ 연산이 수행됨을 이용한다. $P-Q$ 연산 수행 시에는 Q을 통해 $-Q$을 계산하는 추가적인 연산이 발생하므로 그에 다른 전력소비가 파형에 나타날 것이다. Fig. 3.은 $P+Q$ 연산과 $P-Q$ 연산의 파형을 나타낸 것이다. 추가연산에 따른 파형의 차이가 확인되었고, 1, $\bar{1}$인 스칼라 값을 모두 알아낼 수 있었다.

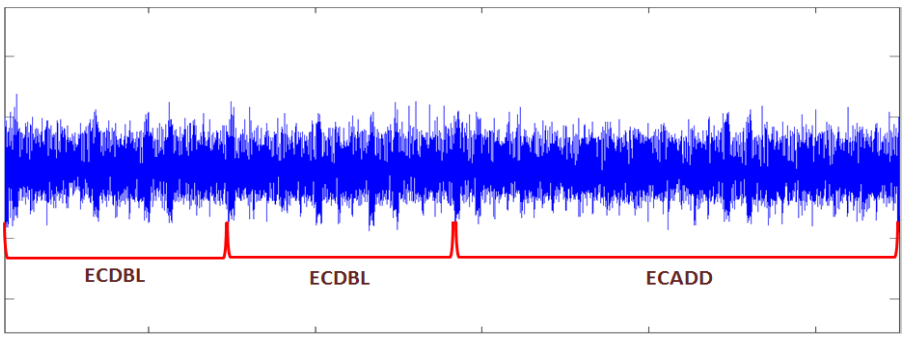

Fig. 2. Power consumption trace of three MSD(Most significant digit) of scalar k recoded by using NAF.

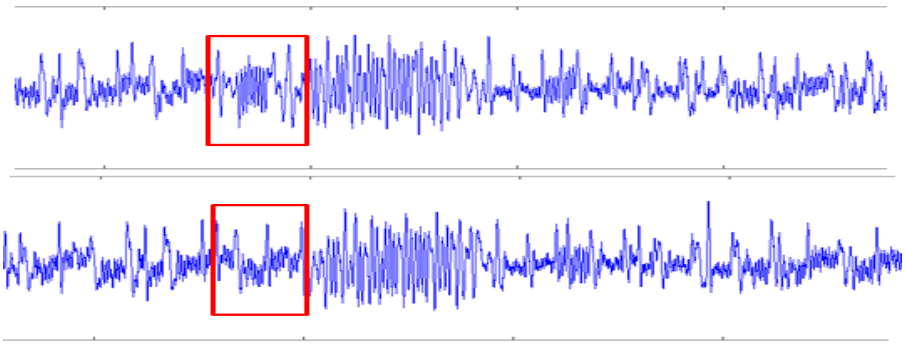

Fig. 3. Above : SPA attack during computation of negative digit.
bottom : SPA attack during computation of positive digit.

4.3 CRR 방법

제안하는 CRR 방법은 스칼라를 0인 digit이 없도록 레코딩하므로 모든 digit에서 ECADD 연산이 수행된다. 또한 4진수 표현을 사용하므로 각 digit에 대해 두 번의 ECDBL 연산이 수행된다. 따라서 모든 digit에 대하여 일정하게 ECDBL, ECDBL, ECADD 연산이 수행되므로 전력파형에서 규칙적인

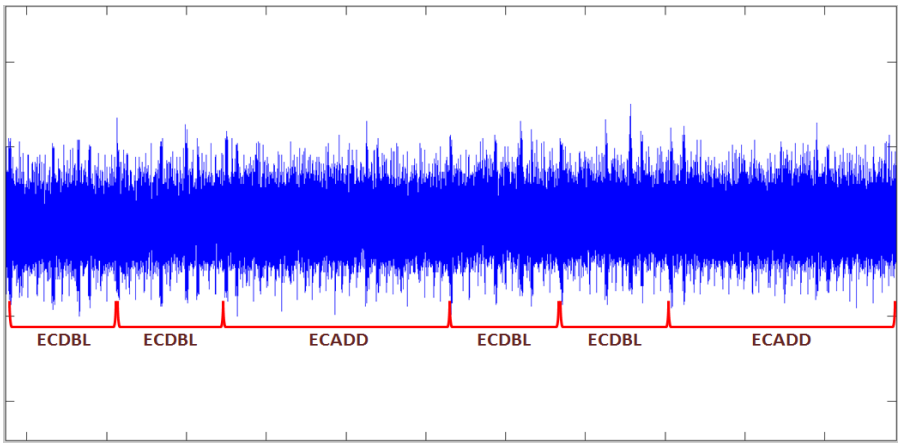

Fig. 4. Power consumption trace of two digits of scalar k recoded by using CRR.

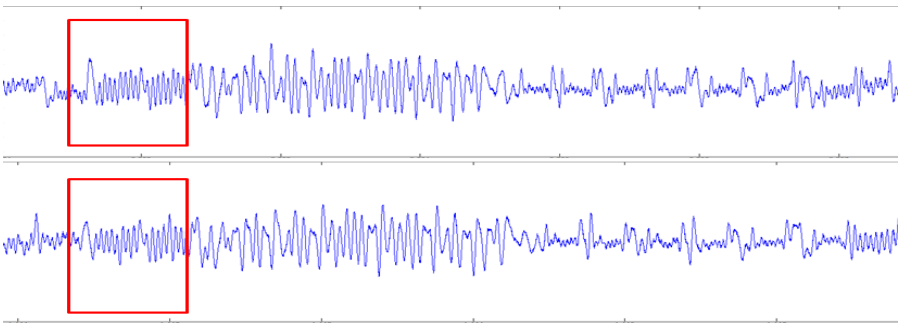

Fig. 5. Above : Application of countermeasure for computation of positive digit.
bottom : Application of countermeasure for computation of negative digit.

패턴이 나타날 것이다. Fig. 4.는 2개의 digit에 대한 타원곡선 연산의 파형이다. 위에서 설명한 것과 같이 ECDBL, ECDBL, ECADD 연산의 파형이 일정하게 반복됨을 확인할 수 있었다. 따라서 NAF 방법과 달리 0인 digit를 구별할 수 없다.

제안하는 ECADD 알고리즘은 $P+Q$ 연산과 $P-Q$ 연산을 사전연산 테이블의 값을 이용하여 동일하게 수행하므로 $P-Q$ 연산에 있어서 $-Q$을 계산하는 추가연산이 필요하지 않다. 즉, digit 1과 digit $\bar{1}$에 대한 연산이 동일하다. Fig. 5.는 Fig. 3.과는 달리 digit $\bar{1}$에 대한 연산의 파형에서 추가연산이 나타나지 않으며 digit 1과 digit $\bar{1}$의 연산 차이가 없음을 보여준다. 따라서 SPA를 통해 digit의 값이 1인지 $\bar{1}$인지 구별할 수 없다.

제안하는 CRR 방법 및 ECADD 방법은 NAF 방법과 달리 digit 값에 대한 연산의 차이를 발생시키지 않으므로 이를 이용한 SPA의 취약점을 보완할 수 있다.

V. 결 론

본 논문에서는 SPA와 DPA에 안전한 스칼라 레코딩 기법을 제안하였다. 제안하는 방법은 랜덤수를 사용하여 스칼라를 랜덤하게 레코딩하며 동시에 모든 digit에 대해 항상 같은 연산이 수행되도록 레코딩하는 방법이다. 또한 ECADD 연산에서 digit의 부호정보가 노출되지 않도록 사전연산 테이블과 ECADD 연산의 변형을 이용하여 $P+Q$ 연산과 $P-Q$ 연산을 동일하게 하였다. 사전연산 테이블은 각 점에 대해 $(x, y, -y)$와 같이 세 개의 좌표를 저장하고, ECADD 연산의 알고리즘은 $P+Q$ 연산에는 사전연산의 y 값을 사용하고, $P-Q$ 연산에는 사전연산의 $-y$ 값을 사용하도록 변형하였다. 이론적으로 분석한 바와 같이 제안하는 CRR 방법을 적용한 타원곡선 스칼라 곱셈 연산이 SPA에 안전함을 실험을 통해 확인하였다. 실험 결과 NAF 방법을 적용한 스칼라 곱셈 연산에서는 $0, 1, -1$에 해당하는 모든 digit이 분석됐지만 CRR 방법을 적용한 스칼라 곱셈 연산에서는 양수 digit과 음수 digit의 ECADD 연산이 구분되지 않았으며 모든 digit에 대해 ECDBL, ECDBL, ECADD 패턴의 연산이 동일하게 수행됨을 확인했다.

256비트 스칼라에 대한 연산을 기준으로 했을 때, 제안하는 방법은 SPA 및 DPA에 안전한 기존 연구에 비해 약 11% 적은 연산량을 가진다. 따라서 제안한 레코딩 기법을 사용하여 보다 효율적이고 안전하게 타원곡선 스칼라 곱셈 연산을 수행 할 수 있다.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Others Systems." *CRYPTO'96, LNCS 1109*, pp. 104-113, Aug. 1996.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *CRYPTO'99, LNCS 1666*, pp. 388-397, 1999.

[3] D. Hankerson, A. Menezes, and S. Vanstone, "Guide to elliptic curve cryptography", *Springer Professional Computing. Springer-Verlag,* New York, 2004.

[4] M. Joye, M. Tunstall, "Exponent recoding and regular exponentiation algorithms." *Africacrypt 2003, ed. by M. Joye. LNCS, vol.5580,* pp. 334-349, Springer. 2009.

[5] J.-S. Coron, "Resistance Against Differential Power Analysis for Elliptic Curve Cryptography.", *Advances in Cryptology - CHES'99*, volume 1717 of LNCS, pages 292-302. Springer-Verlag, 1999.

[6] Liping Wang, Weike Wang, Rong, Zhang, and Xiang Wang, "A New ECC scalar Multiplication Algorithm with Randomized Power Consumption", ICSICT2014, Guilin, China, 2014.

[7] E. Guerrini, L. Imbert, and T. Winterhalter, "Randomizing scalar multiplication using exact covering systems of congruences." Cryptology ePrint Archive, Report 2015/475, 2015.

[8] Peng LUO, Dengguo FENG, Yongbin Zhou, "An New Anti-SPA Algorithm of NAF scalar Multiplication used in ECC",

International Journal of Advancements in Computing Technology, Nov2012, Vol.4 Issue 20, p692, Nov. 2012.

〔9〕 JaeCheol Ha, SangJae Moon, "Randomized signed-scalar multiplication of ECC to resist power attacks.", in *Pre-Proceedings of Workshop on Cryptographic Hardware and Embedded Systems CHES'02*, Springer-Verlag, pp.553-565, Springer. 2002.

〔10〕 MahnKi Ahn, JaeCheol Ha, HoonJae Lee, SangJae Moon, "A Random M-ary Mehtod-Based Countermeasure against Power Analysis Attacks on ECC", *Journal of the Korean Institute of Information Security and Cryptology* v.13 no.3, pp.35 - 43, 2003

〔11〕 Feng, M., Zhu, B.B., Xu, M., Li, S. "Efficient comb elliptic curve multiplication methods resistant to power analysis." Cryptology ePrint Archive, Report 2005/222, 2005.

〔12〕 Jean-Sebastien Coron. "Resistance against differential power analysis for elliptic curve cryptosystems." In C.K. Koc and C.Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES '99), volume 1717 of Lecture Notes in Computer Science*, pages 292-302. Springer-Verlag, 1999.

〔13〕 Peter L. Montgomery. "Speeding the Pollard and elliptic curve methods of factorization." *Mathematics of Computation*, 48(177):243-264, January 1987.

〔14〕 Marc Joye and Sung-Ming Yen. "The Montgomery powering ladder." In B.S. Kaliski Jr., C.K. Koc, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES2002*, Lecture Notes in Computer Science. Springer-Verlag, To appear.

〔15〕 V. Dimitrov, L. Imbert, and P.K. Mishra, "Efficient and Secure Elliptic Curve Point Multiplication using Double Base Chain. In: Roy, B. (ed.)," ASIACRYPT 2005, LNCS 3788, pp. 59-79, 2005.

〔16〕 T. Izu and T. Takagi, "A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks," PKC 2002, LNCS 2274, pp. 280-296, 2002.

〔17〕 R. L. Rivest, A. Shamir, and L.M. Adelman, "A method for obtaining digital signatures and public key cryptosystems", *Communications of the ACM*, 21, pp.120126, 1978.

〔18〕 N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, vol. 48, pp.203~209, 1987.

〔19〕 Josyula R. Rao and Pankaj Rohatgi. "EMpowering Side-Channel Attacks", Available at http://eprint.iacr.org/complete/

〔20〕 C. H. Lim and P. J. Lee, "More flexible exponentiation with precomputation", *CRYPTO'94, LNCS2200*, pp.324~334, Springer-Verlag, 1994.

〔21〕 Y. J. Choi, D. H. Cho, J. C. Ryou, "Implementing Side Channel Analysis Evaluation Boards of KLA-SCARF system." *Journal of The Korea Institude of Information Security & Cryptology*, Vol.24, no. 1 pp.229-240. 2014

〔22〕 Yoo-Jin Baek, 'Scalar recoding and regular 2w-ary right-to-left EC scalar multiplication algorithm', Information Processing Letters 113, pp. 357-360, 2013.

〈저자소개〉

유 효 명 (Hyo Myoung Ryu) 학생회원
2014년 7월: 광운대학교 수학과 학사
2014년 8월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 부채널 공격, 공개키 암호 알고리즘, 암호구현

조 성 민 (Sung Min Cho) 학생회원
2008년 2월: 광운대학교 수학과 학사
2011년 8월: 고려대학교 정보경영공학전문대학원 석사 졸업
2011년 8월~현재: 고려대학교 정보보호대학원 정보보호학과 박사과정
〈관심분야〉 부채널 공격, 공개키 암호 알고리즘, 암호구현

김 태 원 (Taewon Kim) 학생회원
2010년 2월: 광운대학교 수학과 학사
2012년 8월: 고려대학교 정보보호대학원 석사 졸업
2012년 8월~2016년 2월: 고려대학교 정보보호대학원 박사수료
2016년 3월~현재 :(주)SNTWORKS 책임연구원
〈관심분야〉 부채널 공격, 스마트 카드 보안, 암호시스템 안전성 분석 및 고속구현

김 창 한 (Chang Han Kim) 종신회원
1985년 2월: 고려대학교 수학과 이학사
1987년 2월: 고려대학교 수학과 이학석사
1992년 2월: 고려대학교 수학과 이학박사
1992년 2월~현재: 세명대학교 정보통신학부 교수
〈관심분야〉 정수론, 공개키 암호, 암호프로토콜

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식