https://doi.org/10.5392/JKCA.2020.20.01.356

효율적인 키-난수화를 사용한 차분 전력 분석 공격에 대응하는 타원곡선 위의 스칼라 곱셈 방법

A Method for Scalar Multiplication on Elliptic Curves against Differential Power Analysis using Efficient Key-Randomization

정석원
목포대학교 정보보호학과

Seok Won Jung(jsw@mokpo.ac.kr)

요약

사물인터넷 시대가 되면서 다양한 디바이스가 유·무선으로 연결되고 있다. 이에 따른 일상생활의 편리성 향상과 함께 사생활 침해, 정보유출, 서비스 거부 등의 보안 문제가 증가하고 있다. 공개키 암호 시스템의 하나인 타원곡선 암호 시스템 ECC는 사용하는 키의 크기가 RSA 알고리즘보다 상대적으로 작아 제약적인 환경의 디바이스에 널리 사용되고 있다. 그러나 제약적인 환경의 디바이스에 적용된 ECC의 비밀 키는 스칼라 곱셈 연산을 수행하는 과정에서 전력 분석 공격법에 의해 노출될 수 있다. 본 논문에서는 SECG 표준 타원곡선 파라미터의 스칼라 곱셈 방법에 대해 차분 전력 분석에 대응하고 연산의 효율성을 증가시키는 방법을 알아본다. 제안하는 방법은 비밀 키에 타원곡선 위수의 난수 배를 더하여 차분 전력 분석에 대응하는 Coron의 방법을 사용한다. 연산의 효율성을 증가시키기 위해 SECG 표준 파라미터의 위수 n을 상대적으로 작은 상수 c로 $n=2^l \pm c$로 표현하고, $2^lP = \mp\ cP$인 성질을 이용한다. 임의의 난수를 사용한 Coron의 키-난수화 방법은 스칼라 곱셈 수행을 $2l$번 하는데, 본 논문에서 제안하는 방법은 위수 성질을 이용하면 스칼라 곱셈 수행을 약 $(3/2)l$번 수행하게 되어 25% 정도 연산의 효율성이 향상된다.

■ 중심어 : | 타원곡선 암호 시스템 | 스칼라 곱셈 | 차분 전력 분석 | 키-난수화 |

Abstract

As a becoming era of Internet-of-Things, various devices are connected via wire or wirless networks. Although every day life is more convenient, security problems are also increasing such as privacy, information leak, denial of services. Since ECC, a kind of public key cryptosystem, has a smaller key size compared to RSA, it is widely used for environmentally constrained devices. The key of ECC in constrained devices can be exposed to power analysis attacks during scalar multiplication operation. In this paper, a key-randomization method is suggested for scalar multiplication on SECG parameters. It is against differential power analysis and has operational efficiency. In order to increase of operational efficiency, the proposed method uses the property $2^lP=\mp\ cP$ where the constant c is small compared to the order n of SECG parameters and $n=2^l \pm c$. The number of operation for the Coron's key-randomization scalar multiplication algorithm is $2l$, but the number of operation for the proposed method in this paper is $(3/2)l$. It has efficiency about 25% compared to the Coron's method using full random numbers.

■ keyword : | Elliptic Curve Cryptosystem | Scalar Multiplication | Differential Power Analysis | Key-randomization |

* 이 논문은 2016년도 목포대학교 특별 연구비를 지원받아 수행되었음.

접수일자 : 2019년 10월 01일
수정일자 : 2019년 10월 29일
심사완료일 : 2019년 10월 29일
교신저자 : 정석원, e-mail : jsw@mokpo.ac.kr

I. 서 론

최근 다양한 디바이스가 개발되고, 이들에 인터넷이 연결되면서 지능화된 서비스를 제공하는 사물인터넷 IoT(Internet of Things) 환경이 도래하였다. IoT 기기의 보급률은 2021년에 약 160억 개에 이를 것으로 전망되고 있으며, 시장 성장률도 연평균 23%에 이를 것으로 예상되고 있다[2]. IoT 시장의 성장에 따라 사생활 침해, 데이터 위변조, 정보유출, 서비스 거부 등의 보안 문제 해결이 새로운 과제로 떠오르고 있다[4]. 최근 OWASP(The Open Web Application Security Project)는 안전하지 않은 네트워크 서비스, 안전하지 않은 데이터 전송 및 저장 등 IoT의 10대 보안 취약점을 발표하였다[3].

정부인터넷 보안은 사물인터넷 보안, 디바이스 보안, 게이트웨이·네트워크 서버 보안, 서비스 보안으로 나누고, 각각에 대한 보안위협과 보안 요구사항을 설명하고 있다[5]. 2017년 한국인터넷진흥원이 발간한 “사물인터넷(IoT) 환경에서의 암호·인증기술 이용 안내서”에서는 사물인터넷 서비스 전 주기의 위협에 대응하는 기술적 방안에 따른 권고 암호 서비스를 안내하고 있다[1]. 이 안내서에서 타원곡선 알고리즘은 키 관리/분배 기법 제공, 전자 서명 기반의 부인 방지 기법, 인증서 기반의 SSL(Secure Socket Layer) 등을 이용한 안전한 채널 통신 제공 등에 사용되고 있다[1]. 타원곡선 알고리즘은 공개키 알고리즘으로 소인수 분해의 어려움에 근거한 RSA에 비해 키의 크기가 작아 실행 속도가 빠르고 전송 데이터 양이 작은 장점이 있다. 이러한 장점으로 인해 타원곡선 알고리즘은 6LoWPAN과 CoAP 프로토콜에서 사용되며, 비트코인에도 사용되고 있다[5][6].

1996년 Kocher가 제약적인 환경에 효율성만 추구하고 구현한 RSA 알고리즘에 대해 곱셈과 제곱 연산의 시간 차이를 이용하여 비밀 키를 찾는 시간 공격법을 제안하였다[19]. 그 이후 암호장치의 연산 과정에 발생하는 소비전력, 전자파, 소리 등의 부채널 정보를 이용한 공격법이 꾸준히 발표되고 있다. 1999년에 RSA 알고리즘과 ECC 알고리즘에 대해 각 알고리즘을 수행할 때의 내부 연산의 소비전력의 차이를 측정하여 비밀 키를 찾아내는 전력 분석법이 제안되었다[10][20]. 이외에도 전자기파 분석법, 차분 오류 분석법, 캐시 공격법 등이 제안되었다[12][15][18].

타원곡선 암호 알고리즘에 대한 단순 전력 분석법 SPA(Simple Power Analysis)에 대해서는 덧셈 연산과 두 배 연산에 같은 공식을 사용하는 방법, 두 배 연산과 덧셈 연산을 항상 하는 방법, 정수 스칼라를 사용하는 ZSD(zeroless signed-digit expansion) 등 여러 가지 대응 법이 제안되었다[7][10][14]. 그러나 제안된 대응 법에 대해 Fouque 등의 두 배 공격(doubling attack), Goubin의 타원곡선 상의 특이점을 이용한 공격 등 새로운 공격 방법이 다시 제안되었다[11][13].

타원곡선 암호 알고리즘에 대한 차분 전력 분석법 DPA(Differential Power Analysis)에 대해 Coron의 키-난수화 방법, Clavier 등의 키 분해(exponent splitting) 방법, Coron의 점 가리기(blinding the point) 방법 등이 제안되었다[9][10]. 그러나 제안된 대응법에 대해 Ciet 등의 SECG(Standards for Efficient Cryptography Group) 권고 타원곡선의 키-난수화 취약점, Ha 등의 2-Torsion 공격법, Fouque 등의 두 배 공격(doubling attack) 등 새로운 공격 방법이 다시 제안되었다[8][11][16].

본 논문에서는 타원곡선 알고리즘의 차분 전력 분석법에 대응하기 위해 제안된 Coron의 방법에 대해 알아본다. 또한 SECG 표준 타원곡선 파라미터에 Coron의 방법을 적용할 때 취약점이 알려져 있는데, 이를 개선하기 위해 키-난수화 방법을 사용한다. 이때 연산의 효율성이 떨어지게 되는데 타원곡선 위수의 성질을 이용하여 연산 효율성을 향상시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 타원곡선 스칼라 곱셈 방법, 단순 전력 분석법과 차분 전력 분석법에 대해서 알아본다. 3장에서는 키-난수화 방법과 타원곡선 위수를 이용한 효율성 증대 방법에 대해서 알아보고, 이에 대한 안전성과 효율성을 기존 알고리즘과 비교해 본다. 끝으로 4장에서 결론을 내린다.

II. 이론적 배경

1. 타원곡선 스칼라 곱셈 알고리즘

소수 p가 3보다 클 때, 유한체 $\mathbb{F}_p$ 위에 정의된 타원곡선은 식 (1)과 같이 정의된다.

$$E: y^2 = x^3 + ax + b \quad (1)$$

여기에서 $a, b \in \mathbb{F}_p$ 이다. 무한 원점을 O라고 표시하면 식 (2)는 덧셈에 대한 군을 이루고, 이를 타원곡선 군이라 부른다.

$$E(\mathbb{F}_p) = \{(x, y) \mid y^2 = x^3 + ax + b\} \bigcup \{O\} \quad (2)$$

타원곡선 군의 점 P와 양의 정수인 k에 대해 스칼라 곱셈 kP를 식 (3)과 같이 정의한다.

$$kP = \overbrace{P + P + \cdots + P}^{(k-1)\text{번}} \quad (3)$$

스칼라 곱셈을 효율적으로 처리하기 위해 l비트 스칼라 k를 이진법으로

$$k = 2^{l-1}k_{l-1} + 2^{l-2}k_{l-2} + \cdots + 2k_1 + k_0 \quad (4)$$

와 같이 표현해 보자. 여기에서 $i = 0, 1, \cdots, l-1$에 대해 $k_i = 0$ 또는 1의 값을 갖는다. 그러면

$$\begin{aligned} kP &= (2^{l-1}k_{l-1} + 2^{l-2}k_{l-2} + \cdots + 2k_1 + k_0)P \\ &= 2^{l-1}k_{l-1}P + 2^{l-2}k_{l-2}P + \cdots + 2k_1P + k_0P \\ &= k_{l-1}2^{l-1}P + k_{l-2}2^{l-2}P + \cdots + k_1 2P + k_0P \\ &= k_{l-1}2(2^{l-2}P) + k_{l-2}2(2^{l-3}P) + \cdots \\ &\quad + k_1(2P) + k_0P \end{aligned} \quad (5)$$

이다. 식 (5)를 맨 오른쪽에서부터 보면 점 P를 계속 2배한 값을 키 비트 k_i에 따라 선택적으로 더하는 것으로 볼 수 있다. 따라서 다음과 같은 오른쪽-왼쪽 이진 스칼라 곱셈 알고리즘 1을 구할 수 있다[17].

알고리즘 1 (오른쪽-왼쪽 이진 스칼라 곱셈)

[입력] $k = (k_{l-1}, k_{l-2}, \cdots, k_0), P \in E(\mathbb{F}_p)$

[출력] $Q = kP$

1. $Q \leftarrow O$
2. For i = 0 to $l-1$ do
   2.1 If $k_i = 1$ then $Q \leftarrow Q + P$
   2.2 $P \leftarrow 2P$
3. Return(Q)

알고리즘 1의 단계 2.1에서 두 점의 덧셈은 점 $P(x_1, y_1)$와 점 $Q(x_2, y_2)$에 대해 $P + Q = (x_3, y_3)$라 하면 식 (6)과 같이 계산된다[17].

$$\begin{aligned} x_3 &= \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, \\ y_3 &= \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \end{aligned} \quad (6)$$

알고리즘 1의 단계 2.2에서 한 점의 두 배 연산은 점 $P(x_1, y_1)$에 대해 $2P = (x_4, y_4)$라 하면 식(7)과 같이 계산된다[17].

$$\begin{aligned} x_4 &= \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1, \\ y_4 &= \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_4) - y_1. \end{aligned} \quad (7)$$

2. 단순 전력 분석법

알고리즘 1의 단계 2.1과 단계 2.2는 서로 다른 두 점의 덧셈 연산과 한 점의 두 배 연산이 사용된다. 그런데 식 (6)과 식 (7)에서 보듯이 $P+Q$와 $2P$의 계산양이 달라 이들을 계산할 때 소비되는 전력량이 다르게 된다. 따라서 연산 시 소비되는 전력을 측정하면 $P+Q$와 $2P$를 연산할 때 소비되는 전력 파형이 다르게 나타나고, 소비전력 파형을 살펴보면 덧셈 연산과 두 배 연산을 구별할 수 있게 된다. 알고리즘 1의 단계 2에서 키 비트 k_i가 1일 때에는 단계 2.1이 수행되어 덧셈과 두 배가 순차적으로 계산된다. 그러나 키 비트 k_i가 0일 때에는 단계 2.1이 수행 안 되어 두 배만 계산된다. 따라서 소비전력량을 측정하면 키 비트가 1일 때와 0일 때의 파형의 모양이 다르므로 킷값이 1인지 0인지를 알 수 있게 된다[10]. 이렇게 암호화 연산을 한 번 수행하면서 소비전력을 측정한 후 소비전력 파형으로부터 비밀 키의 값을 알아내는 공격법을 단순 전력 분석법 SPA(Simple Power Analysis)이라고 한다[20].

Coron은 논문 [10]에서 알고리즘 1에 대해 SPA 분석이 가능하게 된 원인인 조건 분기문 If문을 없애고, 항상 두 배 연산과 덧셈 연산을 하도록 가짜 연산을 추가하여 SPA에 대응하는 알고리즘 2를 제안하였다.

알고리즘 2 (항상 두 배와 덧셈을 하는 스칼라 곱셈)

[입력] $k=(k_{l-1}, k_{l-2}, \cdots, k_0), P \in E(\mathbb{F}_p)$

[출력] $Q=kP$

1. $Q[0] \leftarrow P$
2. For i = $l-1$ down to 0 do
   2.1 $Q[0] \leftarrow 2Q[0]$
   2.2 $Q[1] \leftarrow Q[0] + P$
   2.3 $Q[0] \leftarrow Q[k_i]$
3. Return($Q[0]$)

3. 차분 전력 분석법

1999년 Kocher 등은 DES 알고리즘에서 키가 입력되어 연산되는 S-Box의 소비전력을 여러 개 수집한 후, 추측 키와 분류함수(selection function)를 사용하여 소비전력을 분류하고, 이들의 전력 소비량의 차분을 이용하여 비밀 키를 찾아내는 차분 전력 분석법 DPA(Differential Power Analysis)를 제안하였다[20].

Coron은 논문 [10]에서 Kocher 등이 제안한 방법을 사용하여 알고리즘 2를 분석하였다. 알고리즘 2의 단계 2.1은 이전 For 루프의 단계 2.3의 결과 값인 $Q[0]$의 값이 두 배가 되는 것이다. 즉, 알고리즘 2의 단계 2.1은 이전 For 루프의 키 비트가 0이었으면 이전에 입력된 값 $Q[0]$이 2배 된 값 $2Q[0]$을 다시 2배하는 것으로 결과적으로 $4Q[0]$을 계산하게 된다. 그런데 키 비트가 1이었으면 이전 For 루프의 단계 2.3의 결과가 $Q[1]$의 값이므로 $4Q[0]$이 계산되지 않는다. Coron은 이 사실을 사용하여 여러 개의 타원곡선 점 입력에 대한 소비전력 파형을 수집한 후 키 비트의 추측에 따라 소비전력들을 분류하였다. 이들의 평균값 분포를 이용하여 키 비트의 추측이 올바른 지를 판단하여 비밀 키 비트를 찾아내는 방법을 제안하여 알고리즘 2가 차분 전력 분석 공격에 취약함을 보였다.

위에서 설명한 차분 전력 분석법은 비밀 키 값이 고정되어 있어 이전 루프 값과 현재 루프 값이 연관되어 계산된다. 그 결과 소비전력의 상관성이 있게 되고, 이 사실을 이용하여 비밀 키를 찾아내는 방법이다. Coron은 이러한 취약점인 For 루프 계산 과정의 연관성을 제거하기 위해 키를 난수화하는 방법을 제안하였다[10]. 정수 n을 타원곡선 위의 점의 위수라 하면

$$nP = O \tag{8}$$

가 성립함은 잘 알려진 사실이다. Coron은 식 (8)의 사실을 이용하여 매번 비밀 키의 비트 값이 달라질 수 있도록 난수 r을 선택하고 kP를 계산하는 대신에 $(k+rn)P$를 계산하는 것을 제안하였다. 난수 r이 비밀 키의 길이와 같은 l비트이면, $k+rn$은 $2l$비트가 되어 알고리즘 2의 For 루프가 $2l$번 수행된다. 이 경우 $(k+rn)P$의 계산은 kP 보다 2배 많은 연산을 수행하게 된다. Coron은 연산의 효율성을 높이기 위해 난수 r을 20비트 정도 되는 값을 택할 것을 권고하였다[10].

Ⅲ. 본 론

1. 표준 타원곡선에서의 문제점

SECG(Standards for Efficient Cryptography Group)에서는 타원곡선 알고리즘의 효율적인 구현을 위해 타원곡선 파라미터를 권고하고 있다[21]. 그런데 Ciet과 Joye는 SECG에서 권고하고 있는 secp224k1 파라미터의 경우, Coron이 제안한 키-난수화 방법은 비밀 키의 여러 비트가 난수화 되지 않고 그대로 노출되어 전력분석 공격법에 의해 비밀 키가 밝혀질 수 있음을 언급했다[8]. 즉, secp224k1에서 타원곡선의 위수 n은 헥사 값으로

$$n = 01\ 00000000\ 00000000\ 00000000\ 0001DCE8\ D2EC6184\ CAF0A971\ 769FB1F7 \tag{9}$$

이다. 식 (9)의 위수 n은 비트 값이 '0'인 부분이 많아 Coron이 제안한 키-난수화 방법으로 난수화를 하면

$$k+rn = (r)_2 \| k_{l-1} \cdots k_{l-t} \| \text{some bits} \tag{10}$$

로 표현된다[8]. 즉, 식 (10)에서 보는 바와 같이 $k+rn$의 중간 부분에 비밀 키 비트가 그대로 노출되는 문제점이 있다.

2. 모듈러 감산된 키-난수화 스칼라 곱셈 방법

Coron은 논문 [10]에서 차분 전력 분석 공격법을 막기 위해 비밀 키를 타원곡선 위수의 배수로 난수화시키

는 방법을 제안하였다. 계산의 효율성을 위해 난수의 크기를 20비트 정도로 제한하는 것을 권고하였다. 그러나 3장 1절에서 살펴보았듯이 SECG에서 권고하는 타원곡선 파라미터들 중 일부에 대해서는 키 비트가 완전히 난수화되지 않는 경우가 발생하여 차분 전력 분석 공격이 가능하게 된다.

본 논문에서는 이러한 문제점을 극복하기 위해 비밀키 k와 같은 크기의 l비트 난수 r을 사용하도록 한다. 이 경우 $k+rn$는 $2l$비트가 되므로 알고리즘 2를 사용할 때, $(k+rn)P$ 계산에 $2l$번의 For문이 실행된다. 그러므로 $(k+rn)P$의 계산이 kP의 계산보다 2배 더 많은 연산을 하게 되어 효율성이 떨어진다.

l비트 난수 r을 사용하면서도 연산의 효율성을 높이기 위해 본 논문에서는 다음과 같은 위수의 성질을 사용한다. SECG의 파라미터 중 일부의 위수 n은

$$n = 2^l + c \tag{11}$$

또는

$$n = 2^l - c \tag{12}$$

로 나타낼 수 있다. 여기에서 c는 $t(<l)$비트로 위수 n에 비해서 매우 작은 정수이다. 예를 들어 secp224k1 파라미터의 위수 n은 225비트 정수로

$$n = 2^{224} + c \tag{13}$$

이고, 여기에서

c= 1DCE8 D2EC6184 CAF0A971 769FB1F7

으로 113비트 정수이다. 그러면 식 (11)의 경우는 위수의 성질에 따라

$$O = nP = (2^l + c)P = 2^l P + cP \tag{14}$$

이고, 식 (14)를 정리하면

$$2^l P = -cP \tag{15}$$

를 얻을 수 있다. 식 (12)의 경우는

$$O = nP = (2^l - c)P = 2^l P - cP \tag{16}$$

이고, 식 (16)을 정리하면

$$2^l P = cP \tag{17}$$

를 얻을 수 있다. 식 (15)의 경우와 식 (17)의 경우는 부호가 다르다는 것 이외에는 타원곡선의 스칼라 곱셈을 계산하는 데에는 차이가 없으므로 식 (17)의 경우로 설명하겠다.

식(17)의 양변에 계속 2를 곱하면

$$\begin{aligned} 2^{l+1}P &= 2\cdot 2^l P = (2c)P, \\ 2^{l+2}P &= 2^2\cdot 2^l P = (2^2 c)P, \\ &\cdots \\ 2^{2l-1}P &= 2^{l-1}\cdot 2^l P = (2^{l-1}c)P \quad 1 \end{aligned} \tag{18}$$

를 얻을 수 있다. $k+rn$이 $2l$비트 이므로 $k+rn$의 계산 결과는 $2l$비트 정수 d로 표현된다.

$$k+rn = d = d_{2l-1}2^{2l-1} + \cdots\ d_l 2^l + \cdots + d_0 \tag{19}$$

로 놓고, 식 (18)의 결과를 이용하면,

$$\begin{aligned} (k+rn)P &= (d_{2l-1}2^{2l-1} + \cdots\ d_l 2^l + \cdots + d_0)P \\ &= d_{2l-1}2^{l-1}2^l P + \cdots + d_l 2^l P + d_{l-1}2^{l-1}P + \cdots + d_0 P \\ &= d_{2l-1}2^{l-1}cP + \cdots + d_l cP + d_{l-1}2^{l-1}P + \cdots + d_0 P \\ &= (d_{2l-1}2^{l-1}c + \cdots + d_l c + d_{l-1}2^{l-1} + \cdots + d_0)P \end{aligned} \tag{20}$$

이다. 즉, 식 (20)에서 보듯이 $(k+rn)P$을 계산하는 것은 $(d_{2l-1}2^{l-1}c + \cdots + d_l c + d_{l-1}2^{l-1} + \cdots + d_0)P$ 를 계산하는 것과 같다. 그런데

$$\begin{aligned} &d_{2l-1}2^{l-1}c + \cdots + d_l c + d_{l-1}2^{l-1} + \cdots + d_0 = \\ &\quad (d_{2l-1}2^{l-1} + \cdots + d_l)c + (d_{l-1}2^{l-1} + \cdots + d_0) \end{aligned} \tag{21}$$

이다. 식 (21)의 오른쪽은 l비트 정수와 t비트 정수 c를 곱한 값에 l비트 정수를 더한 것이다. 따라서 식 (21)의 결과는 $l+t$비트 정수가 된다. 그러므로 알고리즘 2를 사용할 때, 식 (21)을 이용하면 $(k+rn)P$의 계산에 For 문이 $2l$번 수행에서 $l+t$번 수행만으로 처리될 수 있다. 예를 들어 secp224r1 파라미터의 경우,

$$\begin{aligned} n &= 2^{224} - 2^{112} + 16A2\ E0B8F03E \\ &\qquad 13DD2945\ 5C5C2A3D \\ &= 2^{224} - (2^{112} - 16A2\ E0B8F03E \\ &\qquad 13DD2945\ 5C5C2A3D) \end{aligned} \tag{22}$$

로 나타낼 수 있다. 식 (22)에서 보듯이 위수 n의 비트 수는 224비트이고 상수 c의 비트 수는 112비트이다. 식 (21)을 이용하면 c가 112비트이므로 $(k+rn)P$의 계산을 위해 For 문을 $2\cdot 224$번 수행하는 대신에 $224+112$번 수행하면 된다. 이 경우 효율성은 25% 정도 증진된다. SECG에서 권고하고 있는 위수 n에 대해 상수 c를 계산하면 [표 1]과 같다.

표 1. SECG 권고 타원곡선 위수 n과 상수 c

SECG 파라미터 명 [21]	위수 n [21]	상수 c
secp192k1	FFFFFFFF FFFFFFFF FFFFFFFE 26F2FC17 0F69466A 74DEFD8D	2^{97} - 26F2FC17 0F69466A 74DEFD8D
secp192r1	FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831	2^{96} - 99DEF836 146BC9B1 B4D22831
secp224k1	01 00000000 00000000 00000000 0001DCE8 D2EC6184 CAF0A971 769FB1F7	1DCE8 D2EC6184 CAF0A971 769FB1F7
secp224r1	FFFFFFFF FFFFFFFF FFFFFFFF FFFF16A2 E0B8F03E 13DD2945 5C5C2A3D	2^{112} - 16A2 E0B8F03E 13DD2945 5C5C2A3D
secp256k1	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141	2^{129} - BAAEDCE6 AF48A03B BFD25E8C D0364141
secp256r1	FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551	2^{128} - BCE6FAAD A7179E84 F3B9CAC2 FC632551
secp384r1	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF C7634D81 F4372DDF 581A0DB2 48B0A77A ECEC196A CCC52973	2^{192} - C7634D81 F4372DDF 581A0DB2 48B0A77A ECEC196A CCC52973
secp512r1	01FF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFA 51868783 BF2F966B 7FCC0148 F709A5D0 3BB5C9B8 899C47AE BB6FB71E 91386409	2^{257} - 51868783 BF2F966B 7FCC0148 F709A5D0 3BB5C9B8 899C47AE BB6FB71E 91386409

3. 안전성과 효율성 분석

3장 2절에서는 타원곡선 스칼라 곱셈에 키-난수화를 사용하여 단순 전력 분석과 차분 전력 분석에 대응하도록 하고, 연산의 효율성을 위해 타원곡선 위수 n의 성질을 이용하였다.

[표 2]는 기존 알고리즘과 제안 방법에 대해 단순 전력 분석 공격과 차분 전력 분석 공격의 가능성 유무와 스칼라 곱셈 수행에 필요한 연산 횟수를 요약하였다. 알고리즘 1은 연산의 효율성만 강조한 것으로 단순 전력 분석 공격과 차분 전력 분석 공격이 가능하다. 알고리즘 2는 If 분기문을 없애 단순 전력 분석 공격을 막지만 차분 전력 분석 공격은 막지 못한다. Coron의 임의 난수를 사용한 키-난수화 방법은 알고리즘 2의 단점을 보완하여 차분 전력 분석 공격까지 막는 방법이다. 그러나 효율성 증대를 위해 20비트 난수를 사용하는 경우, SECG 파라미터와 같이 위수의 값에 0의 열 또는 1의 열이 많은 경우에는 차분 전력 분석 공격에 취약하다. [표 2]에서 보는 바와 같이 본 논문에서 제안하는 방법은 키 비트를 모두 난수화시켜 SECG 파라미터에 대해서도 차분 전력 분석 공격을 막고 있다. 임의 난수를 사용한 Coron의 키-난수화 방법은 연산 횟수가 키의 길이가 l비트 일 때, 덧셈 $2l$번과 두 배 $2l$번의 연산을 수행하지만, 제안 방법은 위수의 상수 c가 t비트 일 때, 덧셈 $l+t$번과 두 배 $l+t$번의 연산만을 수행한다.

표 2. 알고리즘 안전성 비교

알고리즘	단순 전력 분석 공격	차분 전력 분석 공격	연산횟수
알고리즘 1	O	O	덧셈: 키의 1의 개수 두 배: l번
알고리즘 2	×	O	덧셈: l번 두 배: l번
Coron의 키-난수화 (임의 난수)	×	×	덧셈: $2l$번 두 배: $2l$번
Coron의 키-난수화 (20비트 난수)	×	△	덧셈: $l+20$번 두 배: $l+20$번
제안 방법	×	×	덧셈: $l+t$번 두 배: $l+t$번

(O: 가능, ×: 불가능, △: 일부 가능)

[표 3]은 SECG에서 권고하고 있는 타원곡선 파라미터에 대하여, 임의 난수를 사용한 Coron의 키-난수화 방법과 본 논문에서 제안하는 방법을 알고리즘 2로 수행할 때의 For문 횟수를 비교한 것이다. SECG의 파라미터에 대해 제안하는 방법이 Coron의 방법 대비 연산 횟수가 약 75% 정도를 차지하고 있다. 즉, 제안하는 방법이 Coron의 방법보다 전체 계산이 약 25% 정도 효율적이다.

표 3. SECG 파라미터에 대한 효율성 비교

SECG 파라미터 명 [21]	위수 n 비트 크기	상수 c 비트 크기	For 문 횟수		
			임의 난수 Coron의 키 난수화 방법 (A)	제안 방법 (B)	연산 효율 (B/A× 100%)
secp192k1	192	97	384	289	75.26
secp192r1	192	96	384	288	75.00
secp224k1	225	113	450	338	75.11
secp224r1	224	112	448	336	75.00
secp256k1	256	129	512	385	75.19
secp256r1	256	128	512	384	75.00
secp384r1	384	192	768	576	75.00
secp512r1	521	257	1,042	778	74.66

Ⅳ. 결 론

최근 사물인터넷 환경의 발전에 따라 다양한 디바이스에 타원곡선 알고리즘이 탑재되고 있다. 그러나 사물인터넷 환경의 디바이스에 타원곡선 스칼라 곱셈을 적절하게 구현하지 않으면 전력 분석법, 전자파 분석법 등의 부채널 공격법에 취약한 상황에 놓일 수 있다.

본 논문에서는 타원곡선 알고리즘의 스칼라 곱셈에 대한 구조와 전력 분석 공격법에 대해 알아보았다. 2장에서는 Coron이 제안한 스칼라 곱셈 알고리즘을 전력 분석 공격법에 따라 분석해 보았다. 3장에서 Coron이 제안한 키-난수화 방법에 타원곡선 군의 위수 특징을 이용하여 난수화 된 키의 비트 크기를 줄여 연산의 효율성을 얻는 방법을 살펴보았다.

본 논문에서 제안하는 타원곡선 상의 모듈러 감산을 이용한 키-난수화 스칼라 곱셈 방법은 단순 전력 분석과 차분 전력 분석에 강인하다. 또한 SECG 파라미터에 대해 본 논문에서 제안하는 방법이 임의의 난수를 사용한 Coron의 방법보다 약 25% 정도 효율적이다.

본 논문에서 제안하는 스칼라 곱셈 방법은 타원곡선을 이용한 전자서명, 키 교환 등을 구현하는 핵심요소로 사물인터넷의 디바이스 인증, 핀테크 분야의 전자서명, 디지털 콘텐츠 보호를 위한 키 교환 등에 활용될 수 있다. 향후, 연산의 효율성을 좀 더 줄이는 방법에 대한 연구와 Coron의 방법 이외의 다른 전력 분석 대응법과의 연산 효율성 비교에 대한 연구가 필요하다.

참 고 문 헌

[1] 미래창조과학부, 한국인터넷진흥원, *사물인터넷(IoT) 환경에서의 암호·인증기술 이용 안내서*, p.14, 2017.

[2] 송근혜, 이승민, "4차 산업혁명과 보안 패러다임 변화," 주간기술동향 1847호, 정보통신기술진흥센터, pp.16-27, 2018.

[3] 안철수연구소, "IoT 시스템관리, 가장 큰 문제와 취약점은 무엇일까?," 월간 안 3월호, pp.26-27, 2019.

[4] 최동진, "5G 시대의 차세대 IoT 보안," 주간기술동향 1914호, pp.2-16, 2019.

[5] 행정안전부, 한국정보화진흥원, *정부사물인터넷 도입 가이드라인*, pp.19-22, 2019.

[6] Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly, pp.65-81, 2014.

[7] E. Brier and M. Joye, "Weirstrass elliptic curves and side-channel attacks," PKC 2002, LNCS 2274, pp.335-345, 2002.

[8] M. Ciet and M. Joye, "(Virtually) Free Randomization Techniques for Elliptic Curve Cryptography," ICICS 2003, LNCS 2836, pp.348-359, 2003.

[9] C. Clavier and M. Joye, "Universal exponentiation algorithm," CHES 2001, LNCS 2162, pp.300-308, 2001.

[10] J. S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES'99, LNCS 1717, pp.292-302, 1999.

[11] P. A. Fouque and F. Valette, "The doubling attack why upwards is better than downwards," CHES 2003, LNCS 2779, pp.269-280, 2003.

[12] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," CHES 2001, LNCS 2162, pp.251-261, 2001.

[13] L. Goubin, "A refined power-analysis attack on elliptic curve cryptosystem," PKC 2003, LNCS 2567, pp.199-211, 2002.

[14] R. R. Goundar, M. Joye, A. Miyaji, M. Rivain, and A. Venelli, "Scalar multiplication on Weierstass elliptic curves from Co-Z arithmetic," J. of Crytographic Engineering, Vol.1, No.2, pp.161-176, 2011.

[15] D. Gullasch, E. Bangerter, and S. Krenn, "Cache Games - Bringing Access Based Cache Attacks on AES to Practice," IEEE Symposium on Security and Privacy, pp.490-505, 2011.

[16] J. Ha, J. Park, S. Moon, and S. Yen, "Provably secure countermeasure resistant to several types of power attack for ECC," WISA 2007, LNCS 4867, pp.333-344, 2007.

[17] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, pp.75-97, 2004.

[18] M. Joye, A. K. Lenstra, and J. J. Quisquater, "Chinese remaindering cryptosystems in the

presence of faults," J. Cryptol., Vol.12, No.4, pp.241-245, 1999.
[19] P. Kocher, "Timming Attacks on implementations of Diffie-Hellman, RSA, DSS and Other Systems," CRYPTO'96, LNCS 1109, pp.104-113, 1996.
[20] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," CRYPTO'99, LNCS 1666, pp.388-397, 1999.
[21] D. R. L. Brown, *SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0*, 2010.

저 자 소 개

정 석 원(Seok Won Jung) 정회원

- 1991년 : 고려대학교 이과대학 수학과(이학사)
- 1993년 : 고려대학교 일반대학원 수학과(이학석사)
- 1997년 : 고려대학교 일반대학원 수학과(이학박사)
- 2004년 ~ 현재 : 목포대학교 정보보호학과 교수

〈관심분야〉 : 암호 알고리즘 분석 및 구현, 암호 프로토콜 설계, IoT 디바이스 보안