타원곡선 스칼라 곱셈에 대한 비밀키 blinding을 적용한 hardware fault cryptanalysis 대응방법

여일연*, 이경근*, 김환구*, 문상재*

*경북대학교 전자공학과, 이동네트워크 정보보호기술 연구센터

A countermeasure using secret-key blinding for hardware fault cryptanalysis on elliptic curve scalar multiplication

Il-yeon Yeo*, Kyung-keun Lee*, Hwan-koo Kim*, Sang-jae Moon*

*School of Electronic and Electrical Eng., Kyungpook National University

* Mobile Network Security Research Center

요약

본 논문에서는 타원곡선 스칼라 곱셈에 대하여 새로운 형태의 hardware fault cryptanalysis를 적용해 보고, 이에 대한 대응방법으로서 비밀키 blinding방법을 제안하고 있다. 또한 비밀키 blinding 방법을 사용함으로써 늘어나는 연산량을 기존의 대응 방법과 비교하고, 이러한 비밀키 blinding방법이 사용될 수 있는 범위에 대해 다루고 있다.

I. 서론

최근 실험을 통한 side-channel이라는 추가적인 정보의 경로를 이용한 새로운 형태의 공격법이 소개되었다. 이러한 공격을 side-channel공격 혹은 물리적 공격이라 일컫는다[1], [2].

side-channel 공격은 크게 세 가지 형태의 공격 방법이 있다. 하나는 1996년 Bellcore사에서 발표한 논문에서 처음 소개된 것으로서 암호 연산이 수행되는 기기에 오류를 주입하거나 자연적으로 발생한 오류가 암호문에 포함될 때 공격의 대상인 비밀키를 알아내는 하드웨어 오류 공격(hardware fault cryptanalysis) 방법이다[3]. 다른 공격방법으로는 1996년 P. Kocher에 의해 제안된 시차공격(timing attack)방법과 1998년 P. Kocher에 의해 제안된 전력분석(power analysis) 공격이 있다[4], [5]. 시차공격 방법은 암호 연산의 수행시간을 정보로 하여 공격하는 방법이고, 전력분석 공격 방법은 암호 기기의 연산 수행시 소모되는 전력을 분석하여 공격하는 방법이다.

본 논문에서는 세 가지 side-channel 공격중 하드웨어 오류 공격에 대해 다루고자 한다. 다양한 오류 공격의 기법에 따라 그에 대한 대응방법 또한 여러 가지가 있지만, 논문에서 다루고 있는 타원곡선 스칼라 곱셈에 대한 공격의 대응방안으로 제시된 방법은 크게 두 가지로 나뉘어 질 수 있다. 하나는 결과값을 검증하는 방법이 있고, 다른 하나는 정보를 숨기는 방법이 있다. 기존의 정보를 숨기는 방법은 서명될 메시지에 대한 blinding이었다. 본 논문에서는 이러한 방법에 비해 제안하는 비밀키 blinding 방법이 좀 더 효율적으로 하드웨어 오류 공격을 막을 수 있음을 보이고, 다른 형태의 side-channel 공격에 대한 안전성에 대해

서도 살펴보기로 하겠다.

II. 타원곡선 스칼라 곱셈에 대한 HFC

1. 타원곡선 상의 연산

타원곡선은 Affine 좌표계, Projective 좌표계 그리고 Jacobian 좌표계 등의 여러 가지 좌표계에서 표현 될 수 있다. 이 절에서는 p > 3인 소수에 대해서 기저체 F_p위에서 이러한 여러 좌표계에서 정의된 타원곡선 상의 점에 대한 덧셈 공식과 그에 필요한 유한체 연산수를 살펴보겠다.

1) Affine 좌표계

Weierstrass 등식 $y^2=x^3+ax+b$를 만족시키는 점 $P=(x_1, y_1)$, $Q=(x_2, y_2)$에 대해 $P+Q=(x_3, y_3)$를 계산하는 연산식은 다음과 같다.

◆ Affine 좌표계에서의 덧셈 연산 $(P\neq\pm Q)$

$t_A(P+Q)$

$x_3=\lambda^2-x_1-x_2$ S

$y_3=\lambda(x_1-x_3)-y_1$ 2M

$\lambda=\frac{y_2-y_1}{x_2-x_1}$ I

◆ Affine 좌표계에서의 두배 연산 $(P=Q)$

$t_A(2P)$

$x_3=\lambda^2-2x_1$ S

$y_3=\lambda(x_1-x_3)-y_1$ 2M

$\lambda=\frac{3x_1^2+a}{2y_1}$ I+S

연산수 $t_A(P+Q)=I+2M+S$

$t_A(2P)=I+2M+2S$

2) Projective 좌표계

Weierstrass 등식 $Y^2Z=X^3+aXZ^2+bZ^3$ $(x=X/Z,\ y=Y/Z)$를 만족시키는 점 $P=(X_1, Y_1, Z_1)$, $Q=(X_2, Y_2, Z_2)$에 대해 $P+Q=(X_3, Y_3, Z_3)$를 계산하는 연산식은 다음과 같다.

◆ Projective 좌표계에서의 덧셈 연산(P≠±Q)

$t_P(P+Q)$

$X_3=\lambda_6^2-\lambda_7\lambda_3^2$ M+2S

$Y_3=(\lambda_9\lambda_6-\lambda_8\lambda_3^3)\ /\ 2$ 3M

$Z_3=Z_1Z_2\lambda_3$ 2M

$\lambda_1=X_1Z_2^2$ M+S

$\lambda_2=X_2Z_1^2$ M+S

$\lambda_3=\lambda_1-\lambda_2$

$\lambda_4=Y_1Z_2^3$ 2M

$\lambda_5=Y_2Z_1^3$ 2M

$\lambda_6=\lambda_4-\lambda_5$

$\lambda_7=\lambda_1+\lambda_2$

$\lambda_8=\lambda_4+\lambda_5$

$\lambda_9=\lambda_7\lambda_3^2-2X_3$

◆ Projective 좌표계에서의 두배연산 $(P=Q)$

$t_P(2P)$

$X_3=\lambda_1^2-2\lambda_2$ S

$Y_3=\lambda_1(\lambda_2-X_3)-\lambda_3$ 2M

$Z_3=2Y_1Z_1$ M

$\lambda_1=3X_1^2+aZ_1^4$ 3S

$\lambda_2=4X_1Y_1^2$ M+S

$\lambda_3=8Y_1^4$ S

연산수 : $t_P(P+Q)=12M+4S$

$t_P(2P)=4M+6S$

3) Jacobian 좌표계

Weierstrass 등식 $Y^2=X^3+aXZ^4+BZ^6$ $(x=X/Z^2,\ y=Y/Z^3)$를 만족시키는 점 $P=(X_1, Y_1, Z_1)$, $Q=(X_2, Y_2, Z_2)$에 대해 $P+Q=(X_3, Y_3, Z_3)$를 계산하는 연산식은 다음과 같다.

◆ Jacobian 좌표계에서의 덧셈 연산(P≠±Q)

$t_J(P+Q)$

$X_3=-H^3-2U_1H^2+r^2$ 2M+2S

$Y_3=-S_1H^3+r(U_1H^2-X_3)$ 3M

$Z_3=Z_1Z_2H$ 2M

$U_1=X_1Z_2^2$ M+S

$U_2=X_2Z_1^2$ M+S

$S_1=Y_1Z_2^3$ 2M

$S_2=Y_2Z_1^3$ 2M

$H=U_2-U_1$

$r=S_2-S_1$

◆ Jacobian 좌표계에서의 두배 연산 $(P=Q)$

$t_f(2P)$

$X_3 = T$

$Y_3 = -8Y_1^4 + K(S-T)$ 2M+2S

$Z_3 = 2Y_1Z_1$ M

$S = 4X_1Y_1^2$ M

$K = 3X_1^2 + aZ_1^4$ 3S

$T = -2S + K^2$ S

연산수 : $t_f(P+Q) = 13M+4S$

$t_f(2P) = 4M+6S$

4) 스칼라 곱셈(scalar multiplication)

한점 P를 k번 더하는 연산을 한점에 대한 스칼라 곱셈(scalar multiplication)이라 하고 kP로 표기한다.

kP의 계산은 정수 k의 이진 표현법을 이용한 double-and-add 방법을 통하여 바로 얻을 수 있다. k를 이진 표현법을 이용하여 나타내면 $k=(k_{l-1}, k_{l-2}, \ldots, k_0)$로 쓸 수 있다. 여기서 k_{l-1}이 k의 최상위 비트이다.

Algorithm (Double-and-Add)

---

input P
Q ← P
for i from $l-2$ to 0 do
Q ← 2Q
if k_i = 1 then Q ← Q + P
output Q

그림1: 스칼라 곱셈 알고리즘

2. hardware fault cryptanalysis

본 장에서는 기존의 타원곡선 암호시스템에 대한 오류 공격 기법에 대해 간단히 살펴본 후, 본 논문에서 적용한 오류 공격 기법에 대해 살펴본다.

1) 기존의 공격법

◆ KMOV 시스템에 대한 오류 공격

Joye에 소개된 공격에 의하면 KMOV라 불리는 새로운 공개키 암호 시스템에 대한 오류공격을 제안하고 있다[6]. 이 공격법은 메시지가 서명되는 순간 발생하는 transient 오류를 사용하고 있다. KMOV 시스템에서는 메시지가 타원곡선상의 한 점으로서 표현된다[7]. 서명될 메시지를 $P = (x_1, y_1)$라고 하고 올바른 서명 값은 $Q = dP$, 잘못된 서명 값은 $Q' = d'P$라고 하자. 여기서 d'는 d의 j번째 비트에 오류를 발생시켜 "0"을 "1"로 "1"을 "0"으로 바꾼 것이다. 즉 $d' = d \pm 2^j$이다. 이렇게 오류가 발생한 비트를 알기 위해 공격자는 $eQ' - P$, 즉 $(d' - d)eP$를 계산하면 된다.

$$eQ' - P = \begin{cases} 2^j(x_2, y_2) & \text{, if } d_j = 0 \\ 2^j(x_2, -y_2) & \text{, if } d_j = 1 \end{cases}$$

여기서 $(x_2, y_2) = eP$이다.

또한 위 식에서 $P = (x_1, y_1)$은 타원곡선 위의 점이므로 P의 역원을 $-P = (x_1, -y_1)$으로 표시하였다.

◆ 차분 오류 공격(differential fault attacks)

Biehl 외 2명에 의해 소개된 공격법으로서 식

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

로 주어지는 Weierstrass 등식에서 기저체 GF(p)의 특성수(characteristic) p가 2이든지 2이상의 소수이든지, 타원곡선상의 점의 연산에서는 계수 a_6가 사용되지 않는다는 점을 이용하여 타원곡선의 스칼라 곱셈연산을 오류 공격하였다[8]. 이 공격법은 서명될 메시지 P가 저장된 레지스터에 발생된 오류를 이용하고 있다.

이 논문에서는 시스템 파라미터로 주어지는 암호학적으로 강한 타원곡선 E위의 점 P를 입력으로 하는 대신 계수 a_6만 다른 암호학적으로 약한 타원곡선 E'위의 점 P'를 스마트 카드의 입력으로 주입할 경우 출력값 $d \cdot P'$역시 E'위의 점이 된다. 여기서 점 P'는 작은 위수(order) r을 갖게 되고, 타원곡선 E'의 위수는 이 r로써 나누어진다고 가정하고 있다. 그렇게 됨으로써 주어진 P'와 $d \cdot P'$를 가지고 공격자는 $d \bmod r$을 구할 수가 있게된다. 이러한 과정을 몇 개의 다른 P'로 수행하게 되면 공격자는 얻어진 몇 개의 $d \bmod r$을 가지고 중국인 나머지 정리를 이용하여 구하고자 하는 비밀키 d를 얻을 수가 있다.

2) transient 오류 공격

Bao 외 5명에 의해 소개된 오류 공격 방법은 CRT 기반이 아닌 RSA 시스템과 이산대수 문제에 기반을 둔 시스템에 관한 공격 방법이다[9]. 본 절에서는 CRT기반이 아닌 RSA 시스템의 멱승

알고리즘에 대한 공격을 ECC에서의 스칼라 곱셈 알고리즘에 적용하여 공격하는 방법을 소개한다. 이 공격법은 $Q = dP$의 계산이 일어나는 순간 오류가 발생하는 transient 오류를 가정한다.

ECC에서 스칼라 곱셈 $Q = dP$계산시 Q와 P는 공개된 값이고 d는 스마트 카드와 같은 장치에 저장된 비밀정보 값이다. 여기서 공격자가 d의 비트수를 안다고 가정할 때 d를 이진 표현 방식으로 나타내면 t비트의 d는

$$d = 2^{t-1}d_{t-1} + \ldots + 2^i d_i + \ldots + 2d_1 + d_0$$

와 같이 쓸 수 있다. 여기서 d_i는 i 번째 비트 값으로써 "1" 또는 "0"의 값을 가진다. 또한

$$\begin{aligned} Q &= d \cdot P \\ &= (2^{t-1}P)d_{t-1} + \ldots + (2^iP)d_i + \ldots + Pd_0 \\ &= P_{t-1}d_{t-1} + \ldots + P_i d_i + \ldots + P_0 d_0 \end{aligned}$$

로 표현 가능하다. 여기서, 임의의 $i \in \{0, 1, \ldots, t-1\}$에 대하여 $P_i = 2^i \cdot P$ 이다.

◆ 공격 I

$Q = dP$를 계산하는 과정에서 임의의 $i \in \{0, 1, 2, \ldots, t-1\}$에 대해 P_i에 한 비트의 오류가 발생했다고 가정하자. 오류가 발생한 P_i를 P_i'라고 쓰면 스마트 카드로부터의 결과값은

$$Q' = P_{t-1}d_{t-1} + \ldots + P_i'd_i + \ldots + P_0 d_0$$

가 된다.

이제 공격자는 올바른 결과값 Q와 오류가 발생한 결과값 Q'로부터

$$Q - Q' = (P_i - P_i')d_i$$

값을 얻을 수 있다. 이 식으로부터 공격자는

$$Q - Q' = \begin{cases} P_i - P_i' & , \text{ if } d_i = 1 \\ 0 & , \text{ if } d_i = 0 \end{cases}$$

이므로 비밀키 d의 임의의 i번째 비트 값 d_i를 얻을 수가 있다.

위의 공격 방법은 P_i가 단지 한 비트의 오류만을 가지고 있고 또한 P_i의 오류가 P_{i+1}이상의 값을 계산할 때 더 이상 영향을 미치지 않는다고 가정하고 있다.

◆ 공격 II

비밀키 d의 이진 표현법에서 하나의 비트, 즉 d_i에 오류가 있다고 가정하자. 오류가 발생한 d의 i번째 비트 d_i를 d_i'라고 하면 공격자는

$$Q' = P_{t-1}d_{t-1} + \ldots + P_i d_i' + \ldots + P_0 d_0$$

의 값을 스마트 카드로부터 얻을 수가 있다.

따라서 $Q - Q'$ 값으로부터

$$Q - Q' = P_i(d_i - d_i')$$

을 얻을 수 있고

$$Q - Q' = \begin{cases} P_i & , \text{ if } d_i = 1 \\ -P_i & , \text{ if } d_i = 0 \end{cases}$$

로부터 공격자는 $Q - Q'$와 P_i , $-P_i$를 비교하여 d_i를 알아낼 수가 있다.

3. 기존의 대응방법 및 분석

앞서 서술한바와 같이 타원곡선 스칼라 곱셈에 대한 공격의 대응 방안으로 제시된 방법은 크게 두 가지로 나뉘어 질 수 있다. 하나는 결과값을 검증하는 방법이 있고, 다른 하나는 정보를 숨기는 방법이 있다. 결과값을 검증하는 방법은 어떠한 입력값에 대한 결과값을 외부로 출력하기전에 다시 한번 계산하여 두 결과값이 동일한지에 따라 결과값을 계산하는 과정에서 오류가 발생했는지 유무를 판단하는 방법이 있고, RSA에서의 공개키와 같은 계산을 검증할 수 있는 공개 파라미터를 이용하여 결과값의 오류 발생 유무를 검증하는 방법이 있다. 정보를 숨기는 방법은 계산될 평문이나 타원곡선의 점에 대해 랜덤수를 사용하여 blinding하는 방법이 있다[9], [3].

결과값을 검증하는 방법에 있어 두 번 계산하는 방법은 가장 일반적으로 사용되는 방법이지만, 그 연산의 수행과정에 있어 연산량이 두배로 증가하게 되는 단점이 있고, 공개 파라미터를 이용하여 검증하는 방법은 앞서 설명한 오류 공격 기법중 Joye에 의해 제시된 공격 기법에는 적용이 가능하다. 왜냐하면 Joye의 논문에서는 타원곡선 기반의 새로운 암호 시스템인 KMOV에 대한 오류공격이기 때문이다. 하지만 공격 대상이 본 논문에서와 같이 타원곡선 스칼라 곱셈일 경우, 이 계산을 검증할 공개 파리미터가 존재하지 않기 때문에 적용할 수가 없다. 계산될 메시지나 점에 랜덤수를 사용하여 blinding하는 방법은 연산 수행후 올바른 결과값을 얻기 위해 blinding 된 메시지나 점에 추가적인 연산을 행하여 랜덤수를 없애 주어야 하는 단점이 있다.

III. 비밀키 blinding을 통한 대응방법

1. ECC에서의 blinding 방법

ECC에서 $Q=dP$의 연산시 비밀키 d를 blinding하는 방법은 RSA 암호 시스템에서의 멱승 알고리즘에서 비밀키 d를 $d+k\Phi(n)$으로 blinding하는 것과 유사하게 랜덤수 k를 이용하는 것이다. 여기서 $\Phi(n)$은 n에 대한 Euler totient 함수 값이다.

GF(q)상에서 정의된 타원곡선 $E(F_q)$위의 모든 점의 개수를 $\#\varepsilon$라고 하자. 여기서 $q=p^m$이다. 보통 $\#\varepsilon$를 타원곡선군의 위수(order)라고 한다. 그러면 비밀키 blinding을 통한 $Q=dP$의 연산은 아래와 같이 할 수 있다.

- n bit의 랜덤수 k를 선택한다.
  실제적으로 n은 보통 20 bit정도이다.
- $d' = d + k\cdot \#\varepsilon$를 계산한다.
- $Q = d'\cdot P = (d + k\cdot \#\varepsilon)P$
  $= dP + k\cdot \#\varepsilon P = dP + 0 = dP$

그림 2: ECC에서의 d blinding 방법

2. 연산량 및 안전성 분석

1) 연산량 분석

본 절에서 앞으로 살펴볼 타원곡선은 GF(p), p>3인 소수, 위의 타원곡선으로 가정한다. $Q = d\cdot P$의 스칼라 곱셈시 d가 160 비트의 수라면 평균적으로 1과 0이 80비트씩 있다고 보고 160번의 두배 연산과 80번의 덧셈 연산이 필요하게 된다.

연산량 분석시 체 곱셈에 대한 체 제곱과 체 역원 연산의 연산량 비는 기저체와 그 연산을 수행하는 알고리즘에 크게 영향을 받는다. 본 논문에서는 최근 발표된 논문에 의거해 S/M = 0.814, I/M=4.784로 정했다[10]. 여기서 체 역원의 연산은 IM(inversion with multiplication) 알고리즘을 사용하였다[11].

먼저 Affine 좌표계에 대해 두 점의 덧셈과 한 점의 두배연산에 필요한 연산량을 살펴보면 아래와 같다.

$$t_A(P+Q) = I+2M+S = (4.784+2+0.814)M = 7.598M$$
$$t_A(2P) = I+2M+2S = (4.784+2+(2\times 0.814))M = 8.412M$$

또한 160 비트의 비밀키 d를 사용하는 $Q = d\cdot P$의 연산시 모두 $160 \times 8.412 + 80 \times 7.598 \cong 1.954\times 10^3$번 정도의 체 곱셈 연산이 필요하게 된다.

Hasse 정리로부터 $\#\varepsilon(F_q)$, $q = p^m$는 q가 160 비트의 수 일 경우 최대 161 비트의 수까지 될 수 있음을 알 수 있다. 따라서 d를 $d' = d + k\cdot \#\varepsilon$로 blinding할 경우 d'은 최대 182 비트까지 될 수 있다. 물론 이는 최대 경우의 수이고, 일반적으로 180 비트 정도의 수가된다.

따라서 같은 방식으로 각 좌표계에서의 연산량을 계산하면 아래와 같다.

표 1: 각 좌표계에서의 타원곡선 스칼라 곱셈 연산량

	연산	# of field mult.
Affine	$Q=d\cdot P$	1.954×10^3
	$Q_b=d'\cdot P$	2.222×10^3
Projective	$Q=d\cdot P$	2.642×10^3
	$Q_b=d'\cdot P$	3.005×10^3
Jacobian	$Q=d\cdot P$	2.722×10^3
	$Q_b=d'\cdot P$	3.096×10^3

두 번 계산함으로써 결과 값의 오류발생 유무를 판단할 수 있는 double-computation 방법은 $Q = d\cdot P$ 연산을 두 번 행하는 것이므로 연산량이 배로 증가하게 된다. KMOV 암호 시스템에 대한 Joye의 공격법의 경우, 공개 파라미터 e를 이용한 $e\cdot Q = P$의 검증 과정을 통하여 $Q = d\cdot P$의 스칼라 곱셈과정에서 오류 발생 유무를 확인할 수가 있다. 이 경우 KMOV 암호 시스템에서 e를 보통 16비트 이상 사용할 것을 추천하고 있으므로, $e\cdot Q$의 계산에는 16번의 두배 연산과 8번의 덧셈 연산이 필요하게 된다[12]. 따라서 $e\cdot Q = P$의 검증과정에는 일반적인 160비트 d의 $Q=d\cdot P$ 연산에서 필요한 연산수의 10분의 1이 소요된다. 위의 표1 에서 Q_b는 비밀키 d를 blinding한 후의 결과값을 말한다.

표 2: 비밀키 blinding후의 연산량 증가분

	# of field mult.
$\Delta Q_b/Q$ *in A*	0.137
$\Delta Q_b/Q$ *in P*	0.134
$\Delta Q_b/Q$ *in J*	0.137

표2에서 $\Delta Q_b/Q$는 Q_b를 계산하는데 더 증가한 연산량 ($\Delta Q_b = Q_b - Q$)을 Q를 계산하는데 필요한 연산량으로 나눈 것이다. 위의 표에서 알 수 있듯

이 d를 $d' = d + k \cdot \#\varepsilon$으로 blinding 함으로써 각 좌표계에서 모두 13% 정도 체 곱셈수가 증가함을 알 수가 있다.

2) 안전성 분석

먼저 비밀키 d의 blinding을 통한 오류 공격의 대응 방법은 기존의 KMOV 공격법과 본 논문에서 언급한 transient 오류 공격 방법에 대해서는 안전하다. transient 오류를 이용한 공격중에서 공격 I에서는

$$Q' = P_{t-1}d_{t-1} + \ldots + P_i'd_i + \ldots + P_0d_0$$

와 같이 $P_i = 2^i \cdot P$에 비트 오류를 발생 시켜 $Q - Q' = (P_i - P_i')d_i$로부터 d_i가 1인지 0인지에 따라 $Q-Q'$가 바뀌게 됨으로써 d_i를 판단하였다. 하지만 blinding된 d, 즉 d'를 사용할 경우 t비트의 d를 알아내기 위해 알고리즘을 t번 실행시킬 때마다 $d' = d + k \cdot \#\varepsilon$의 랜덤수 k가 매번 바뀌게 되고, 이렇게 하여 생성된 d_i, for $i = 0, 1, 2, \ldots, t-1$의 조합으로 만들어진 d는 본래의 d와는 전혀 무관한 별개의 것이 된다.

마찬가지로 공격 II에서도

$$Q' = P_{t-1}d_{t-1} + \ldots + P_id_i' + \ldots + P_0d_0$$

와 같이 d_i에 한 비트 오류를 발생시켜 $Q - Q' = P_i(d_i - d_i')$로부터 d_i가 1인지 0인지에 따라 $Q-Q'$가 달라지는 것을 이용하여 d_i를 찾아내는 공격이기 때문에 d를 blinding할 경우 공격자가 찾아낸 d는 본래의 d와 전혀 무관할 것이다.

하지만 차분 오류 공격에서처럼 스마트 카드의 결과값을 이용한 공격에서는 비밀키 blinding 방법으로 대응 할 수가 없게 된다. 이 공격에서 비밀키 d를 blinding 하여도 입력값 P'가 들어가게 되면 출력값은 $d' \cdot P' = d \cdot P'$가 얻어지므로 공격자는 P'와 $d \cdot P'$를 이용하여 $d \bmod r$을 계산할 수가 있다.

3. 적용 범위

앞에서 말한 바와 같이 비밀키 blinding 기법은 처음 시차 공격에 대한 대응방법으로 소개되었다[4]. 이후 전력분석 공격에 대해서도 대응 방안으로 사용 될 수 있음이 밝혀졌다[13], [14]. 하지만 현재까지 발표된 논문에서는 하드웨어 오류공격에 대한 대응방법으로서 비밀키 blinding 방법을 사용하지 않았다.

비밀키 d의 blinding 방법은 약간의 추가적인 연산량을 필요로 하면서 지금까지 살펴본 하드웨어 오류공격과 시차공격, 전력분석 공격 모두에 대한 대응 방법으로 사용될 수가 있다.

IV. 결론

스마트 카드와 같은 부정조작에 견디는 장치에 대한 새로운 형태의 다양한 공격 방법이 소개됨에 따라 그에 대한 대응 방법의 모색 또한 필요로 하게 되었다. 본 논문에서는 기존의 시차 공격과 전력분석 공격에 대한 대응 방법으로 소개된 비밀키 blinding 방법을 다른 물리적 공격방법인 하드웨어 오류 공격 방법에 대해 적용해 보고 그에 따른 연산량과 안전성에 대해 분석하였다. 특히 타원곡선 암호 시스템에 대한 오류 공격에 대하여 살펴보았다. 본 논문에서 살펴본 대로 비밀키 blinding을 통한 대응방법은 약간의 연산량 증가만을 통하여 시차공격, 전력분석 공격, 하드웨어 오류 공격 모두에 대하여 대응 방법으로서 사용될 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, Septemper 1998.

[2] S.M. Yen, "A countermeasure against one physical cryptanalysis May Benefit Another Attack," Tech. Report TR-2K1-8, NCU LCIS, June 24, 2001.

[3] Bellcore Press Release, "New threat model breaks crypto codes," Sept. 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.

[4] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *In Proceedings of Advances in Cryptology-CRYPTO '96*, pp. 104-113, Springer-Verlag, 1996.

[5] P. Kocher, J. Jaffe, and B. Jun, "Introduction to Differential Power Analysis and Related Attacks," http://www.cryptography.com/dpa/technical, 1998.

[6] M. Joye, J.-J.Quisquater, F. Bao, and R.H.

Deng, "RSA-type sinatures in the presence of transient faults," *In Cryptography and Coding, LNCS 1355*, pp. 109-121, Springer-Verlag, Berlin, 1997.

[7] K. Koyama, U. Maurer, T. Okamoto, S. Vanstone, "New public-key schemes based on elliptic curves over the ring Z_n," *Advance in Cryptology-Crypto '91*, pp. 252-266, Springer-Verlag, 1992.

[8] I. Biehl, B. Meyer, and V. Muller, "Differential Fault Attacks on Elliptic Curve Cryptosystems," *In Advances in Cryptology - CRYPTO 2000, LNCS 1880*, pp. 131-146, Springer-Verlag, Berlin, 2000.

[9] F. Bao, R.H. Deng, Y. Han, A. Jeng, A.D. Narasimbalu, and T. Ngair, "Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults," *In Pre-proceedings of the 1997 Security Protocols Workshop*, Paris, France, 1997.

[10] 정재욱, 심상규, 이필중," $GF(p^m)$상에서 정의되는 타원곡선을 위한 복합 좌표계 응용," 통신정보보호학회 논문지, vol. 10, pp. 77-87, 2000.

[11] C.H. Lim and H.S. Hwang. "Fast Implementation fo Elliptic Cruve Arithmetic in $GF(p^m)$," *Public Key Cryptography, LNCS 1751*, pp. 405-421, Springer-Verlag, 2000.

[12] D. Bleichenbacher, "On the Security of the KMOV Public Key Cryptosystem," *Advances in Cryptology- CRYPTO '97 LNCS1294*, pp. 235-248, Springer-Verlag, 1997.

[13] J. S. Coron, "Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems," *In Workshop on Cryptographic Hardware and Embedded Systems*, pp. 292-302, Springer-Verlag, 1999.

[14] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power Analysis Attacks on Moular Exponentiation in Smartcards," *In Proceedings of Workshop on Cryptographic Hardware and Embedded Systems*, pp. 144-157, Springer-Verlag, 1999.