Exponent Blinding 기법에 대한 전력 공격

김형섭[1)2)], 백유진[1)], 김승주[2)], 원동호[2)*]

1)삼성전자 System LSI사업부 SoC 연구소

2)성균관대학교 정보통신공학부 정보보호연구소

Power Attack against an Exponent Blinding Method

Hyungsup Kim[1)2)], Yoojin Baek[1)], Seungjoo KIm[2)], Dongho Won[2)*]

1)Samsung Electronics, System LSI Division, Soc R&D Center

2)Information Security Group, School of Information and Communication Engineering, Sungkyunkwan University

요 약

전력 공격은 암호화 연산 과정 중 발생하는 소비 전력의 파형을 측정하여 비밀 정보를 알아내는 공격 방식이다. 이러한 전력 공격에 대한 취약성을 막기 위하여 message blinding, exponent blinding과 같은 기법들이 적용되어 왔다. 본 고에서는 ECC[1]암호화 연산 과정에서, r이 임의의 정수일 때, $dP=(d-r)P+rP$ 인 관계를 이용하는 exponent blinding기법[2]에 대하여 언급하고, 위 기법을 전력 공격의 대응책으로 적용 시 적절히 구현되지 않으면 power attack에 대하여 매우 취약하다는 것을 보인다.

I. 서론

암호 시스템은 동작 중에 내부의 비밀 정보를 외부로 누출하지 않아야 한다. 하지만 암호 연산을 구성하는 내부 회로들은 부 채널 정보들을 유출 할 수 있으며 암호 시스템은 이러한 정보를 이용하여 공격당할 수 있다. 이러한 공격을 부채널 공격(Side Channel Attack)이라 한다. Paul Kocher가 1996년에 timing attack[3]을 발표한 후, Simple Power Attack(SPA), Differential Power Attack(DPA)[4]과 같은 많은 부채널 공격 방식들이 제안되었다. SPA는 하나의 입력 값에 대한 암호 연산 과정의 소비 전력 파형을 관측하여 비밀 정보를 알아내는 방식이며, DPA는 여러 입력 값에 대한 소비 전력 파형을 통계적 기법을 이용하여 비밀 정보를 알아내는 공격 방식이다. 이러한 DPA에 대한 대응책의 하나로 exponent blinding 기법이 있으며 M. Ciet등은 참고 논문[2] 에서 exponent blinding 기법의 일종으로 $dP=rP+(d-r)P$ 인 관계와 Shamir의 simultaneous multiple points exponentiation trick[5]을 적용한 Multiplier Randomization 방식을 제안하였다. 그러나 본 고에서는 이 방어책이 DPA에 취약함을 보인다.

* 교신저자 : 원동호(dhwon@security.re.kr)

II. 본론

이번 장에서는 Exponent Blinding 기법의 일종인 Multiplier Randomization 방식에 대하여 설명하고, 이 방식에 DPA를 적용 시 필요한 정리를 유도한 후 이 정리를 이용하여 Multiplier Randomization 방식이 적절히 구현되지 않으면 안전하지 않다는 것을 보인다.

2.1 전력 공격

전력 공격의 기본 개념은 암호 시스템의 소비 전력 파형은 시스템 내부의 연산 과정과 밀

접한 연관이 있어서, 외부에서 소비 전력 파형을 분석하면 시스템 내부에서 어떤 연산이 수행되었는지 알 수 있다는 것이다. 아래 알고리즘은 일반적으로 dP를 계산하는 알고리즘이다.

Algorithm 1 Simple doubling and add method

Input : $d=\sum_{i=0}^{n-1} d_i 2^i, P$

Output : dP

1. Set $S=P$;
2. For $i=n-2 to 0 by -1, do$
   2.1 if $d_i=0$ then, $S=2S$;
   2.2 else $S=2S+P$;
3. Return S.

위의 Algorithm 1을 예로 들면 Step 2에서 d_i의 값에 따라 Step 2.1 아니면 Step 2.2가 수행되며, 덧셈 또는 곱셈 연산 시에 암호 시스템 내부에서 소비되는 전력량이 다르므로 소비전력 파형에 차이가 발생 되고, 그 차이를 측정하여 d_i값을 유추하는 것이 전력 공격이다.

이와 같은 전력 공격은 ECC를 포함한 모든 암호 시스템에 적용 가능하며, 이를 막기 위한 다양한 방어책들이 제시되어 왔다.

2.2 Multiplier Randomization 기법

M. Ciet등은 참고 논문[2] 에서 새로운 DPA 방어법인 Point Randomization기법과 Multiplier Randomization기법을 제안하였다. 본 고에서는 Multiplier Randomization 기법에 대하여 DPA를 이용한 공격이 가능하다는 것을 보인다.

Multiplier Randomization 기법은 multiplier d와 타원 곡선위의 한 점 P에 대하여 $Q=dP$를 계산시, 먼저 랜덤한 정수 r $(0 \le r < d)$을 선택한 후 이를 이용하여 $Q=(d-r)P+rP$를 계산하는 방식이다. 또 다른 가능한 방법은 d를 $d=(d/r)\cdot r+(d \bmod r)$로 분해하는 것이다. 이 경우 $S=rP$라고 하면, $Q=dP$는 다음과 같이 계산 될 수 있다: $Q=d_1P+d_2S$. 이때, $d_1=d \bmod r$이고 $d_2=d/r$이다. 위 식은 $Q=(d-r)P+rP$방식과 달라 보이지만, $r'=d \bmod r$ 라 하면, $(d/r)\cdot r=d-r'$이기 때문에 결국 두 방식은 표현만 다를 뿐 근본적으로 같은 방식이다.

Multiplier Randomization 기법을 구현 시 $(d-r)P$와 rP를 serial하게 계산한 후 더하는 방법은 부채널 공격에 대하여 안전하게 보이지만 연산 시간이 두 배로 늘어나므로 실효성이 떨어진다. 따라서 M. Ciet등은 참고 논문[2]에서 $(d-r)P$와 rP를 Shamir의 simultaneous multiple points exponentiation trick[5]을 이용하여 parallel하게 계산하는 방법을 제안하였다. 따라서 이 방법은 $(d-r)=e$라고 했을 때 아래 Algorithm 2 와 같이 구현될 수 있다.

Algorithm 2 Multiplier Randomization

Input : $\sum_{i=0}^{n-1} d_i 2^i, p$

Output : dP

1. Choose a Random integer $r=\sum_{i=0}^{n-1} r_i 2^i$ such that $0 \le r < d$
2. Calculate $d-r=\sum_{i=0}^{n-1} e_i 2^i$ 계산.
3. $T[0]=O, T[1]=P, T[2]=2P$
4. $S=T[0]$;
5. For $i=n-1$ to $i=0$ by -1, do
   5.1 $S=2S$;
   5.2 $S=S+T[r_i+e_i]$;
6. Return S.

상기 알고리즘에서 $r_i=e_i=0$ 일 때 Step 5.2에서 아무 연산도 안 일어나므로, Algorithm 2는 SPA에 매우 취약함을 알 수 있다. 이를 보완하기 위하여 M. Ciet등은 상기 알고리즘에 Montgomery ladder를 적용 할 것을 제안 하였지만 본 고에서는 분석의 용이성을 위하여 Montgomery ladder부분을 생략하였다. 하지만 Algorithm 2는 DPA에 대한 방어책으로서는 M. Ciet등이 작성한 참고 논문[2]에서 제안된 방식과 동일하다. 따라서 본 고에서 제안한 전

력 공격 방식은 Montgomery ladder가 적용 되더라도 동일하게 적용될 수 있다.

다음 절에서는 Algorithm 2에 대하여 DPA를 하기 위한 기본 정리를 도출하며 이를 이용하여 Algorithm 2가 적절히 구현되지 않으면 안전하지 않다는 것을 보인다.

2.3 DPA를 위한 정리 및 증명

Algorithm 2에 대한 전력 공격을 위하여 다음과 같은 표기법을 채택한다. 비밀 키인 양의 정수 $d=\sum_{i=0}^{n-1} d_i 2^i, d_i \in \{0,1\}$와 d상의 특정 위치를 가리키는 index값을 k $(0 \le k \le n-1)$라 하면, k부터 $n-1$까지 범위의 상위 d값은 $d_H^k := d_H = \sum_{i=k}^{n-1} d_i 2^{i-k}$로 표기하고, 0부터 $k-1$까지 범위의 하위 d 값은 $d_L^k := d_L = \sum_{i=0}^{k-1} d_i 2^i$로 표기한다. 그리고 r이 주어진 랜덤한 정수이고 dP나 rP 또는 $(d-r)P$를 Algorithm 1을 이용하여 계산할 때, $i=k$가 될 때 까지 계산한 결과 값을 각각 $D_k(P)$, $R_k(P)$, $E_k(P)$라 표기한다. 이때 $d=2^k d_H+d_L$이고 $D_k(P)=d_H P$, $R_k(P)=r_H P$, $E_k(P)=(d-r)_H P$이다.

이와 같이 표기할 때, 본 고에서 주장하는 것은 DPA에 대한 방어책이 적용된 Algorithm 2를 이용하여 dP를 계산할 경우에 $i=k$가 될 때 까지 계산한 결과 값은$R_k(P)+E_k(P)$가 되며, DPA에 대한 방어책이 적용되지 않은 Algorithm 1을 이용한 $D_k(P)$는 carry의 발생 여부에 따라 bias된 확률(1/2보다 크거나 작은 값을 갖는 확률)로 $R_k(P)+E_k(P)$ 또는 $R_k(P)+E_k(P)+P$ 값을 갖는다는 것이다. 따라서 우리가 $D_k(P)=R_k(P)+E_k(P)$인지 $D_k(P)=R_k(P)+E_k(P)+P$ 인지 구분할 수 있으면 Algorithm 2에 대하여 DPA가 가능하다. 이에 대한 자세한 분석을 수행하기 위하여 $d_L=r_L+(d-r)_L$ 이면 $d_H=r_H+(d-r)_H$가 되며, 이런 조건에서만 $D_k=R_k+E_k$가 된다는 사실에 기반을 두어 정리 1을 내린다.

정리 1.

1) 주어진 양의 정수 d에 대하여

만약 r이 $\{0,\cdots,2^n-1\}$범위에서 선택되면

$$Pr(r_L+(d-r)_L=d_L)=\frac{d_L+1}{2^k} \quad (1)$$

이고, r이 $\{0,\cdots,d-1\}$범위에서 선택되면

$$Pr(r_L+(d-r)_L=d_L)=\frac{d_H d_L+d_H+d_L}{d} \quad (2)$$

이 된다.

2) 만약 $d_{k-1}=1$이면, 확률 (1)과 (2)는 1/2보다 크다.

3) 만약 $d_{k-1}=0$이고 k값의 범위가 $2 \le k \le n-2$ 이면 확률 (1)은 항상 1/2 보다 작거나 같고, 확률 (2)는 매우 높은 확률로 1/2보다 작거나 같다.

위 정리의 증명을 위하여 우리는 다음 사실에 주목한다. $r_L+(d-r)_L=d_L$이 성립되기 위해서는 r_L과 $(d-r)_L$값의 합이 carry를 발생하지 않아야 하며, 이는 r_L값이 $r_L \le d_L$인 조건을 만족하는 경우이다.

이와 같은 사실에 기반 하여, 정리 1의 직관적인 증명을 위하여 r이 $\{0,\cdots,2^n-1\}$중에서 선택되는 경우를 가정하여 먼저 증명하고, 그 후 Algorithm 2의 r값 범위를 만족하도록 r이 $\{0,\cdots,d-1\}$중에서 선택되는 경우를 증명한다.

먼저 r이 $\{0,\cdots,2^n-1\}$중에서 선택된다고 가정하면, r_L은 $\{0,\cdots,2^k-1\}$중에서 선택된 랜덤한 정수임이 분명하므로, $r_L \le d_L$을 만족하는 r_L을 선택할 수 있는 경우의 수를 따져 보면 확률 (1)이 나온다. 확률 (1)에 대한 정리 1의 2), 3)을 증명하기 위해 다음 항등식을 이용한다.

$$\frac{d_L+1}{2^k}-\frac{1}{2}=\frac{2d_L+2-2^k}{2^{k+2}} \quad (3)$$

만약 $d_{k-1}=1$이면, $d_L \geq 2^{k-1}$이므로 식(3)는 항상 0보다 크며, 만약 $d_{k-1}=0$이면, $d_L \leq 2^{k-1}-1$ 이므로 식(3)은 항상 0과 같거나 작으므로 확률 (1)에 대한 정리 1의 2), 3)이 증명된다.

r이 $\{0,\cdots,d-1\}$범위에서 선택된다고 가정하면, $r_L \leq d_L$인 경우는 $r_H < d_H$ 이고 $r_L < d_L$이거나, $r_H = d_H$ 이고 $r_L < d_L$인 경우뿐이다. 첫 번째 경우의 수는 $d_H(1+d_L)$이고 두 번째 경우의 수는 d_L이며 둘을 합하면 정리 1의 확률 (2)가 된다. 확률 (2)에 대한 정리 1의 2), 3)을 증명하기 위하여 $d=2^k d_H+d_L$인 관계를 적용한 다음 항등식을 이용한다.

$$\frac{d_H d_L+d_H+d_L}{d}-\frac{1}{2}$$
$$=\frac{2d_H d_L+d_L+2d_H-2^k d_H}{2d} \quad (4)$$

만약 $d_{k-1}=1$이면, $d_L \geq 2^{k-1}$이므로 식(4)는 항상 0보다 커서 확률 (2)에 대한 정리 1의 2)는 증명된다. 그러나 만약 $d_{k-1}=0$이면, 식(4)에서 분자의 값이 양수일지 음수일지 쉽게 판별하기 어렵다. 이러한 이유로 기댓값을 이용한다. d_H와 d_L에 대한 기댓값 $E(d_H)$와 $E(d_L)$은 다음과 같이 구할 수 있다. 만약 $d_{k-1}=0$이고, d_H 와 d_L이 각각 $\{0,\cdots,2^{n-k}-1\}$과 $\{0,\cdots,2^{k-1}-1\}$ 범위에서 선택된 값이라면, 우리는 다음과 같은 기댓값을 얻을 수 있다.

$$E(d_H)=\frac{2^{n-k}-1}{2}$$
$$E(d_L)=\frac{2^{k-1}-1}{2}.$$

위 기댓값을 식(4)의 분자에 적용하면

$$E(2d_H d_L+2d_H+d_L-2^k d_H)$$
$$=2E(d_H)E(d_L)+2E(d_H)E(d_H)$$
$$=\frac{-2^{n-1}+2^{n-k}+2^k-2}{2}$$

가 된다. 이 값은 $2 \leq k \leq n-2$일 경우에는 명확히 음수 값이다. 이것으로 확률(2)에 대한 정리 1의 3)은 증명된다.

정리 1을 확증하기 위하여 d_{k-1}의 값에 따라 $r_L+(d-r)_L=d_L$인 경우가 발생하는 횟수를 시뮬레이션 하였다. 이 때, $n=160$이며, d는 n비트의 값, k의 범위는 $2 \leq k \leq n-2$, r의 범위는 $0 \leq r < d$ 라는 각각의 조건을 만족하는 랜덤하게 선택된 d, k, r 값을 이용하여 d와 k값이 1번 바뀔 때 마다 r값을 10000번 바꿔가면서 $r_L \leq d_L$인 경우(즉 $r_L+(d-r)_L=d_L$을 만족하는 경우)가 발생하는 횟수를 카운트 하였다. 이러한 방식의 시뮬레이션을 200회 반복하여 그 결과를 그림 1로 나타내었다.

시뮬레이션은 Visual C++과 OpenSSL 0.9.7e library를 사용하여 수행하였다.

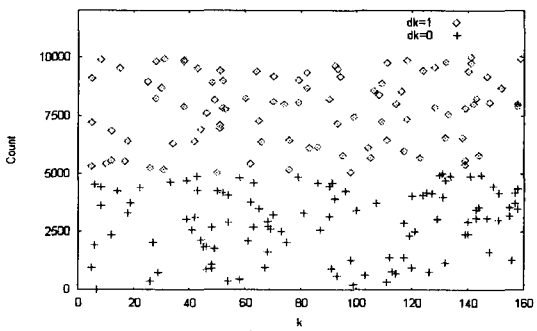

그림 1. 시뮬레이션 수행 결과

시뮬레이션 수행 결과 정리 1의 예측대로 $d_{k-1}=1$이면 $r_L+(d-r)_L=d_L$인 경우는 5000번 이상 발생하여 발생 확률이 1/2보다 컸으며, $d_{k-1}=0$이면 $r_L+(d-r)_L=d_L$인 경우는 5000번 이하로 발생하여 1/2보다 낮은 발생 확률을 보였다.

2.4 Algorithm 2에 대한 DPA

이 절에서는 Algorithm 2에 대한 DPA 방법을 설명한다. DPA를 하기 위하여 랜덤한 입력 points $P_1,\cdots,P(t)$에 대한 소비 전력 파형 $C(1),\cdots,C(t)$를 수집한다. 그 후 d의 상위 비트 $d_{n-1},\cdots,d_k$값을 알고 있다는 전제 조건 하에 $d_{k-1}=1$이라 가정하고, $1 \le r \le t$의 범위에 대하여 $Q_r=(\sum_{i=k-1}^{n-1} d_i)P_r$을 계산한 후 Q_r의 특정한 한 비트의 값이 1이 되는 P_r들을 S_1으로 분류하고 나머지는 S_0로 분류한다. 이렇게 분류된 S_1, S_0에 대한 소비 전력 파형의 차를 구한다.

$$< C(r) >_{P_r \in S_1} - < C(r) >_{P_r \in S_0}$$

정리 1에 따르면 $d_{k-1}=1$ 일 때 $Q = R(P)+E(P)$가 될 확률과, $d_{k-1}=0$일 때 $Q = R(P)+E(P)+P$가 될 확률은 모두 1/2보다 크다. 그러므로 위에서 가정한 $d_{k-1}=1$이라는 것이 사실이면 P_r은 S_1과 S_0로 제대로 분류 되어 소비 전력 파형의 차는 peak를 나타낼 것이고 그렇지 않으면 ($d_{k-1}=0$) peak를 나타내지 않는다. 이런 방식으로 d의 $k-1$번째 비트 값을 알아낼 수 있으며 d의 나머지 비트 값에 대하여도 동일하게 이 공격 방식을 적용할 수 있다.

마지막으로 고려해야 할 점은 위 과정에서 d_{k-1}값 혹은 그 이후의 값을 잘못 추측하였을 때 그것을 어떻게 알아낼 지에 관한 것이다.

만약 우리가 d_{k-1}값을 잘못 가정하게 되면 Q_r 계산 값이 틀려지게 되어 올바른 포인터 집합 S_1, S_0을 준비 할 수 없게 되고, 그 결과 소비 전력 파형의 차에도 peak가 없거나 매우 적게 나타나게 된다. 우리는 DPA를 재귀적 방법으로 사용하므로 한번 잘못된 추측을 하면 그 이후의 모든 DPA에 대하여 peak가 발생하지 않게 된다. 이러한 경우 d_{k-1}값에 대한 우리의 추측이 틀렸다는 것을 인지할 수 있다.

III. 결론

본 논문에서는 ICICS'03에서 제안된 ECC상의 DPA방어 기법이 일반적인 ECC pointer multiplication연산 과정의 중간 결과 값과 높은 확률로 유사하게 됨으로써 DPA에 대하여 매우 취약하다는 것을 증명하였다. 따라서 암호 시스템 구현 시 내부 연산과정의 Algorithm은 매우 주의하여 구현해야 한다.

[참고문헌]

[1] N.Koblitz, "Elliptic curve cryptosystems," In *Mathematics of Computation*, vol.48, no.177, pp. 203-209, Jan.1987.

[2] Mathieu Ciet, Marc Joye. "(Virtually) Free Randomization Techniques for Elliptic Curve Cryptography," *ICICS2003*, pp. 348-359, Springer-Verlag, 2003.

[3] Paul Kocher. "Timing attacks on implementations of Diffie-Hellman, RSA,DSS, and other systems," In N. Koblitz,editor, *Advances in Cryptology - CRYPTO'96, volume 1109 of Lecture Note in Computer Science*, pp. 104-113, Springer-Verlag, 1996.

[4] Paul Kocher, Joshua Jaffe, and benjamin Jun. "Differential power analysis," *Advanced in Cryptology - CRYPTO'99, LNCS 1666*, pp.388-397, Aug.1999

[5] Jerome A.Solinas, "Low-weight binary representations for pair of integers," *Tech.Report CORR 2001/41*, CACR, Waterloo, 2001.