Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.1, Feb. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.1.261

m-ary 멱승 연산에 대한 새로운 단순 전력 분석 공격

안 성 준,[1†] 최 두 호,[2] 하 재 철[3‡]
[1]과학기술연합대학원대학교, [2]한국전자통신연구원, [3]호서대학교

A New Simple Power Analysis Attack on the m-ary Exponentiation Implementation

Sung-jun Ahn,[1†] Doo-ho Choi,[2] Jae-Cheol Ha[3‡]
[1]University of Science and Technology, [2]ETRI, [3]Hoseo University

요 약

RSA와 같은 공개 키 암호시스템을 구현할 경우 멱승 연산이 사용되며 이를 고속화하기 위한 방법들이 연구되었다. 반면, 공격자들은 비밀 키를 얻기 위해 멱승 알고리듬의 연산 과정에서 발생하는 소비 전력이나 전자기파를 이용하는 부채널 공격을 시도해 왔다. 본 논문에서는 멱승의 고속화를 위해 사용되는 알고리듬 중 m-ary 멱승 알고리듬에 대한 새로운 전력 분석 공격을 제시하고자 한다. 제안 공격 방식의 핵심은 사전 계산을 사용하는 멱승 연산에서 비밀 키와 연관성을 가진 소비 전력 패턴을 구별해 낼 수 있도록 공격자가 입력 메시지를 조정하는 것이다. 논문에서는 m-ary 알고리듬을 실제 실험용 보드에 구현한 후 제안된 단순 전력 분석 공격을 시도한 결과, 멱승에 사용된 비밀 키가 노출되는 취약성이 있음을 확인하였다.

ABSTRACT

There are many researches on fast exponentiation algorithm which is used to implement a public key cryptosystem such as RSA. On the other hand, the malicious attacker has tried various side-channel attacks to extract the secret key. In these attacks, an attacker uses the power consumption or electromagnetic radiation of cryptographic devices which is measured during computation of exponentiation algorithm. In this paper, we propose a novel simple power analysis attack on m-ary exponentiation implementation. The core idea of our attack on m-ary exponentiation with pre-computation process is that an attacker controls the input message to identify the power consumption patterns which are related with secret key. Furthermore, we implement the m-ary exponentiation on evaluation board and apply our simple power analysis attack to it. As a result, we verify that the secret key can be revealed in experimental environment.

Keywords: RSA, Side-channel attack, Simple power analysis, m-ary exponentiation

I. 서 론

정보를 보호하기 위한 RSA(Rivest Shamir and Adelman)나 ElGamal과 같은 공개키 암호시스템이 제안된 이후 이를 효과적으로 구현하고자 하는 연구들이 있었다[1, 2]. 이러한 공개키 암호시스템에서는 큰 정수에 대한 모듈라 멱승을 수행하게 된다. 그리고 멱승 연산에 대한 효율성은 알고리듬의 안전성을 유지할 수 있는 큰 정수에 대한 수행 시간이나 사용되는 메모리의 양을 기준으로 삼아 왔다.

그 동안 멱승 연산을 위해서는 기본적으로 이진 방법(binary method), 이를 확장시킨 m-ary 방법 그리고 윈도우 방법 등이 사용되어 왔다[3, 4]. 이진 방식은 연산 속도는 다소 떨어지지만 사용하는 메모리

접수일(2013년 12월 31일), 게재확정일(2014년 1월 23일)
† 주저자, asj503@etri.re.kr
‡ 교신저자, jcha@hoseo.edu (Corresponding author)

가 적어 낮은 성능과 저 메모리 환경의 디바이스를 구현할 때 주로 사용되고 있다. 반면 m-ary나 윈도우 방식은 사전 계산 과정이 필수적이므로 많은 양의 메모리를 필요로 하지만 멱승을 고속으로 처리할 수 있는 장점이 있다. 최근에는 암호용 디바이스의 구현 환경이 개선되어 사용되는 메모리가 어느 정도 필요하더라도 고속 구현을 위해 m-ary나 윈도우 멱승 알고리듬을 사용하기도 한다.

공개 키 암호 시스템에서의 멱승 연산은 보통 수백 번의 모듈라 곱셈 연산과 자승 연산으로 이루어져 있는데 이를 소프트웨어적인 기법으로 구현한 후 암호용 디바이스에 탑재하는 경우도 있으며 FPGA나 전용 칩 형태로 구현되기도 한다. 본 논문에서는 소프트웨어적인 구현 기법을 이용하여 멱승 알고리듬을 개발하여 이를 스마트 카드나 임베디드 모듈에서 구동하는 환경을 가정한다.

한편, 공개 키 암호 시스템을 공격하기 위한 방법으로 알고리듬을 디바이스에 구현하는 과정에서 생긴 결함이나 취약점을 이용하는 부채널 공격(Side-Channel Attack, SCA)이 최근 주목을 받고 있다. 부채널 공격은 암호용 디바이스에 비밀 키 정보를 내장하고 있는 시스템에서 암호 알고리듬을 수행할 때 발생하는 소비 전력이나 전자기파 등을 수집하여 이를 분석함으로써 비밀 키를 추출하는 공격 기법이다[5-7]. 지금까지 제안된 부채널 공격 방법 중에서 소비 전력 측정 기법을 이용하여 비밀 키를 추출하는 전력 분석(Power Analysis, PA) 공격이 많이 사용되고 있는데, 이 공격 방법은 크게 단순 전력 분석(Simple Power Analysis, SPA)과 차분 전력 분석(Differential Power Analysis, DPA) 공격으로 나누어진다.

단순 전력 분석 공격은 디바이스가 연산 과정에서 소모하는 하나의(혹은 평균) 전력 파형을 분석하여 비밀 키 정보를 찾아내는 기법이다. 반면 차분 전력 분석 공격 기법은 수십 개 혹은 수백 개의 전력 파형을 모아 이를 통계적 기법을 통해 분석함으로써 비밀 키를 찾아내는 방법이다. 이러한 전력 분석 공격에 대응하기 위해서는 크게 전력 소모 정보의 누출을 막기 위한 숨김(hiding) 기법을 사용하거나 알고리듬을 변경하거나 수행되는 데이터를 감추기 위해 마스킹(masking) 이나 블라인딩(blinding) 기법이 사용되고 있다[8, 9]. 또한, 부채널 공격이 수동적인 공격 방법인 것에 반해 능동적인 물리적 공격 기법으로 오류 주입 공격(Fault Attack, FA)이 제안되기도 하였으며[10-12] 최근에는 단순 전력 분석 공격과 오류 공격을 결합한 조합 공격(Combined Attack, CA)이 제안되기도 하였다[13, 14].

본 논문에서는 사전 계산에 기반한 m-ary나 윈도우 멱승 알고리듬에 대해 새로운 단순 전력 분석 공격 기법을 제시하고자 한다. 제안하는 단순 전력 분석 공격의 핵심은 멱승 연산에서 비밀 키와 연관성을 가진 소비 전력 패턴을 구별해 낼 수 있도록 공격자가 입력 메시지를 조정하는 것이다. 그리고 m-ary 알고리듬을 실제 실험용 보드에 구현한 후 제안된 단순 전력 분석 공격을 시도해 보았다. 실험 결과, 1024비트의 멱승 연산을 순수하게 m-ary 기법으로 구현할 경우 사용된 비밀 키가 공격자에게 노출될 수 있음을 확인하였다.

논문의 2장에서는 m-ary 멱승 알고리듬과 전력 분석 공격에 대해 정리한다. 또한 3장에서는 본 논문에서 제안한 단순 전력 분석 공격의 핵심 이론과 공격 방법을 설명한다. 4장에서 구체적인 실험 결과 및 공격에 대한 대응책을 제시하고 5장에서 결론을 맺는다.

II. 멱승 알고리듬과 전력 분석 공격

2.1 이진 멱승 방법과 단순 전력분석 공격

RSA와 같은 공개키 암호 시스템에서는 1024비트 이상의 모듈라 멱승 연산을 필요로 한다. 일반적으로 RSA 암호 시스템에서 메시지 M에 대한 개인 키 d의 서명을 다음과 같이 나타낸다.

$$C = M^d \bmod N$$

여기서 k를 d의 비트 수라고 가정하면 $k = 1 + \lfloor \log_2 d \rfloor$와 같이 쓸 수 있으며 지수 d를 이진수로 나타내면 아래와 같다.

$$d = (d_{k-1} d_{k-2} \cdots d_1 d_0) = \sum_{i=0}^{k-1} d_i 2^i,\ d_i \in \{0,1\}$$

이러한 멱승 연산을 효과적으로 수행하기 위해 이진 멱승 방법을 사용하는데 이진 멱승 방식은 사용되는 지수 d를 왼쪽 최상위 비트(Most Significant Bit, MSB)부터 처리하는 지, 아니면 오른쪽 최하위 비트(Least Significant Bit, LSB)부터 처리하는 지의 여부에 따라 Left-to-Right 방식과 Right-to-Left 방식으로 나누어진다.

다음 Fig. 1은 Left-to-Right 이진 멱승 방법을 나타낸 것이다. 여기서 지수의 최상위 비트는 1이라

가정한다. 그림에서 보면 지수 비트를 최상위 비트부터 탐색하면서 0이면 자승 연산만, 1이면 자승과 곱셈 연산을 수행하게 된다. 따라서 k번의 자승과 $H(d)$번의 곱셈 연산이 필요하다. 여기서 $H(d)$는 d에 대한 Hamming weight 값을 나타낸 것으로 평균 $0.5k$이다.

입력 : M, d, N
출력 : $C = M^d \bmod N$

1. $C = 1$
2. for $i = k-1$ downto 0 do {
3. $\quad C = C \cdot C \bmod N$
4. $\quad$ if($d_i = 1$) $C = C \cdot M \bmod N$ }
5. Return C

Fig.1. Left-to-Right binary algorithm

그러나 Fig. 1의 이진 방식은 자승 연산과 곱셈 연산이 서로 다른 전력 소비량을 보이거나 단계 4의 분기문에서 분기 결과가 파형 측정을 통해 구별이 가능하다면 단순 전력 분석을 통해 비밀 키 d가 노출되는 취약점을 가지고 있다.

따라서 단순 전력 분석 공격에 강인한 이진 멱승 알고리듬이 제안되었는데[9] 이 알고리듬은 비밀 키 비트와 무관하게 항상 자승과 곱셈을 수행한다는 의미에서 square-and-multiply-always 방법으로 불린다. 다음 Fig. 2는 단순 전력 분석에 강인한 이진 멱승 알고리듬을 나타낸 것이다[15]. 여기서 $\neg d_i$는 d_i에 대한 부정(negative) 논리를 의미하는 것으로 d_i가 0이면 1을, d_i가 0이 아니면 0을 출력하는 논리 연산이다. 이 멱승 방식에서는 큰 정수의 중간 값을 저장하기 위한 두 개의 메모리가 필요하며 총 k번의 자승과 k번의 곱셈을 수행하게 되므로 연산량이 Fig. 1의 알고리듬에 비해 평균 $0.5k$번의 곱셈이 추가적으로 필요하다.

입력 : M, d, N
출력 : $C = M^d \bmod N$

1. $C_0 = 1$
2. for $i = k-1$ downto 0 do {
3. $\quad b = \neg d_i$
4. $\quad C_0 = C_0 \cdot C_0 \bmod N$
5. $\quad C_b = C_0 \cdot M \bmod N$ }
6. Return C_0

Fig.2. SPA-protected Left-to-Right binary algorithm

그러나 Fig. 2의 멱승 알고리듬은 C-Safe Error 공격이라는 오류 주입 공격에 취약함이 밝혀졌다[16]. 그 후 단순 전력 분석 공격과 C-Safe Error 공격을 동시에 방어할 수 있는 알고리듬으로 Montgomery Ladder 멱승 기법이 제안되기도 하였다[17]. 그러나 이 방법 역시 RDA(Relative Doubling Attack)라는 공격에 의해 비밀 키가 노출할 수 있음이 밝혀졌다[15].

2.2 m-ary 멱승 방법

이진 멱승 방법은 사용하는 메모리가 적어 컴퓨팅 성능이 낮거나 가용 메모리 양이 적은 환경에서 주로 구현되어 사용되었다. 그러나 최근에는 컴퓨팅 능력이 향상된 일반 기기에 암호 알고리듬을 탑재하여 사용하는 경우가 많아져 고속 구현에 중점을 두고 멱승 알고리듬을 설계하고 있다. 예로서 안드로이드 기반의 스마트 폰에서도 멱승을 위해 이진 알고리듬을 이용하는 것이 아니라 윈도우 알고리듬을 사용하고 있다[18]. 이러한 관점에서 본 논문에서는 사전 계산 기법을 이용하는 m-ary 방식에 초점을 맞추어 부채널 공격에 대한 취약성을 분석하고자 한다.

m-ary 멱승 알고리듬은 지수 비트를 한 비트씩 처리하는 것이 아니라 한번에 2비트 이상씩을 탐색하여 멱승을 수행하는 방식이다. 따라서 한번에 처리하는 지수 비트가 r비트이고 $m = 2^r$일 때 m-ary 멱승 방법이라고 부른다. 즉, 비밀 키 d를 r비트씩 나누어 s개의 블록으로 표현할 수 있는데 지수 d와 계수 f_i를 다음과 같이 나타낼 수 있다.

$$d = \sum_{i=0}^{k-1} d_i 2^i = \sum_{i=0}^{s-1} f_i (2^r)^i = \sum_{i=0}^{s-1} f_i \, m^i$$

$$f_i = (d_{ir+r-1} d_{ir+r-2} \cdots d_{ir}) = \sum_{j=0}^{r-1} d_{ir+j} 2^j$$

m-ary 멱승 방법에서는 사전에 M^{f_i}값을 계산해 두고 연산 시에 필요한 값을 사용한다. 그리고 지수 d를 한번에 r비트씩 좌측에서 우측으로 탐색하면서 멱승 연산을 수행한다. m-ary 멱승 알고리듬을 나타낸 것이 Fig. 3이다. 그림에서 보는 바와 같이 단계 5에서 r번의 제곱 연산을 수행하며, 단계 6에서 f_i의 값에 따라 곱셈 연산을 수행한다. 단, S_0값은 실제 출력과는 전혀 관계없는 것으로서 f_i가 0일 때만 곱해주는 더미(dummy) 연산용 값이므로 임의의 값으로 지정

할 수 있다. Fig. 3의 멱승 알고리듬은 Fig. 2를 m-ary 형태로 변경한 것으로서 단순 전력 분석 공격에는 안전한 것으로 알려져 있다. 다만 계수 f_i가 0인 경우에는 C-safe Error 공격에 의해 일부 키가 노출될 위험성이 있다[16].

입력 : M, d, N
출력 : $C = M^d \bmod N$

1. $S_{f_i} = M^{f_i} \bmod N$ for $f_i = 1, 2, ., m-1$
   S_0 = temporary value
2. $C_0 = 1$
3. for $i = s-1$ downto 0 do {
4. $b = \neg f_i$
5. $C_0 = C_0^{2^r} \bmod N$
6. $C_b = C_0 \cdot S_{f_i} \bmod N$ }
7. Return C_0

Fig.3. Left-to-Right m-ary algorithm

m-ary 멱승의 경우 사전 계산 단계인 1단계에서 에서 $2^r - 2$개의 곱셈이 필요하다. 그리고 단계 5에서 k번의 자승 그리고 단계 6에서 $\lceil k/r \rceil$번의 곱셈이 필요하다. Fig. 2의 이진 방식과 비교하여 자승 연산 수는 같으나 곱셈 수는 $k - 2^r + 2 - \lceil k/r \rceil$번을 줄일 수 있다. 예를 들어 1024비트 연산을 가정하고 r이 4이면 754번 정도의 곱셈을 줄일 수 있으며 r이 5이면 789번 정도의 곱셈을 줄일 수 있어 연산 속도 면에서는 매우 효율적이다. 그러나 r이 4이면 16개 정도의 1024비트 정수를 저장할 메모리가 필요하고, r이 5이면 32개 정도의 메모리가 필요하다. 즉, m-ary 멱승에서는 r이 1비트 늘어나면 메모리는 2배로 늘어나게 된다.

III. m-ary 멱승 알고리듬에 대한 단순 전력분석 공격

본 논문에서는 m-ary 멱승 알고리듬을 구현하여 사용할 경우 비밀 키를 추출할 수 있는 공격 방법을 제안하고자 한다. 공격의 최종적인 목표는 Fig. 3에서 비밀 키 d의 계수인 각 f_i를 찾는 것이다. 그러나 일반적인 경우 Fig. 3의 단계 6에서의 연산 파형만 관측해서는 비밀 성분 f_i를 알 수 없다. 따라서 본 논문에서는 새로운 접근 방식을 통해 비밀 키의 계수 f_i를 찾아내고자 한다.

제안 공격 방법의 핵심은 단계 6의 연산을 비밀 계수 값에 따라 구별이 가능하도록 입력 메시지 M을 사용하는 것이다. 즉, 단계 6의 $C_b = C_0 \cdot S_{f_i} \bmod N$ 연산에서 곱해지는 S_{f_i}값을 구분할 수 있도록 함으로써 단순 전력 분석을 수행하고자 한다.

제안 공격 방법의 설명을 위해 모듈라 N이 1024비트이고 프로세서의 처리 단위를 16비트라고 가정하면 N은 모두 64개의 디지트로 구성된다. 즉, 16비트의 단정도 정수 값이 64개의 배열에 저장되는 구조이다. 연산에 사용된 1024비트의 다른 변수 값도 이와 같은 배열 구조를 갖는다고 가정한다.

본 논문에서 단순 전력 분석 공격을 위해 한 디지트의 입력 M을 사용한다고 가정해 보자. 그리고 단계 1에서 사전 계산을 수행하게 되는데 모두 m개의 사전 계산 값을 S_{fi} 변수에 저장하게 된다. 이 경우 각 변수 값은 서로 다른 길이(디지트) 값을 가지되 Fig. 4와 같은 배열에 채워질 것이다. 그림에서 진하게 표시된 부분이 한 디지트의 M에 대해 사전 계산을 했을 경우 S_{f_i}가 차지하는 배열이다.

서로 다른 길이의 S_{fi} 값이 저장되면 반복적인 곱셈 연산을 수행하는 단계 6에서 모듈라 곱셈시 소비 전력이 차이를 보이게 된다. 특히, 모듈라 감소에서는 두 정수를 곱한 결과 값의 길이에 따라 수행 시간이 많은 차이를 나타낸다. 예를 들어 f_i 가 1인 경우 1024비트(64자리) C_0와 16비트(1자리) S_{f_1}를 곱하여 모듈라 감소를 하는 경우와 f_i가 5인 경우 1024비트(64자리) C_0와 80비트(5자리) S_{f_5}를 곱하여 모듈라 감소를 하는 경우, 모듈라 곱셈을 처리하는 시간과 전력량은 많은 차이를 보이게 된다. 결론적으로 본 논문에서 제안하는 단순 전력 분석 공격은 단계 6에서 f_i값에 따라 수행 시간 및 소비 전력을 구별할 수 있도록 효과적으로 입력 메시지의 크기를 조절하는 것이다.

그러나 한 디지트의 M 값을 사용하여 단순 전력 분석 공격을 시도했음에도 소비 전력량이나 수행 시간을 구별하기 어려울 수도 있다. 이 경우 M의 크기를 조절할 수도 있다. 즉, 공격자는 사전에 계산된 값들과 임의의 정수 값을 곱하는 과정에서 발생하는 전력 소비량과 시간 차이가 많이 나도록 M을 조절할 수 있다.

위의 예에서 M을 두 디지트 수를 사용한다면 각 S_{fi} 값은 최대 2디지트의 차이를 보일 것이므로 단순 전력 공격이 더욱 용이할 것이다. 따라서 이 공격에서

배열 / 사전 계산 값	31	30	⋯	16	15	14	…	5	4	3	2	1	0
$S_1 = M$													
$S_2 = M^2 \bmod N$													
$S_3 = M^3 \bmod N$													
$S_4 = M^4 \bmod N$													
$S_5 = M^5 \bmod N$													
:								:	:	:	:	:	:
$S_{15} = M^{15} \bmod N$													

Fig.4. Occupied digits of each variable after precomputation processing

곱셈 연산을 구별하기 위한 효과적인 M의 디지트 수는 s를 사전 계산이 필요한 메모리 수 $m = 2^r$으로 나눈 $\lfloor s/m \rfloor$것이 될 것이다. 예를 들어 위의 m-ary 멱승에서 N이 64개의 디지트를 사용하고 r이 4인 경우, $\lfloor S/m \rfloor = 64/2^4 = 4$ 자릿수의 M을 입력하는 것이 가장 효과적인 방법이다.

본 논문에서 제시한 단순 전력 분석 공격을 단계별로 정리하면 다음과 같다.

〔공격 단계 1〕 : 입력 메시지 M의 자릿수(디지트)를 결정한다. 즉, m-ary 멱승 방식에서 $\lfloor s/m \rfloor$을 계산하여 이용할 수 있고 반복적인 실험을 통해 효과적인 자릿수를 결정할 수도 있다.

〔공격 단계 2〕 : 임의의 값과 사전 계산된 값과의 곱셈 연산 시간과 전력 소비량을 측정한다. 여기서는 여러 번의 반복 시행을 통해 평균값을 구하여 사용할 수 있다.

〔공격 단계 3〕 : 단계 1에서 구한 입력 값을 이용하여 실제 m-ary 멱승 알고리즘을 수행하되 전체 전력 소비 파형을 수집한다.

〔공격 단계 4〕 : 수집된 전력 파형에서 자승 연산과 곱셈 연산을 구별해 내고 Fig. 3의 단계 6의 연산 과정을 관측하여 m진 비밀 키의 각 계수를 추출해 낸다.

〔공격 단계 5〕 : 지수의 첫 번째 디지트 f_{s-1}는 전수 조사를 통해 쉽게 얻을 수 있다.

IV. 공격 실험 결과 및 대응책

4.1 단순 전력 분석 공격 실험

본 장에서는 m-ary 멱승 알고리듬이 단순 전력 공격에 취약함을 실험을 통해 확인하고자 한다. 이를 위해 실험에서는 32비트 ARM920T 프로세서를 장착한 KLA-SCARF 평가 보드를 사용하였다〔19〕. 이 보드의 동작 주파수는 50MHZ이며 오실로스코프는 Lecroy WaveRunner 104Xi-A를 사용하였다. 논문에서는 N과 d가 모두 1024비트로 하는 m-ary 멱승 알고리듬을 개발하여 보드에 구현한 후 공격 실험을 수행하였다. Fig. 5는 실험용 보드와 오실로스코프의 설치 환경을 나타낸 것이다.

또한, 1024비트를 16비트씩 64디지트로 배열화하였으며 먼저 r을 4로 했을 경우를 가정하고 실험하였다. 이 경우 사전 계산되는 변수의 개수는 16개이다. 따라서 메시지 M은 4디지트 값을 사용하였다. 물론, 이론적으로 볼 때 1024비트 m-ary에서는 r이 5인 경우가 속도 면에서의 최적 조건이다〔4〕. 그러나 이 경우는 소요되는 메모리가 많은 단점이 있어 현실적인 구현 가능성은 r을 4로 했을 경우보다 낮을 수 있다. 먼저 3장의 〔공격 단계 2〕를 통해 임의의 값과 사전 계산된 값과의 곱셈 연산 시간을 측정하였으며 그 결과는 Fig. 6과 같다. 여기서 S_0값은 실제 출력과는 전혀 관계없는 더미(dummy) 연산용 값으로 한 자리

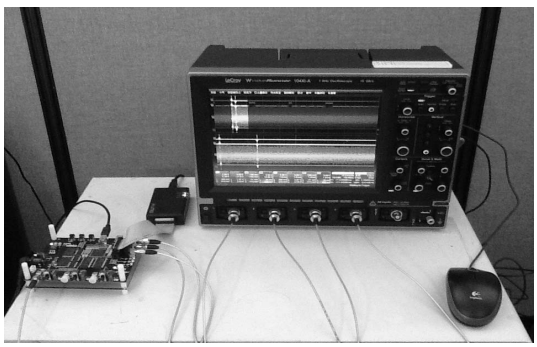


Fig.5. Experimental environment

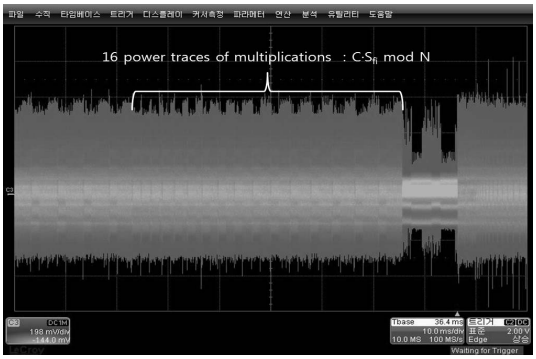


Fig.6. Power trace of modular multiplications $C \cdot S_{f_i} \bmod N$

값을 사용하였다.

그림에서 보는 바와 같이 사전 계산 값의 자릿수 증가에 따라서 전력 파형도에 나타난 소비 시간도 선형적으로 증가함을 알 수 있었다. 이 모듈라 곱셈 과정을 파형으로 측정하여 분석한 것을 수치화하여 나타낸 것이 Table 1이다. 표에서 보는 바와 같이 1024비트인 두 수의 모듈라 곱셈은 약 4.7ms 정도의 시간이 걸렸으며 피승수의 디지트 수가 작은 경우에는 그 보다 작은 시간이 소요되었다.

논문에서는 m-ary 멱승 알고리듬을 구현하여 직접 단순 차분 공격을 시도해 보았다. 구현한 알고리듬을 수행한 전력 소비 파형은 Fig. 7과 같다. 오실로스코프를 통해 분석한 결과 자승 연산과 곱셈 연산은 충분히 육안으로 구별이 가능하였다. Fig. 8은 단순 차분 공격을 시도하기 위해 Fig. 7의 일부분을 확대하여 나타낸 것이다. 그림에서 보는 바와 같이 4번의 자승 연산 $C^{2^r} \bmod N$의 파형과 이어지는 한 번의 곱셈 연산 과정을 볼 수 있다. 따라서 이 곱셈 시간과 Table 1과의 유사성을 검사하여 f_i 값들을 쉽게 추측할 수 있었다.

단순 전력 분석 실험은 r을 5로 구현한 환경에서도 시도하였다. 이 경우에는 입력 메시지를 2디지트로 설정한 후 공격을 시도하였다. 실험 결과, 파형의 잡음으로 인해 다소간의 오차는 있었지만 좀더 세밀한 분석과 수차례 실험을 반복하면 비밀 키를 찾아내는 데 큰 문제는 없었다.

Table 1. Computational time of $C \cdot S_{f_i} \bmod N$.

S_{f_i}(디지트)	$C \cdot S_{f_i} \bmod N$ 연산 시간(ms)
S_0(1)	1.728
S_1(4)	1.945
S_2(8)	2.147
S_3(12)	2.385
S_4(16)	2.612
S_5(20)	2.799
S_6(24)	3.053
S_7(28)	3.151
S_8(32)	3.498
S_9(36)	3.544
S_{10}(40)	3.796
S_{11}(44)	3.994
S_{12}(48)	4.176
S_{13}(52)	4.372
S_{14}(56)	4.601
S_{15}(60)	4.737

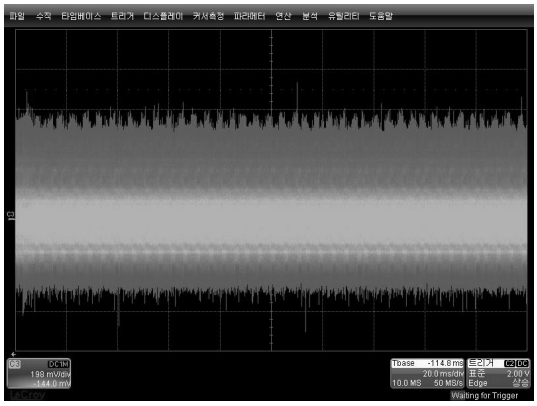


Fig.7. Computational power trace of m-ary exponentiation algorithm

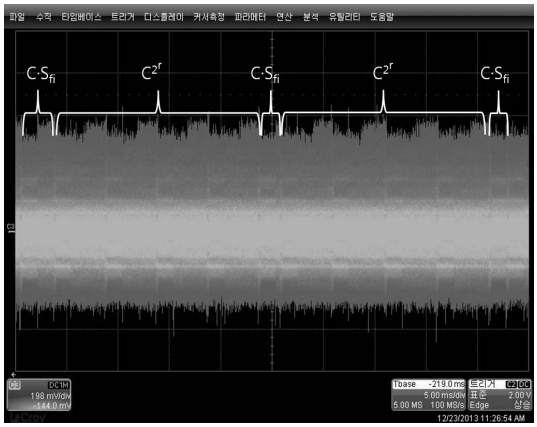


Fig.8. Detailed power trace for simple power analysis

4.2 새로운 단순 전력 공격에 대한 대응 방법

지금까지의 단순 전력 분석 공격은 멱승 연산시 발생하는 소비 전력을 통해 자승과 곱셈 연산을 구별함으로써 비밀 키를 추출하는 방법으로서 주로 이진 멱승 방식에 적용되었다. 그러나 본 논문에서 제시한 공격 방법은 사전 계산을 필요로 하는 m-ary나 윈도우 멱승 알고리듬에 적용 가능한 공격 기법으로서 사전 계산된 값의 사용 여부를 단순 전력 분석 기법을 통해 판단함으로써 비밀 키를 추출하는 방식이다.

따라서 제안하는 단순 전력 분석 공격을 방어하기 위해서는 입력 메시지에 대한 검사나 표준화된 입력 형식을 점검하는 것이 중요하다. 예로서 RSA 서명시에는 PKCS#1 v2.2에서 제공하는 입력 형식을 확인하여야 한다[20, 21]. 하지만 입력을 암호문으로 하는 복호화 과정에서는 그 형식을 확인하는 형식이 없으므로 주의를 해야 하며 최소한 입력 암호문이 모듈러스와 비슷한 크기의 값을 가지는 지는 확인해야 한다. 그러나 단순히 입력문의 크기만을 검사하는 방법은 $M=(N-M')$ 값을 사용하는 공격에 일부 취약할 수 있다. 여기서 M'은 단순전력 분석 공격에 용이한 작은 디지트를 가지는 정수이다.$(N-M')$ 값을 이용하는 단순 차분 공격에서 작은 크기의 M'값을 사용하면 M 값은 큰 정수처럼 보이지만 짝수 배를 곱한 값은 아래와 같이 작은 자리의 정수가 되므로 결국 짝수 계수를 갖는 비밀 키 정보는 단순 전력 분석에 의해 노출되게 된다.

$$M^2=(N-M')^2=(M')^2 \bmod N$$

$$M^4=(M')^4 \bmod N \quad \ldots.$$

제안 공격을 방어하는 두 번째 방법으로 입력 메시지를 블라인딩하는 것을 고려해 볼 수 있다[8]. 일반적으로 메시지 블라인딩 기법은 다음과 같은 방법을 이용한다. 여기서 r는 랜덤 수를 나타낸다.

방법 1. 지수를 랜덤화하는 방법

- $d^*=d+r\cdot\phi(N)$
- $C=M^{d^*} \bmod N$

방법 2. 모듈러스 N을 랜덤화하는 방법

- $N^*=N\cdot r$
- $C=(M^d \bmod N^*) \bmod N$

방법 3. 입력 메시지를 랜덤화하는 방법

- $M^*=M\cdot r^e \bmod N$
- $C^*=(M^*)^d \bmod N$
- $C=C^*\cdot r^{-1} \bmod N$

그러나 블라인딩 방법 중에서 방법 1은 제안하는 단순 전력 분석 공격에 의해 d^*을 노출할 수 있는데 이는 비밀 키 d와 같은 지수 성질을 가지고 있기 때문에 적절한 대응책이 될 수 없다. 또한, 두 번째 블라인딩 방법은 멱승에 사용된 지수와 무관한 랜덤화 기법이므로 제안하는 단순 전력 분석 공격을 막을 수는 없다. 마지막으로 3번째 방법은 제안하는 단순 전력 분석 공격을 방어하는 하나의 블라인딩 기법이 된다. 하지만 이 대응 기법은 $r^e \bmod N$ 값과 $r^{-1} \bmod N$ 값을 사전에 구하여 미리 저장해야 하는 단점을 가지고 있다.

V. 결 론

공개 키 암호 시스템을 암호용 디바이스에 구현하여 수행할 경우 고속의 멱승 알고리듬이 필수적으로 사용된다. 그러나 최근 멱승 알고리듬을 수행할 경우 발생하는 소비 전력 신호를 분석하여 비밀 키를 찾아내는 전력 분석 공격이 주목을 받고 있다. 특히, 멱승 알고리듬 중 m-ary나 윈도우 방법들은 그 동안 사전 계산 정보를 저장해야 하는 메모리의 부담으로 인해 하드웨어 구현 시 주목을 받지 못했지만 어느 정도 구현 환경이 제공되는 디바이스에서는 고속 연산을 목적으로 사용되고 있다.

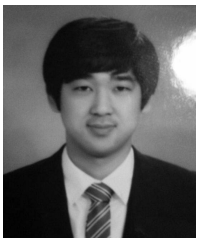
본 논문에서는 m-ary 멱승 알고리듬에 대한 새로운 단순 전력 분석 방법을 제안하였다. 제안하는 공격 방식에서는 입력 메시지를 조절하여 사전 계산 값을 곱하는 연산에서 피승수의 값을 전력 파형 분석을 통해 예측할 수 있도록 한 것이다. 제안하는 공격 방법이 타당함을 검증하기 위해 실제 평가용 보드에 관련 알고리듬을 구현한 후 단순 전력 분석 공격을 실시하였다. 실험 결과 전력 파형을 구별할 수 있도록 충분히 작은 입력을 사용하면 비밀 키를 추출해 낼 수 있음을 확인하였다. 제안 방식은 사전 연산을 수행하는 윈도우 멱승 알고리듬에도 확대 적용이 가능하므로 이에 대한 대책이 강구되어야 하며 개발자는 암호 알고리듬을 구현 시 세심한 주의를 기울여야 할 것이다.

References

[1] R. Rivest, A Shamir, and L, Adelman, "A method for obtaining digital signature and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, vol. 31, no.4, pp. 469-472, July 1985.
[3] D. E. Knuth, The art of computer programming : Seminumerical algorithms, Addison-Wesley, Reading, MA, 1981.
[4] C. K. Koc, "Analysis of sliding window techniques for exponentiation," Computers and Mathematics with Application, vol. 30, no. 10, pp. 17-24, Nov. 1995.
[5] P. Kocher, "Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and other systems," CRYPTO'96, LNCS, vol. 1109, pp. 104-113, Aug. 1996.
[6] P. Kocher, J. Jae, and B. Jun, "Differential power analysis," CRYPTO'99, LNCS, vol. 1666, pp. 388-397, Aug. 1999.
[7] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks on modular exponentiation in smartcards," CHES'99, LNCS vol 1717, pp. 144-157, Aug. 1999.
[8] S. Mangard, E. Oswald, and T. Popp, "Power analysis attacks - Revealing the secrets of smart cards," Springer-Verlag, 2010.
[9] J. S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES'99, LNCS, vol. 1717, pp. 292-302, Aug. 1999.
[10] D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of checking cryptographic protocols for faults," EUROCRYPTO'97, LNCS, vol. 1233, pp. 37-51, May 1997.
[11] C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J. P. Seifert, "Fault attacks on RSA with CRT: Concrete results and practical countermeasures." CHES'02, LNCS, vol. 2523, pp. 260-275, Sep. 2003
[12] A. Boscher, R. Naciri, and E. Prouff, "CRT-RSA algorithm protected against fault attacks," WISTP'07, LNCS vol. 4462, pp. 237-252, May 2007.
[13] F. Amiel, K. Villegas, B. Feix, and L. Mercel, "Passive and active combined attacks: Combining fault attacks and side channel analysis," Proceedings of the 4th Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC'07), pp. 92-102, Sep. 2007.
[14] H. D. Kim and J. C. Ha, "A physical combined attack and its countermeasure on BNP exponentiation algorithm," Journal of the Korea Institute of Information Security, vol. 23, no. 4, pp. 585-591, Aug. 2013.
[15] S. M. Yen and M Joye, "Checking before output may not be enough against fault-based cryptanalysis," IEEE Transactions on Computers, vol. 49, no. 9, pp. 967-970, Sep. 2000.
[16] M. Joye and S. M. Yen, "The Montgomery powering ladder," CHES'02, LNCS, vol. 2523, pp. 291-302, Aug. 2002.
[17] S. Yen, L. Ko, S. Moon, and J. Ha, Relative doubling attack against Montgomery ladder, ICISC'05, LNCS, vol. 3935, pp. 117-128, Dec. 2005.
[18] S. K. Jung, Y. J. Choi, D. H. Choi and J. C. Ha, "Vulnerability of exponentiation algorithm for smartphone by simple power analysis Attack," Proceedings of Chungcheong Regional Conference on Korea Institute of Information Security, vol. 17, pp. 16-22, Sep. 2013
[19] ETRI and ICTK, SCARF evaluation board SCARF-ARM, Available at http://www.k-scarf.or.kr
[20] RSA Laboratories, "PKCS #1 v2.2 : RSA Cryptography standard," October, 2012,
[21] S. M. Yen, W. C. Lien, S. J. Moon, and J. C. Ha, "Power analysis by exploiting chosen message and internal collisions - Vulnerability of checking mechanism for RSA-decryption," Mycrypt'05, LNCS, vol. 3715, pp 183-195, Sep. 2005.

〈저자소개〉



안 성 준 (Sung-jun Ahn) 학생회원
2012년 8월: 가톨릭대학교 수학과 졸업
2012년 9월~현재: 과학기술연합대학원대학교 정보보호공학과 석사
〈관심분야〉 정보보호, 암호 엔지니어링, 부채널 분석

최 두 호 (Doo-ho Choi) 정회원
1994년 2월: 성균관대학교 수학과 졸업
1996년 2월: KAIST 수학과 석사
2002년 2월: KAIST 수학과 박사
2002년 1월~현재: 한국전자통신연구원 책임연구원
〈관심분야〉 암호 엔지니어링, 부채널 분석, IoT 보안

하 재 철 (Jae-Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 부교수
2007년 3월~현재: 호서대학교 정보보호학과 교수
〈관심분야〉 정보보호, 네트워크 보안, 부채널 공격