情報保護學會論文誌
第19卷 第6號, 2009. 12

RSA 멱승 알고리즘의 제어문에 대한 오류 주입 공격

길 광 은,† 백 이 루, 김 환 구, 하 재 철‡
호서대학교

Fault Analysis Attacks on Control Statement of RSA Exponentiation Algorithm

Kwang-Eun Gil,† Yi-Roo Baek, Hwan-Koo Kim, Jae-Cheol Ha‡
Hoseo University

요 약

최근의 연구는 RSA와 같은 암호 시스템에서 멱승 알고리즘을 구현할 경우 물리적 공격에 취약하여 비밀 키를 노출할 수 있음을 보이고 있다. 특히, Schmidt와 Hurbst는 RSA 이진 멱승(binary exponentiation) 실행시 수행하는 제곱(squaring) 연산을 건너뛰게 하여 얻은 오류 서명값을 이용하여 비밀 키를 얻을 수 있음을 실험적으로 보였다. 본 논문에서는 Schmidt와 Hurbst의 공격 가정에 기반하여 곱셈(multiplication) 연산이나 반복 제어문 연산을 건너뛰어 비밀 키를 공격하는 방법을 제안한다. 또한, 반복 제어문을 건너뛰는 오류 공격을 확장하여 단순 전력 분석 공격(simple power analysis)공격에 대응하기 위해 제안된 몽고메리(Montgomery ladder) 멱승 알고리즘도 공격할 수 있음을 보인다.

ABSTRACT

Many research results show that RSA system mounted using conventional binary exponentiation algorithm is vulnerable to some physical attacks. Recently, Schmidt and Hurbst demonstrated experimentally that an attacker can exploit secret key using faulty signatures which are obtained by skipping the squaring operations. Based on similar assumption of Schmidt and Hurbst's fault attack, we proposed new fault analysis attacks which can be made by skipping the multiplication operations or computations in looping control statement. Furthermore, we applied our attack to Montgomery ladder exponentiation algorithm which was proposed to defeat simple power attack. As a result, our fault attack can extract secret key used in Montgomery ladder exponentiation.

Keywords: RSA, Exponentiation algorithm, Fault analysis attack, Montgomery ladder algorithm, Control statement

I. 서 론

스마트 카드와 같은 임베디드 시스템을 이용하여 인증, 암호, 디지털 서명 등 보안 서비스를 제공하기 위해서는 다양한 암호 알고리즘을 이용한다. 현재 상용으로 사용되는 암호 알고리즘 중 대표적인 것이 RSA 암호 시스템이다[1]. 그러나 이러한 암호 시스템은 최근 오류 주입 공격(fault analysis attack, FA)과 같은 물리적 공격(physical attack)에 의해 비밀 정보가 노출될 수 있음이 밝혀졌다[2-8].

오류 주입 공격은 1997년에 Boneh와 Biham에 의해 처음 제안되었으며 암호 시스템이 구동되는 중간에 오류를 주입하여 나온 출력을 분석하여 사용된 암호 알고리즘의 비밀 키를 얻는 공격이다[2,3]. 지금까지 발표된 오류 주입 공격과 대응책들은 주로 RSA와 같은 공개키 암호 시스템에 많이 적용되었으며, 특히 중국인의 나머지 정리를 적용한 RSA-CRT(RSA with Chinese Remainder Theorem) 알고리즘이 오류 공격에 취약한 것으로 알려져 있다. 그 이유는 RSA-CRT 멱승 연산을 수행할 경우 오류를 넣을

접수일(2009년 9월 1일), 게재확정일(2009년 11월 5일)
† 주저자, kke0805@nate.com
‡ 교신저자, jcha@hoseo.edu

수 있는 시간적 여유가 충분하며 한 번의 오류 주입 공격으로 비밀 키 전체를 찾아낼 수 있기 때문이다. 이와 같이 RSA-CRT 알고리즘 이외에도 일반적인 RSA 멱승상의 오류 공격으로는 safe-error 공격이 제안되기도 하였다[6]. 하지만 safe-error 공격은 메모리나 레지스터에 정밀한 오류 주입이 가능하거나 모듈러 곱셈(modular multiplication) 시점에 순간적인 오류를 주입해야 하므로 공격의 난이도가 높고 비현실적인 가정을 많이 포함하고 있다.

한편, 2007년 Kim과 Quisquater는 알고리즘 실행동안 글리치나 스파크를 일으켜 if문과 같은 특정한 연산을 건너뛰는(skip) 오류 공격을 실현하였다[7]. 최근에는 Schmidt와 Hurbst가 RSA 멱승 연산시 제곱 연산만 건너뛰는 오류를 주입하여 비밀 키를 공격할 수 있는 오류 주입 공격을 실험적으로 성공한 바 있다[8].

본 논문은 Schmidt와 Hurbst가 소개한 것과 동일한 오류 주입 공격 가정하에서 곱셈 연산을 건너뛰는 방법을 이용한 보다 효율적인 오류 주입 공격 방법을 제안한다. 또한, if문과 같은 조건문뿐만 아니라 for문과 같은 반복문을 건너뛸 수 있다면 쉽게 비밀 키를 찾아낼 수 있음을 보인다. 이러한 반복 제어문을 건너뛰는 오류 공격을 확장하면 이진 멱승(binary exponentia- tion) 알고리즘뿐만 아니라 단순 전력분석 공격(simple power analysis)에 대응하기 위해 제안된 몽고메리(Montgomery ladder) 멱승 알고리즘[9]도 공격될 수 있음을 보인다.

II. RSA 멱승 알고리즘과 오류 주입 공격

2007년 Kim과 Quisquater가 제안한 논문에 따르면 공격 대상이 되는 임베디드 암호 칩에 글리치나 전압 스파크를 인가하면 오류가 주입되어 특정 연산을 건너뛸 수 있음을 실험적으로 보였다[7]. 여기에서 저자들은 오류 서명 검사를 위한 if문을 건너뛰는 실험을 통해 RSA-CRT 연산에서의 비밀 키를 찾아내는 오류 주입 공격을 제안하였다. 이후, 글리치나 전압 스파크를 이용하여 두 번의 오류를 주입하는 2차 오류 공격[10]이나 반복 제어문의 카운터 값에 오류를 주입하는 방법으로 암호 시스템의 비밀 키를 찾아내는 연구가 진행되고 있다[11].

최근 Schmidt와 Hurbst는 Kim과 Quisquter가 제안한 오류 주입 방법을 확장한 새로운 오류 공격을 제시하였다. 그들이 제안한 오류 공격은 일반적인 RSA 암호 시스템에서 사용하는 이진 멱승 과정에 오류를 주입하여 비밀 키를 찾아내는 공격으로서, AVR 마이크로 컨트롤러에 Left-to-Right 이진 멱승 알고리즘을 구현하여 실험적으로 검증하였다. 또한 저자들은 그 공격을 Right-to-Left 이진 멱승 방법이나 m-ary 멱승 알고리즘, sliding window 멱승 방법에도 적용할 수 있다고 주장하였다[8].

[그림 1]은 Left-to-Right 이진 멱승 알고리즘을 나타내고 있으며 메시지 m과 비밀 키 $d=(d_{t-1},\ldots,d_1,d_0)_2$를 이용하여 서명 값 S를 계산하는 과정이다. 여기서 비밀 키 d가 공격자가 찾고자 하는 공격 목표가 된다.

입력 : $m,\ N,\ d=(d_{t-1},\ldots,d_1,d_0)_2$
출력 : $S=m^d \bmod N$

1. $S \leftarrow 1$
2. for $(i=t-1; i>=0; i--)$ {
2.1 $S \leftarrow S \cdot S \bmod N$
2.2 if $(d_i=1)$
$S \leftarrow S \cdot m \bmod N$ }
3. $return\ S$

[그림 1] Left-to-right 이진 멱승 알고리즘

Schmidt와 Hurbst가 제시한 오류 공격의 핵심은 [그림 1]의 알고리즘 실행 중에 단계 2.1과 같은 제곱 연산에 오류를 주입하여 오류 결과를 얻는 것이다. 이 공격의 첫 번째 과정은 메시지 m에 대한 정상 서명 S를 구하는 것이다. 그 후 같은 메시지를 가지고 서명을 수행하는데 단계 2의 마지막 루프(loop), 즉, $i=0$일 때의 제곱 연산을 건너뛰어 나온 결과를 얻어낸다. 그리고 이를 정상 서명과 비교함으로써 아래와 같이 비밀 키의 최하위 비트인 d_0를 알아낸다.

$$\begin{bmatrix} S=(S_0)^2 \bmod N & \Rightarrow d_0=0 \\ S=(S_0)^2 \cdot m^{-1} \bmod N & \Rightarrow d_0=1 \end{bmatrix} \quad (1)$$

여기서 S_0는 반복문의 인덱스 i가 0일 때 오류를 주입하여 제곱 연산을 건너뛴 오류 서명을 의미한다. 즉, 식 (1)과 같이 정상 서명 S와 맨 마지막 제곱 연산을 건너뛴 오류 서명 S_0의 관계를 이용하여 마지막 비트 d_0를 결정한다.

최하위 비밀 키 비트를 결정하였으면 그 다음 비밀 키 비트 d_1을 결정하기 위해 이전에 구한 비트 S_0(d_0의 제곱 연산 과정에 오류를 주입하여 얻은 서명)와 $i=1$일 때 오류를 주입하여 제곱 연산을 건너뛴 서명 S_1을

비교하여 d_1을 결정하게 된다. 이 과정을 일반화된 식으로 표현하면 아래와 같다.

$$\begin{bmatrix} S_k = S_{k-1} & \Rightarrow d_k = 0 \\ S_k = m^{2^{k-1}} \cdot S_{k-1} & \Rightarrow d_k = 1 \end{bmatrix} \quad (2)$$

즉, [그림 1]에서 인덱스 i가 $k-1$일 때 제곱 연산 오류를 주입하여 얻은 오류 서명 S_{k-1}과 인덱스 i가 k일 때 오류 서명 S_k로부터 비밀 키 비트 d_k를 결정할 수 있다. 위 과정을 반복적으로 적용하여 그 다음 비밀 키 비트를 순차적으로 찾을 수 있어 결국 모든 비밀 키 비트를 구할 수 있게 된다.

III. 제어문에 대한 오류 주입 공격 제안

3.1 오류 주입 공격에 대한 가정

Schmidt와 Hurbst가 제안한 공격의 오류 주입 지점은 이진 멱승 알고리즘에서 반복문(for문) 안의 제곱 연산에 두고 있다. 즉, 저자들은 암호 칩 내부에 저장된 소프트웨어가 수행될 때 반복문내의 제곱 연산을 하는 서브루틴(subroutine)을 건너뛰는 것이 가능하며 이를 통해 비밀 키를 찾아 낼 수 있음을 실험적으로 검증하였다. 여기에서는 편의상 이 공격을 서브루틴 건너뛰기 오류 공격이라 부르기로 하자.

암호 프로그램을 수행하는 도중에 특정한 명령어를 건너뛰는 공격은 Kim과 Quisquter의 공격 실험에서도 증명이 되었는데 여기에서는 if문을 건너뛰어 RSA-CRT 연산에서의 비밀 키를 찾아내는 방법이었다. 이를 근거로 본 논문에서 세울 수 있는 첫 번째 가정은 현재의 오류 주입 기술을 이용하면 특정 프로그램의 조건문인 if문을 건너뛸 수 있다는 것이다.

본 논문에서의 두 번째 가정은 반복 제어문인 for문을 수행하는 경우 오류 주입이 가능하다고 가정하였다. 이 가정은 if문 건너뛰기나 서브루틴 건너뛰기와 달리 실험적으로 검증된 사례는 없다. 그러나 논문 [11]에서는 반복 제어문인 for문의 인덱스 값에 오류를 주입하여 비밀 키를 공격하는 방법을 이론적으로 제안한 바 있다.

Schmidt와 Hurbst의 공격에서는 전체 비밀 키를 찾기 위해 RSA 멱승 연산을 마지막 루프에서 처음 루프까지 비밀 키를 탐색하면서 공격자가 원하는 시점에 제곱 연산을 건너뛰는 오류를 주입해야 한다. 즉, 하위 비밀 키 비트에 해당하는 루프 연산 동안 제곱 서브루틴 건너뛰기 오류를 발생하여 구한 S_{k-1}를 저장하고 있어야 S_k와의 비교를 통해 d_k를 구하게 된다. 따라서 이 공격이 성공하기 위해서는 정밀한 오류 주입 시점을 결정해야 하며 이전 오류 서명 값을 모두 저장하고 있어야 하는 단점이 있다.

3.2 조건문(if문) 오류를 이용한 공격

본 논문에서는 Schmidt와 Hurbst가 제안한 공격보다 쉽게 비밀 키를 찾는 오류 주입 공격을 제안하고자 한다. 여기서는 [그림 1]의 이진 멱승 알고리즘을 사용함을 전제로 알고리듬 수행 중간에 일정 동작을 수행하지 못하게 함으로써 조건문(if문)을 건너뛸 수 있다고 가정하였다. 즉, [그림 1]에서 단계 2.2의 if문 자체를 건너뛰게 함으로써 조건에 해당하는 서브루틴, 즉 곱셈 연산을 수행하지 않음을 가정하였다. 이러한 공격 모델은 Kim과 Quisquter의 공격[7]에서 세운 가정과 동일하다.

먼저 공격자는 메시지 m에 대한 정상 서명 S를 가지고 있다고 가정한다. 본 논문에서 제안한 공격은 [그림 1]의 멱승 알고리즘이 수행될 때 단계 2.2의 if문을 건너뛰는 오류를 주입하는 것이다. 공격자는 각 루프마다 오류를 주입하여 나온 오류 서명 값을 가질 수 있으며, 이러한 과정으로 나온 오류 서명을 정상 서명 값과 비교하는 과정을 진행할 수 있다.

제안 공격의 핵심은 [그림 1]의 단계 2.2의 곱셈 서브루틴 건너뛰기 오류가 주입될 경우 출력한 오류 서명 값과 정상 서명 값의 차이를 비교하는 것이다. 만약 단계 2.2인 if문에 오류가 들어가 곱셈 과정을 건너뛰게 되면 다음과 같은 수식으로 그 관계를 정리할 수 있다.

$$\begin{bmatrix} S = S_k \bmod N & \Rightarrow d_k = 0 \\ S = S_k \cdot m^{2^k} \bmod N & \Rightarrow d_k = 1 \end{bmatrix} \quad (3)$$

여기서 S_k는 비밀 키의 k번째 비트인 d_k를 찾기 위해 반복문의 인덱스 i가 k일 때 곱셈 서브루틴 건너뛰기 오류를 통해 얻은 오류 서명이다. 따라서 식 (3)과 같이 비밀 키 값 d_k가 1일 경우, if문을 건너뛰게 되면 오류 서명은 정상 서명과 항상 m^{2^k}배의 차이를 가지게 되고 d_k가 0일 경우, if문을 건너뛰게 되면 오류 서명은 정상 서명과 같아야 한다. 이와 같은 방법으로 if문을 건너뛰게 되면 모든 비밀 키를 간단히 찾을 수 있다. 이 방법에서는 이전 하위 비트 오류시 생성된 오류 서명을 저

장할 필요가 없으며 정상 서명 S만 저장하면 된다. 이 과정을 간단한 예를 통해 보이면 [표 1]과 같다. 표에서는 비밀 키를 1001_2로 가정하였으며 S는 제곱 연산을 M은 곱셈 연산을 의미한다.

[표 1] if문에 대한 오류 공격의 예

비밀키	1		0	0	1		서명	공격 결과
과정	S	M	S	S	S	M		
정상	1	m	m^2	m^4	m^8	m^9	m^9	
오류 주입	1	m	m^2	m^4	m^8	×	m^8	1
	1	m	m^2	m^4	m^8	m^9	m^9	0
	1	m	m^2	m^4	m^8	m^9	m^9	0
	1	×	1	1	m	m^2	m^2	1

메시지 m에 대한 서명은 $S = m^d \bmod N$ 이므로 최종 결과는 m^9으로 나타낼 수 있다. 여기서 공개 모듈러 값 N은 표기를 생략한다. [표 1]에서 각 연산에 오류를 주입하여 건너 뛴 부분은 ×표시로 나타내었고, 오른쪽은 해당 연산을 건너뛴 오류 결과 값이다. [그림 1]에서 최하위 루프가 수행될 때 if문을 건너뛰게 되면 결과 값은 m^8이 된다. 그러므로 최하위 비트는 1이 됨을 알 수 있다.

3.3 반복문(for문) 오류를 이용한 공격

[그림 1]의 알고리즘에서는 인덱스 i를 사용하여 $\text{for}\,(i = t-1; i >= 0; i--)$과 같은 for문을 수행하게 된다. 이때 이 인덱스 값에 오류를 주입할 수 있다고 가정하자. 이와 같은 가정은 글리치 또는 스파크와 같은 오류 주입을 통해 반복되는 한 번의 루프 자체를 건너뛸 수 있는데, 논문 [11]에서 Mrabet는 인덱스를 저장하는 프로그램 카운터 값을 변경하여 공격자가 원하는 루프를 건너뛸 수 있다고 가정하였다.

본 논문에서는 for문에 오류를 주입하는 방법을 2가지로 가정해 보았다. 하나는 프로그램의 인덱스 값이 오류에 의해 변하여 0보다 더 작게 되어 연산 중간에 for문을 완전히 벗어나는 가정이다. 다른 하나는 for문 내부의 서브루틴을 건너뛰게 함으로써 한 번의 루프 내부 연산을 건너뛰는 것이다. 이러한 가정을 현실적으로 비교해 보면 첫 번째 방법이 두 번째 방법보다는 훨씬 쉽게 실현할 수 있다.

1) 중간에 for문을 벗어나는 오류 공격

제안하는 오류 주입 공격은 멱승 실행 시 공격자가 원하는 시점까지 계산을 한 후, 프로그램의 for문과 같은 반복하는 루틴을 종료시켜 오류 서명을 출력하게 할 수 있을 때 가능하다. 이 경우 카운터 변수 i값에 오류를 주입하여 음의 값으로 만들면 루프를 종료하게 됨으로써 가능하다. 먼저 공격자는 정상 서명 S를 가지고 있으며 [그림 1]의 단계 2에서 $i=0$일 때 반복문을 종료한 오류 서명 S_0를 얻는다. 이처럼 마지막 비트는 오류 서명과 정상 서명을 식 (4)에 적용하여 비밀 비트를 결정할 수 있게 된다.

$$\begin{bmatrix} S = S_0^2 \bmod N & \Rightarrow d_0 = 0 \\ S = S_0^2 \cdot m \bmod N & \Rightarrow d_0 = 1 \end{bmatrix} \quad (4)$$

두 번째 과정으로 공격자는 $i=1$일 때 반복문을 종료한 후 오류 서명을 얻는다. 여기서 이전에 얻은 $i=0$일 때 반복문을 종료한 오류 서명과 $i=1$일 때 종료하여 얻은 오류 서명을 비교한다.

$$\begin{bmatrix} S_k = (S_{k+1})^2 \bmod N & \Rightarrow d_{k+1} = 0 \\ S_k = (S_{k+1})^2 \cdot m \bmod N & \Rightarrow d_{k+1} = 1 \end{bmatrix} \quad (5)$$

따라서 식 (5)과 같이 이전의 루프를 종료 후 얻은 오류 서명과 현재 얻은 오류 서명의 관계를 통해 비트를 결정할 수 있다. 반복 제어문인 for문을 종료하는 오류 공격의 예를 나타낸 것이 [표 2]이다. 표에서 비밀 키 d의 최상위 비트는 1이라 가정하였다.

[표 2] for문을 종료하는 오류 공격의 예

비밀키	1		0	0	1		서명	공격 결과
과정	S	M	S	S	S	M		
정상	1	m	m^2	m^4	m^8	m^9	m^9	
오류 주입	1	m	m^2	m^4	×	×	m^4	1
	1	m	m^2	×	×	×	m^2	0
	1	m	×	×	×	×	m	0
	1	m	m^2	m^4	m^8	m^9	m^9	1

2) 한 번의 루프 연산을 건너뛰는 오류 공격

이 공격에서는 for문을 수행할 때 오류를 주입하여 for문을 종료하는 것이 아니라 해당 루프를 한 번만 건너뛴다고 가정하였다. 이러한 공격은 루프 카운터 값을 조작하는 것이 아니라 루프는 정상적으로 수행되지만 for문 내부의 서브루틴을 수행하지 않도록 오류를 주입하는 공격을 가정한 것이다. 즉, Schmidt와 Hurbst가 for문 내부의 제곱 연산만 건너뛴다는 가정을 확장하여 for문 내부 전체 서브루틴을 건너뛰는 것을 가정한 것이다.

먼저 공격자는 정상 서명 S를 가지고 있으며 마지

막 루프를 건너뛴 오류 서명 S_0를 가지고 마지막 비트를 판단할 수 있다. 이와 같은 오류 결과는 위에서 언급한 마지막 루프를 수행하지 않고 종료한 식 (4)의 결과와 같다. 이후 공격자는 그 이전 루프를 건너뛰는 오류를 주입하여 순차적으로 각 루프를 건너뛴 오류서명 S_k를 얻어낸다. 이 공격은 [그림 1]의 단계 2에서 for문의 i값이 t부터 0까지 감소하면서 반복될 때, 각각의 루프를 건너뛴 오류서명을 얻어내면 다음 식과 특별한 관계를 가지는 것을 확인할 수 있다.

$$\begin{bmatrix} S_k = S_{k+1} \bmod N \Rightarrow d_k = d_{k+1} \\ S_k \neq S_{k+1} \bmod N \Rightarrow d_k \neq d_{k+1} \end{bmatrix} \quad (6)$$

식 (6)와 같이 하위 비트를 찾아내기 위해 만든 오류 서명 S_k와 그 다음 상위 비트를 얻기 위해 루프를 건너뛴 오류 서명 S_{k+1}가 동일할 경우 그 해당 비밀 비트 값은 서로 같고, 오류 값이 다를 경우는 비밀 비트도 다른 값이 된다. 반복 제어문인 for문의 루프를 건너뛰는 오류 공격의 예를 나타낸 것이 [표 3]이다.

[표 3] for문에 대한 루프 건너뛰기 오류 공격의 예

비밀키	1		0	0	1		서명	공격 결과
과정	S	M	S	S	S	M		
정상	1	m	m^2	m^4	m^8	m^9	m^9	
오류 주입	1	m	m^2	m^4	×	×	m^4	1
	1	m	m^2	×	m^4	m^5	m^5	0
	1	m	×	m^2	m^4	m^5	m^5	0
	×	×	1	1	1	m	m	1

이와 같은 결과가 나오는 이유는 다음과 같다. 비밀키의 $k+1$비트와 k비트가 같을 경우 멱승 알고리즘에서는 서로 같은 종류의 연산(d_i가 0이면 제곱, 1이면 제곱과 곱셈)을 반복하게 된다. 따라서 같은 연산을 가진 두 개의 루프를 한번씩 건너뛴 결과는 서로 동일하다. 따라서 이러한 비트가 같은지 다른지를 결정하기 위해 마지막 루프에 오류를 주입하여 마지막 비트를 결정하면 순차적으로 각 루프에 오류를 주입하여 오류 서명 값이 같은지 다른지를 판단하여 손쉽게 비밀 키를 추출할 수 있다.

IV. Montgomery ladder 멱승 알고리즘에 대한 적용

위에서 제안한 공격은 이진 멱승 알고리즘에 사용된 비밀 키를 찾는 것이었다. 본 장에서는 이와 같은 공격 원리를 Montgomery ladder 멱승 알고리즘으로 확장해 보고자 한다. Montgomery ladder 알고리즘은 일반 이진 멱승 알고리즘보다 수행 시간이 늘어나지만 멱승을 위한 매 루프마다 곱셈과 제곱을 규칙적으로 반복하는 구조로 되어 있어 단순 전력 분석 공격에 강한 특성이 있어 많이 사용되고 있다. [그림 2]는 Montgomery ladder 멱승 알고리즘을 나타낸 것인데 단계 3에서 매번 곱셈과 제곱을 한번씩 수행한다. 메시지에 대한 최종적인 서명은 $S = a_0 = m^d \bmod N$가 된다.

입력 : $m,\ N,\ d = (d_{t-1}, \ldots, d_1, d_0)_2$
출력 : $S \leftarrow a_0 = m^d \bmod N$

1. $a_0 \leftarrow 1$
2. $a_1 \leftarrow m$
3. for $(i = t-1; i >= 0; i--)$
   3.1 $a_{\bar{d}_i} \leftarrow a_0 \cdot a_1 \bmod N$
   3.2 $a_{d_i} \leftarrow a_{d_i}{}^2 \bmod N$
4. $return\ (a_0, a_1)$

[그림 2] Montgomery ladder 멱승 알고리즘

Giraud는 이 알고리즘을 이용하면 단순 전력 분석 공격뿐만 아니라 비교 연산을 통해 오류 주입 공격에도 대응할 수 있다고 주장하고 있다[12]. 이 알고리즘의 큰 특징은 출력 값이 a_0와 a_1이며, 이 두 값의 관계는 항상 m이 곱해진 만큼의 차이를 가지고 있다. 즉, $a_1 = a_0 \cdot m \bmod N$이다. 따라서 Giraud는 멱승 연산 동안 제곱 또는 곱셈에 오류를 주입하게 되면 a_0와 a_1이 m배의 차이가 나는 관계가 깨지기 때문에 [그림 2]의 최종 출력인 두 값의 비교를 통해 서명 오류를 발견할 수 있어 오류 주입 공격을 차단할 수 있음을 보였다.

그러나 논문에서 제안하고자 하는 공격의 핵심은 이러한 a_0와 a_1의 관계식을 만족하면서 비밀 키를 찾아낼 수 있다는 것이다. Montgomery ladder 멱승 알고리즘에 대해 3장에서 제안한 공격 방법을 확장해 보자. 먼저 곱셈이나([그림 2]의 단계 3.1) 혹은 제곱 연산([그림 2]의 단계 3.2)의 서브루틴을 건너뛴다고 가정하자. 이 경우 오류 서명은 $a_1 = a_0 \cdot m \bmod N$이라는 관계식을 만족하지 못하게 되므로 Giraud의 방법처럼 최종 서명을 검사하는 단계에서 오류 서명을 출력하지 않게 된다.

그러나 본 논문에서 제안하는 오류 주입 방법은 for문에 오류를 주입하는 방법이다. 공격 방법도 위에서와 같이 1) 중간에 for문을 벗어나는 오류 공격과 2)

한 번의 루프 연산을 건너뛰는 오류 공격으로 구분해 볼 수 있다.

1) 중간에 for문을 벗어나는 오류 공격

먼저 공격자는 정상 서명 S를 가지고 있으며 [그림 2]의 단계 3에서 $i=0$일 때 반복문을 종료한 오류 서명 S_0를 얻는다. 이처럼 마지막 비트는 오류 서명과 정상 서명을 위의 식 (7)에 적용하면 비밀 비트를 결정할 수 있다.

$$\begin{bmatrix} S = S_0^2 \bmod N & \Rightarrow d_0 = 0 \\ S = S_0^2 \cdot m \bmod N & \Rightarrow d_0 = 1 \end{bmatrix} \quad (7)$$

두 번째 과정으로 공격자는 $i=1$일 때 for문을 종료한 후 오류 서명을 얻는다. 여기서 이전에 얻은 $i=0$일 때 반복문을 종료한 오류 서명과 $i=1$일 때 종료하여 얻은 오류 서명을 비교한다.

$$\begin{bmatrix} S_k = (S_{k+1})^2 \bmod N & \Rightarrow d_{k+1} = 0 \\ S_k = (S_{k+1})^2 \cdot m \bmod N & \Rightarrow d_{k+1} = 1 \end{bmatrix} \quad (8)$$

따라서 식 (8)과 같이 이전의 루프를 종료 후 얻은 오류 서명과 현재 얻은 오류 서명의 관계를 통해 비밀 키 비트를 결정할 수 있다. [표 4]는 for문을 종료하는 오류 공격의 예를 나타낸 것이다. 표에서 비밀 키 d의 최상위 비트는 1이라 가정하였다.

[표 4] for문을 종료하는 오류 공격의 예
(Montgomery ladder 멱승 알고리즘)

비밀키	1		0		0		1		서명	공격결과
과정	M	S	M	S	M	S	M	S		
변수	a_0	a_1	a_1	a_0	a_1	a_0	a_0	a_1		
정상	m	m^2	m^3	m^2	m^5	m^4	m^9	m^{10}	m^9	
오류 주입	m	m^2	m^3	m^2	m^5	m^4	×	×	m^4	1
	m	m^2	m^3	m^2	×	×	×	×	m^2	0
	m	m^2	×	×	×	×	×	×	m	0
	m	m^2	m^3	m^2	m^5	m^4	m^9	m^{10}	m^9	1

2) 한 번의 루프 연산을 건너뛰는 오류 공격

이 공격에서는 for문을 수행할 때 오류를 주입하여 for문을 종료하는 것이 아니라 해당 루프를 한번만 건너뛴다고 가정한다. 먼저 공격자는 정상 서명 S를 가지고 있으며 마지막 루프를 건너뛴 오류 서명 S_0를 가지고 마지막 비트를 판단할 수 있다. 이와 같은 오류 결과는 위에서 언급한 마지막 루프를 수행하지 않고 종료한 식 (7)의 결과와 같다. 이후 공격자는 그 이전 루프를 건너뛰는 오류를 주입하여 순차적으로 각 루프를 건너뛴 오류서명을 S_k를 구하는데 각각의 루프를 건너뛴 오류 서명은 다음 식과 같은 관계를 가지게 된다.

$$\begin{bmatrix} S_k = S_{k+1} \bmod N \Rightarrow d_k = d_{k+1} \\ S_k \neq S_{k+1} \bmod N \Rightarrow d_k \neq d_{k+1} \end{bmatrix} \quad (9)$$

식 (9)와 같이 하위 비트를 찾아내기 위해 만든 오류 서명 S_k와 그 다음 상위 비트를 얻기 위해 루프를 건너뛴 오류 서명 S_{k+1}가 동일할 경우 그 해당 비밀 비트 값은 서로 같고, 오류 값이 다를 경우는 비밀 비트도 다른 값이 된다. 반복 제어문인 for문의 루프를 건너뛰는 오류 공격의 예를 나타낸 것이 [그림]이다.

이와 같은 결과가 나오는 이유 역시 위에서 살펴본 바와 같이 인접한 $k+1$비트와 k비트의 비밀 키가 같을 경우 멱승 알고리즘에서는 서로 같은 종류의 연산을 수행하게 된다. 따라서 같은 연산을 가진 두 개의 루프 중 하나를 건너뛴 결과는 서로 동일하기 때문이다. 결론적으로 반복 제어문인 for문을 수행할 경우 오류를 주입하여 수행하는 루프를 건너뛰는 공격은 Montgomery ladder 멱승 알고리즘에 사용된 비밀 키를 찾을 수 있는 현실적인 공격 방법이 될 수 있다.

[표 5] for문에 대한 루프 건너뛰기 오류 공격의 예
(Montgomery ladder 멱승 알고리즘)

비밀키	1		0		0		1		서명	공격 결과
과정	M	S	M	S	M	S	M	S		
변수	a_0	a_1	a_1	a_0	a_1	a_0	a_0	a_1		
정상	m	m^2	m^3	m^2	m^5	m^4	m^9	m^{10}	m^9	
오류 주입	m	m^2	m^3	m^2	m^5	m^4	×	×	m^4	1
	m	m^2	m^3	m^2	×	×	m^5	m^6	m^5	0
	m	m^2	×	×	m^3	m^2	m^5	m^6	m^5	0
	×	×	m	1	m	1	m	m^2	m	

V. 결 론

본 논문에서는 Schmidt와 Hurbst가 제안한 오류 공격을 분석하고 if문이나 for문과 같은 제어문에 대한 다양한 오류 주입 기법을 제안하였다. 제안한 공격 방법은 암호를 위한 멱승 알고리즘이 수행되는 동안 조건 제어문인 if문내의 서브루틴을 건너뛰는 방법으로 비밀 키를 최하위부터 찾아내는 방법이다. 또한, 반복 제어문인 for문에 오류를 주입하여 프로그램 카운터를 변경하거나 서브루틴을 건너뛰는 방법으로 비밀 키를 찾아낼 수도 있다.

이러한 물리적 공격은 현실적인 실현 가능성을 어

느 정도 가지고 있는 지가 중요한 요소가 된다. 그럼에도 if문을 건너뛰는 공격은 Kim과 Quisquater에 의해, 제곱 연산과 같은 일정한 서브루틴을 건너뛰는 공격은 Schmidt와 Hurbst에 의해 이미 실험적으로 증명되었다. 따라서 향후에는 암호 알고리즘에서 반복문이나 조건문을 건너뛰는 다양한 오류 주입 공격이 가시화될 것이 예상되므로 이에 대한 대응책 마련이 필요하다.

참 고 문 헌

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

[2] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," CRYPTO'97, LNCS 1294, pp. 513-525, 1997.

[3] D. Boneh, R.A. DeMillo, and R.J. Lipton, "On the importance of checking cryptographic protocols for faults," EUROCRYPT'97, LNCS 1233, pp. 37-51, 1997.

[4] M. Joye, A.K. Lenstra, and J.J. Quisquater, "Chinese remaindering based cryptosystems in the presence of faults," Journal of Cryptology, vol. 12, no. 4, pp. 241-245, Dec. 1999.

[5] S.M. Yen, S.J. Kim, S.G. Lim, and S.J. Moon, "A Countermeasure against One Physical Cryptanalysis May Benefit Another Attack," ICISC'01, LNCS 2288, pp. 414-427, 2002.

[6] S.M. Yen and M. Joye, "Checking Before Output May Not Be Enough Against Fault-Based Cryptanalysis," In IEEE Transactions on Computers, vol. 49, no. 9, pp. 967-970, Sep. 2000.

[7] C. Kim and J.J. Quisquater, "Fault attacks for CRT based RSA : new attacks, new result and new countermeasures," WISTP'07, LNCS 4462, pp. 215-228, 2007.

[8] J.M. Schmidt and C. Herbst, "A Practical Fault Attack on Square and Multiply," Fault Diagnosis and Tolerance in Cryptoqraphy, FDTC'08, pp. 53-58, Aug. 2008.

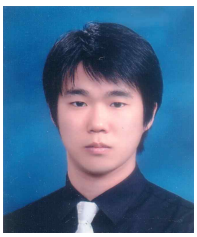
[9] M. Joye and S.M. Yen, "The Montgomery Powering ladder," CHES'02, LNCS 2523, pp. 291 - 302, 2002.

[10] E. Dottax, C. Giraud, M. Rivain, and Y. Sierra, "On Second-Order Fault Analysis Resistance for CRT-RSA Implementation," Available at http://eprint.iacr.org/2009/024, June 2009.

[11] N.E. Mrabet, "What About Vulnerability to a Fault Attack of the Miller's Algorithm Ducing an Identity Based Protocol?," ISA'09, LNCS 5576, pp. 122-134, 2009.

[12] C. Giraud, "An RSA Implementation Resistant to Fault Attacks and Simple Power Analysis," IEEE Trans on Computers, vol. 55, no. 9, pp. 1116-1120, Sep. 2006.

<著者紹介>



길 광 은 (Kwang Eun Gil) 학생회원
2008년 2월: 호서대학교 정보보호학과 졸업
2008년 3월 ~ 현재: 호서대학교 정보보호학과 석사과정
<관심분야> 네트워크 보안, 암호 알고리즘, 인증 및 평가

백 이 루 (Yi Roo Baek) 학생회원
2008년 8월: 호서대학교 정보보호학과 졸업
2008년 9월 ~ 현재: 호서대학교 정보보호학과 석사과정
<관심분야> 네트워크 보안, 프로토콜, 스마트 카드 보안

김 환 구 (Hwan Koo Kim) 종신회원
1987년 2월: 경북대학교 수학과 졸업
1991년 2월: 경북대학교 대학원 수학과 이학석사
1998년 5월: U. of Tennessee-Knoxville, 수학과, Ph. D.
2002년 3월 ~ 현재: 호서대대학교 정보보호학과 부교수
2004년 3월 ~ 현재: 한국정보보호학회 이사
<관심분야> 평가 및 인증, 암호학

하 재 철 (Jae Cheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월 ~ 2006년 1월: 나사렛대학교 전자계산소장, 학술정보관장, 입시학생처장
1998년 3월 ~ 2007년 2월: 나사렛대학교 정보통신학과 부교수
2006년 7월 ~ 2006년 12월: QUT in Australia 연구 교수
2007년 3월 ~ 현재: 호서대학교 정보호호학과 부교수
2002년 3월 ~ 현재: 한국정보보호학회 이사, 논문지 편집위원
2009년 1월 ~ 현재: 한국산학기술학회 이사
<관심분야> 정보보호, 네트워크 보안, 스마트카드 보안