Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.2, Apr. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.2.335

충돌 전력 분석 공격에 높은 공격 복잡도를 갖는 RSA 알고리즘에 대한 취약점 분석 및 대응기법*

김 수 리,[1†] 김 태 원,[1] 조 성 민,[1] 김 희 석,[2‡] 홍 석 희[1]
[1]고려대학교, [2]한국과학기술정보연구원

Analysis and Countermeasure on RSA Algorithm Having High Attack Complexity in Collision-Based Power Analysis Attack*

Suhri Kim,[1†] Taewon Kim,[1] Sungmin Jo,[1] HeeSeok Kim,[2‡] Seokhie Hong[1]
[1]Korea University, [2]Korea Institute of Science and Technology Information

요 약

부채널 분석 중 전력 분석 공격은 가장 실용적이며 강력한 기법으로 알려져 있다. 전력 분석 공격 중 단일 파형 공격은 단 하나의 파형을 이용하여 공개키 암호 시스템의 비밀정보를 복원하는 강력한 분석기법으로 최근에 활발히 연구되고 있다. 가장 최근에 Sim 등은 이러한 공격에 높은 안전성을 갖는 새로운 지수승 알고리즘을 소개하였다. 본 논문에서 Sim 등이 제안한 단일 파형 공격에 높은 공격 복잡도를 갖는 알고리즘의 취약점을 분석한다. 메시지 블라인딩과 지수 분할 기법에 윈도우 기법을 적용해 높은 공격 복잡도를 갖는 알고리즘을 제안하였지만 사전 연산 과정에서 발생하는 정보를 이용하여 비밀정보를 복원할 수 있음을 확인하였다. 또한 취약점을 보완하여 단일파형 공격에 높은 공격 복잡도를 갖는 지수승 알고리즘을 새롭게 제안하였다. 설계된 알고리즘은 사전 연산 과정에서 실제 지수 연산에 사용되는 값들의 재사용을 최소화 하여 충돌 공격에 대해 높은 안전성를 보장한다.

ABSTRACT

It is known that power analysis is one of the most powerful attack in side channel analysis. Among power analysis single trace attack is widely studied recently since it uses one power consumption trace to recover secret key of public cryptosystem. Recently Sim et al. proposed new exponentiation algorithm for RSA cryptosystem with higher attack complexity to prevent single trace attack. In this paper we analyze the vulnerability of exponentiation algorithm described by Sim et al. Sim et al. applied message blinding and random exponentiation splitting method on 2^t-ary for higher attack complexity. However we can reveal private key using information exposed during pre-computation generation. Also we describe modified algorithm that provides higher attack complexity on collision attack. Proposed algorithm minimized the reuse of value that are used during exponentiation to provide security under single collision attack.

Keywords: Collision Attack, RSA, Side Channel Analysis Analysis

Received(02. 04. 2016), Modified(03. 15. 2016), Accepted(03. 29. 2016)

* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대 정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구입니다(No. NRF-2014M3C4A7030649).

† 주저자, suhrikim@gmail.com
‡ 교신저자, hs@kisti.re.kr(Corresponding author)

I. 서 론

RSA 암호시스템은 인수분해의 어려움에 안전성을 보장받고 있다. 계산적으로 RSA 암호시스템의 개인키를 알아내기란 불가능하기 때문에 RSA 암호시스템이 동작할 때 발생하는 전력 소비량을 이용하여 개인키를 복원하는 공격을 행하고 있다. 이렇듯 암호시스템이 동작할 때의 발생하는 부가적인 정보를 이용하여 비밀정보를 복원하는 공격을 부채널 공격(Side-Channel Analysis Attack)이라 한다. 1999년 Kocher 등이 처음 제안한 이후에 많은 관련연구가 진행되고 있다[1,2,9].

부채널 공격 종류에는 시간차 공격 (Timing attack), 오류주입 공격 (Fault injection attack), 전자기파 분석 공격 (Electromagnetic analysis attack), 전력 분석 공격 (Power analysis attack)등이 있다[1,10,11]. 그 중 전력 분석 공격은 가장 강력하고 적은 비용으로 수행할 수 있는 현실적인 공격 방법이다. 이러한 전력 분석 공격에는 단일 파형을 이용한 단순 전력 분석 공격 (Simple Power Analysis, SPA)와 많은 수의 파형을 이용하여 통계적 특성을 적용한 차분 전력 분석 공격(Differential Power Analysis, DPA)가 있다[12].

전력 분석 공격을 막기 위한 다양한 대응기법 또한 연구되고 있다. 대표적으로 공개키 암호 알고리즘의 지수승 연산에서 단순 전력 분석 공격을 막기 위해서 곱셈 연산시 개인키에 의존하지 않고 알고리즘을 수행하는 square-and-multiply-always 가 있다. 또한 차분 전력 분석 공격을 막기 위해 비밀 중간값을 공격자가 예측할 수 없도록 메시지 블라인딩, 지수 분할 기법, 지수 랜덤화, 모듈러스 블라인딩 기법 등이 있다[12].

최근 단 하나의 파형에서 동일한 연산에 대한 충돌쌍을 이용하는 공격법이 활발하게 연구되고 있다[5,7,13]. 국내에서 Kim 등은 다양한 지수승 알고리즘에 대하여 단일 파형 충돌 공격법을 제안하였으며[14] 이에 대한 대응기법으로 역원 연산이 필요 없는 메시지 블라인딩 기법을 소개하였다[4]. 최근에는 Sim 등이 이를 발전시켜 윈도우 기법과 지수 분할 기법을 추가 적용하였다[6]. Sim 등은 역원 연산이 필요 없는 메시지 블라인딩 기법에 윈도우 방법과 지수 분할 기법을 적용하여 공격 복잡도를 높였다.

본 논문에서는 Sim 등이 제안 대응기법의 취약점을 분석한다[6]. 분석결과 제안된 공격 복잡도보다 훨씬 낮은 수치의 공격 복잡도를 가졌으며 이것은 지수승 연산을 위한 사전 테이블 계산 시 발생되는 취약점 때문이었다. 또한 우리는 사전 연산 테이블 구성시 발생할 수 있는 취약점을 보완하여 높은 공격복잡도를 갖는 새로운 지수승 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 충돌 공격과 관련된 기존 연구들에 대해 소개를 하고 3장에서 분석 대상 알고리즘인 Sim 등의 지수승 알고리즘을 소개한다. 4장에서 Sim 등의 알고리즘의 안전성을 분석하고 이에 안전한 대응기법을 제안한다. 마지막 5장에서 결론을 제시한다.

II. Related Works

충돌 공격은 Fouque et al. 의 Doubling attack 에 의해서 처음으로 개념이 제안되었다 [2]. 이 공격은 메시지 M과 M^2이 동일한 지수로 연산될 때 지수의 비트가 0일 경우에 연산 과정에서 동일한 연산이 일어나는 사실을 이용하는 공격이다. 공격자는 입력 값 M, M^2으로 지수승 연산을 수행한 파형을 얻는다. 만일 개인키 $d=(d_{n-1},....,d_0)_2$에서 $d_i=0$이라면 M의 $i+1$번째와 M^2의 i번째 지수승 연산이 동일하여 두 파형의 상관관계가 높게 나타난다. 하지만 $d_i=1$일 경우에는 서로 다른 입력값에 대한 지수승 연산이 수행되기 때문에 상관관계가 낮다. 따라서 위의 사실을 이용하여 공격자는 개인키의 한 비트씩 복원할 수 있다.

단 하나의 파형을 이용하여 개인키를 복원하는 공격으로는 Walter의 Big Mac Attack이 있다[3]. 이 공격은 sliding window를 이용한 지수승 연산을 대상으로 한다. 지수승 연산이 수행되기 전 사전 연산 테이블을 생성하는 단계에서 곱셈의 사용되는 사전 연산값을 특징화 한다. 지수승 연산은 비밀 지수값에 따라 사전 연산값을 참조하므로 미리 특징화해 놓은 것과 지수승과의 상관관계를 이용하여 비밀값을 복원한다.

[7]에서는 수평적 상관관계 분석이란 개념이 최초로 소개되었다. 이는 단 하나의 파형에서 각 지수승 연산을 워드곱 단위로 분할하여 비밀 중간값과 상관관계를 측정한다. 수평적 상관관계 분석은 Big Mac Attack과는 달리 입력 메시지 값을 알아야 공격이 가능하다.

Bauer 등은 단일 파형에서 충돌 공격을 이용하여 개인키를 복원하는 기법을 소개하였다[5]. 이 공격에서 가장 핵심적인 아이디어는 연산 과정에서 동일한 입력 값을 사용하는 경우에 충돌이 일어난다는 사실이다.

Kim 등은 알고리즘 내부에서 발생할 수 있는 키 비트에 의존한 충돌 특성을 이용하였다[15]. 연속한 두 번의 지수승 연산 수행 시 지수에 따라 입력값이 동일하다면 충돌이 발생되고 이러한 충돌로부터 비밀 지수를 복원할 수 있다. 이 공격은 메시지 블라인딩, 랜덤 프로젝티브 코디네이트(Random Projective Coordinate), 몽고메리 레더(Montgomery Ladder)등과 같이 전력 분석 공격에 안전한 알고리즘에도 적용할 수 있다.

III. 기존 대응기법

본 장에서는 충돌 공격에 안전한 대응기법을 소개한다. 우선 Kim 등이 제안한 기법을 서술한 후 이를 발전시킨 Sim 등의 제안기법을 설명한다

3.1 역원 연산이 필요 없는 메시지 블라인딩 기법

Kim 등이 제안한 역원 연산이 필요 없는 메시지 블라인딩 기법은 오일러-파이 함수를 이용하는 특징이 있다[4]. 기존 메시지 블라인딩 기법은 메시지 M과 랜덤값 R에 대해 $(MR^e)^d \bmod n = M^d R \bmod n$ 을 연산한다. 그 후 올바른 암호문 $M^d \bmod n$를 얻기 위해서는 R의 역원을 계산하여 $R^{-1} \bmod n$을 곱하는 추가적인 연산이 필요하다. 하지만 오일러-파이 함수가 임의의 정수 a에 대하여 $a^{\Phi(n)} \equiv 1 \bmod n$임을 이용하면 지수승의 최종 연산값이 $M^d R^{\Phi(n)} \bmod n$이고 $R^{\Phi(n)} \equiv 1 \bmod n$이 되므로 역원연산을 수행할 필요가 없어 효율적이다.

알고리즘 1의 3번째 단계에서 볼 수 있듯이 $M^d R^{\Phi(n)} \bmod n$을 연산할 때 SPA 공격에 대응하기 위해 4.2 단계처럼 지수값에 상관 없이 항상 곱셈 연산이 일어나기 위해 사전연산 테이블을 이용한다. 이 때 사전 연산에 사용되는 값은 [Table 1]과 같다.

위의 표에서 $(d_i, \Phi_i) = (1,0)$일 경우 메시지의 노출이 일어나 DPA 공격에 대해 취약하다. DPA 공격에 대해서도 안전하기 위해 사전 연산된 값에 항상

Algorithm 1

Input : $M, d = (d_{t-1} d_{t-2} \ldots d_0)_2$

$\Phi(n) = (\Phi_{t-1} \Phi_{t-2} \ldots \Phi_0)_2, \Phi_{t-1} = 1$

RSA modulus n

Output : $M^d \bmod n$

I. If $M=1$, return 1; if $M=-1$, return $1-2d_0$

II. Generate random numbers

$R \in \{1,2,\ldots,n-1\}$, $v \in \{0,1,2,3\}$

III. $S=1, S_v = R \bmod n, S_{v \oplus 1} = R^2 \bmod n$

$S_{v \oplus 2} = MR \bmod n, S_{v \oplus 3} = MR^2 \bmod n$

IV. For $i = t-1$ down to 1

4.1 $S = S^2 \bmod n$

4.2 $S = S \times S_{v \oplus (2d_i + \Phi_{i-1})} \bmod n$

5. Return $S^2 \times S_{v \oplus (2d_0+1)} \bmod n$

랜덤값이 곱해진 형태를 테이블에 저장한다. 이를 위해 $\Phi(n) = (\Phi_{t-1} \Phi_{t-2} \ldots \Phi_0)_2$ 대신에 $\widetilde{\Phi}_i = \Phi_i + 1$을 이용해 가능한 $(d_i, \widetilde{\Phi}_i)$를 사전연산 테이블에 저장한다. [Table 2]는 사전 연산 값들을 나타냈다.

[Table 1]와 달리 [Table 2]는 기존에 $(d_i, \Phi_i) = (1,0)$일 때 M을 곱하기 때문에 메시지가

Table 1. Precomputation value of $M^d R^{\Phi(n)} \bmod n$

d_i	Φ_i	Precomputation value
0	0	1
0	1	R
1	0	M
1	1	MR

Table 2. Precomputation value when using $\widetilde{\Phi}_i$

d_i	Φ_i	$\widetilde{\Phi}_i$	Precomputation value
0	0	1	R
0	1	2	R^2
1	0	1	MR
1	1	2	MR^2

노출되지만 $(d_i,\Phi_i+1)=(d_i,\tilde{\Phi}_i)=(1,1)$을 이용하면 사전연산에서 MR을 참조해서 곱하기 때문에 메시지 노출을 방지할 수 있다.

〔Table 2〕를 사전연산 값으로 사용하기 위해서는 $\tilde{\Phi}_i$를 이용하면서 연산이 종료 된 후 $M^d \bmod n$이 출력될 수 있도록 R의 지수를 바꿔야 한다. 식 (1)에서 $2\Phi(n)$은 $\tilde{\Phi}_i$를 이용한 지수 표현이 가능하다. 따라서 $M^d R^{\Phi(n)} \bmod n$ 대신에 $M^d R^{2\Phi(n)} \bmod n$으로 연산을 하게 된다면 연산이 종료 된 후 $M^d \bmod n$가 정상 출력 되면서 $R^{\Phi(n)}$을 사용할 때와 동일하게 메시지 블라인딩에 대한 역원 연산이 필요 없게 된다. 또한 $\tilde{\Phi}_i$을 이용하기 때문에 〔Table2〕 사용이 가능해져 SPA에 안전하고, 모든 사전연산 값이 랜덤값으로 마스킹 되기 때문에 DPA에 안전하다.

$$\begin{aligned} 2\Phi(n) &= \sum_{i=1}^{t} \Phi_{i-1} 2^i = \left\{ \sum_{i=1}^{t-1} (\Phi_{i-1}+1) 2^i + 1 \right\} + 1 \\ &= \sum_{i=1}^{t-1} \tilde{\Phi}_{i-1} 2^i + 2 \\ &= (\tilde{\Phi}_{t-2} \tilde{\Phi}_{t-3} ... \tilde{\Phi}_0 2)_2 \end{aligned} \quad (1)$$

실제 알고리즘 수행에 있어서 $\tilde{\Phi}_i$을 참조할 때와 Φ_i참조할 때 동일한 테이블 값을 사용하게 되기 때문에 $\tilde{\Phi}_i$에 대한 별도의 레코딩 없이 Φ_i을 사용한다. 마지막으로 Algorithm1 단계에서 랜덤값 v는 참조하는 테이블의 레지스터 주소를 알고리즘 수행마다 바꾸기 때문에 Address bit DPA 에 안전하다〔15〕.

3.2 윈도우 기법과 지수 분할 기법을 적용한 메시지 블라인딩 기법

Sim 등은 Kim 등이 제안한 역원 연산이 필요 없는 메시지 블라인딩 기법에 추가적으로 두 가지 기법을 적용하여 공격 복잡도를 높였다. 첫째로 지수승 연산 시 지수 값을 w만큼 스캔하여 연산하는 윈도우 기법과 둘째로는 랜덤값 r을 사용하여 지수 d를 $\hat{d}=d-r$와 r로 분할하여 지수승 연산을 $M^r M^{d-r} R^{\Phi(n)} \bmod n$ 형태로 수행하는 지수 분할 기법이다. 따라서 1비트씩 진행되었던 지수승 연산을 w비트로 확장시켜야 한다. 이를 위해 SPA와 DPA에 안전하도록 사전 연산 테이블을 이용한 지수승 연산을 w비트로 확장시켜야 한다. 윈도우 사이즈 w에 대해서 $W=2^w$이고 $\hat{d}=d-r$라 할 때, 지수승 연산에서 사용하는 세 비밀 지수 $r, \hat{d}, \Phi(n)$은 다음과 같이 표현한다.

$$\begin{aligned} r &= (r_{t-1} r_{t-2} ... r_0)_2 = (r'_{k-1} r'_{k-2} ... r'_0)_W \\ \hat{d} &= (\hat{d}_{t-1} \hat{d}_{t-2} ... \hat{d}_0)_2 = (\hat{d}'_{k-1} \hat{d}'_{k-2} ... \hat{d}'_0)_W \\ \Phi(n) &= (\Phi_{t-1} \Phi_{t-2} ... \Phi_0)_2 = (\Phi'_{k-1} \Phi'_{k-2} ... \Phi'_0)_W \end{aligned}$$

지수승 연산은 각 비밀지수 $(r_i', \hat{d}_i', \Phi_i')$를 읽어 $M^r M^{\hat{d}} R^{\Phi'(n)} \bmod n$ 에 해당하는 값 $M^{r_i'+\hat{d}_i'} R^{\Phi'_i} \bmod n$을 사전 연산 테이블에서 참조하여 곱한다. 따라서 사전 연산되는 종류는 $r'_i+\hat{d}'_i$와 Φ'_{i-1}의 조합의 가짓수에 따라 결정된다. 윈도우 기법은 한번에 w비트를 읽어 $W=2^w$진수로 연산하기 때문에 각 단위 $r_i', \hat{d}_i', \Phi_i'$는 0부터 $W-1$사이의 값을 가지게 되고, $r'_i+\hat{d}'_i$ 값의 범위는 $r'_i+\hat{d}'_i \in \{0, 1, ..., 2W-2\}$ 가 되어 총 $W(2W-1)$ 가지의 사전 연산 값이 존재하게 된다.

이 때 $\Phi'_i=0$일 경우 $M^{r_i'+\hat{d}_i'} \bmod n$이 저장되기 때문에 $2W-2$개의 메시지의 노출이 일어난다. 이를 방지하기 위해 Algorithm 1에서와 유사한 방법으로 Φ_i'대신에 $\widetilde{\Phi'}_i=\Phi'_{k-1}(W-1)+\Phi'_i$를 만족하는 $\tilde{\Phi}_i'$을 이용하여 사전연산 값에 항상 랜덤값이 곱해져 메시지의 노출을 방지하는 테이블을 만든다. 이러한 사전 연산 테이블을 이용하기 위해서는 $\tilde{\Phi}_i'$을 사용하면서 최종 연산 결과가 달라지지 않도록 R의 지수를 변경해야 한다. 식 (3)은 $W\Phi(n)$를 $\tilde{\Phi}_i'$로 이용한 표현이 가능하도록 한다. 따라서 $M^r M^{\hat{d}} R^{W\Phi(n)} \bmod n$으로 연산을 수행 한다면 $R^{\Phi(n)}$을 사용할 때와 동일하게 역원연산이 필요 없으면서 $\tilde{\Phi}_i'$을 이용하는 사전 연산 테이블 참조를 할 수 있다. 또한 Φ_i'을 참조할 때와 $\tilde{\Phi}_i'$을 참조할 때 동일한 테이블 값을 사용하기 때문에 $\tilde{\Phi}_i'$에 대한 별도의 레코딩 없이 Φ_i'을 사용할 수 있다.

Algorithm 2

Input : $M, d=(d_{t-1}d_{t-2}...d_0)_2$

$\Phi(n)=(\Phi_{t-1}\Phi_{t-2}...\Phi_0)_2, \Phi_{t-1}=1$

RSA modulus n

Output : $M^d \bmod n$

1. If $M=1$, return 1; if $M=-1$, return $1-2d_0$
2. Generate random numbers $R\in\{1,2,...,n-1\}$, $v\in\{0,1,...,2W^2-W-1\}$, r (t-length)
3. Compute $\hat{d}=d-r$
4. Pre-computation
   4.1 For $i=0$ up to $W-1$
      4.1.1 $S_{v\oplus i}=R^{\Phi'_{k-1}(W-1)+i} \bmod n$
   4.2 For $i=0$ up to $2W^2-2W$ by W
      4.2.1 For $j=0$ up to $W-1$
         4.2.1.1
         $S_{v\oplus(i+j+W)}=S_{v\oplus(i+j)}\times M \bmod n$
5. For $i=k-1$ down to 1
   5.1 $S=S^W \bmod n$
   5.2 $S=S\times S_{v\oplus(W(r'_i+\hat{d}'_i)+\Phi'_{i-1})} \bmod n$
6. Return $S\times S_{v\oplus(W(r'_0+\hat{d}'_0)+\Phi'_{k-1})} \bmod n$

$$
\begin{aligned}
W\Phi(N) &= \sum_{i=1}^{k}\Phi'_{i-1}W^i \\
&= \Phi'_{k-1}\left\{(W-1)\sum_{i=1}^{k-1}W^i+W\right\}+\sum_{i=1}^{k-1}\Phi'_{i-1}W^i \\
&= \sum_{i=1}^{k-1}\{\Phi'_{k-1}(W-1)+\Phi'_{i-1}\}W^i+W\Phi'_{k-1} \\
&= \sum_{i=1}^{k-1}\widetilde{\Phi}'_{i-1}W^i+W\Phi'_{k-1} \qquad (3)
\end{aligned}
$$

Algorithm2는 3.1에서 설명한 역원 연산이 필요 없는 메시지 블라인딩 기법에 윈도우 기법과 지수 분할 기법을 적용한 알고리즘이다. Algorithm 2의 단계-4에서는 지수승 연산에 사용될 사전 연산 테이블을 구성한다. 이를 이용하여 지수승 연산은 단계-5에서 이루어지며, 단계-5.2에서 지수값에 따라 사전 연산값을 참조하여 곱셈연산을 수행한다. 특히 비밀지수 $r_i', \hat{d}_i', \Phi'_{i-1}$에 관계없이 항상 곱셈이 이루어지기 때문에 SPA에 안전하다.

이해를 돕기 위해 단계-4의 사전 연산을 예를 들어 설명한다. $w=2$일 때 Algorithm2의 단계-4.1에서 $S_{v\oplus 0}=R^3$, $S_{v\oplus 1}=R^4$, $S_{v\oplus 2}=R^5$, $S_{v\oplus 3}=R^6$을 연산한다. 그 후 단계-4.2에서 단계-4.1에 연산한 값을 이용하여 메시지와 곱셈연산을 수행한다. 즉, $S_{v\oplus 4}=S_{v\oplus 0}\times M$ 의 수행을 위하여 사전 연산된 값 $S_{v\oplus(i+j)}$에 메시지를 곱하는 방식으로 사전 연산값을 계산한다. 이와 같은 경우 사전 연산 값은 총 28개가 되고 사전 연산의 값은 다음 테이블과 같다.

만약 $(r_i', \hat{d}_i', \Phi'_{i-1})=(0,1,0)$일 경우 레지스터 S에 MR^3을 불러와 곱하는 방식으로 지수승 연산이 진행된다.

Table 3. Precomputation value of Algorithm 2 when $w=2, \Phi'_{k-1}=1$

r_i'	$\hat{d}'_i$	Φ'_{i-1}	$\hat{\Phi}'_{i-1}$	Precomputation value
0	0	0	3	R^3
0	0	1	4	R^4
0	0	2	5	R^5
0	0	3	6	R^6
0	1	0	3	MR^3
0	1	1	4	MR^4
…	…	..	..	…
3	3	3	6	M^6R^6

3.3 공격 복잡도

본 장에서는 Kim 등과 Sim 등이 제안한 지수승 연산 알고리즘의 공격 복잡도를 설명한다. 두 기법에 대한 공격 모델은 단일 파형 충돌 공격으로 가정한다. 즉, 공격자는 곱셈 연산 시 동일한 피연산자를 사용하는 두 곱셈에 대하여 충돌이 발생함을 전력 소모량(전자파량)을 이용하여 알 수 있다. 또한 Sim 등의 논문과 비교를 위해, 이 논문에서 사용한 공격 복잡도의 정의를 이용한다. Sim 의 논문에 의하면 공격자가 단일 파형 공격을 행할 때, 지수를 복원하

기 위해 추측해야 하는 모든 경우의 수를 공격 복잡도라 정의한다 [6].

Kim 등은 지수승 연산 중 곱셈연산(Algorithm 1의 단계-4.2)을 수행하기 위하여 4가지 종류의 사전연산 값을 사용한다. 따라서 지수승 연산 시 곱셈연산에 사용되는 피연산자 중 하나는 반드시 사전 연산된 4가지 값 중 하나일 것이다. 공격자는 수집된 파형 중 곱셈연산이 수행된 부분을 분리한 후, 충돌 공격을 통해 이 들을 4가지로 그룹화 시킨다. 이 때 마지막 단계-5의 곱해지는 사전 연산 값은 d_0에만 의존하기 때문에 단계-5의 곱해지는 사전 연산 값은 2종류 이다. 따라서 단계-5 부분이 속해있는 그룹은 다른 그룹과 달리 사용되는 사전 연산 값이 2가지 중 하나이다. 나머지 세 개의 그룹에 대해서는 전수조사를 한다면 공격자는 d_0의 가짓수 × 세 그룹의 전수 조사 = $2\times3!=12$ 가지 경우를 추측해야 하므로 총 12의 공격 복잡도를 가지게 된다.

Sim 등의 알고리즘의 공격 복잡도도 동일한 방법으로 기술한다. Sim등의 알고리즘은 사용되는 윈도우 사이즈에 따라 사전 연산 값의 수가 달라진다. 윈도우 사이즈 w를 사용한다고 가정하고 $W=2^w$라 하면 사전 연산 값의 수는 $W(2W-1)$가지이다. 따라서 지수승 연산 시 곱셈 연산에 사용되는 피연산자 중 하나는 반드시 사전 연산된 $W(2W-1)$값 중 하나일 것이다. 공격자는 수집된 파형 중 사전 연산 값과 곱셈연산이 수행된 부분을 분리한 후 충돌 공격을 통해 $W(2W-1)$가지로 그룹화 한다. Algorithm2의 단계5-2와 달리 단계-6은 $r'_i+\hat{d}'_i$과 Φ'_{k-1}에 따라 사용되는 사전 연산 값이 결정된다. $r'_i+\hat{d}'_i$는 $2W-1$가지 Φ'_{k-1}는 $W-1$가지 존재하기 때문에 단계-6이 속한 그룹의 지수는 $(2W-1)(W-1)$가지 중 하나이다. 나머지 $W(2W-1)-1$개의 그룹에 대해서는 전수조사를 진행한다. 공격자는 지수승의 모든 비트를 복원하기 위하여 총 $(2W-1)(W-1)\times(2W^2-W-1)!$가지 경우를 추측해야 하므로 총 $(2W-1)(W-1)\times(2W^2-W-1)!$의 공격 복잡도를 가지게 된다. 윈도우 사이즈가 2, 3, 4에 대해서 Sim등의 알고리즘의 공격 복잡도는 다음 [Table 4]와 같다.

Table 4. Attack complexity by window size

w	2	3	4
Attack complexity	$21\times27!$	$105\times119!$	$465\times495!$

Algorithm 2는 윈도우 사이즈가 2만 되어도 공격복잡도가 12인 Algorithm 1과 차이가 크기 때문에 Algorithm 1에 비해 안전하다

IV. 취약점 분석

본 장에서는 Sim 등이 제안한 지수승 알고리즘에 대해 단일 파형 공격을 이용하여 취약점을 분석한다. 위에서 언급했듯이 단일 파형 공격은 공격자가 곱셈 연산 시 동일한 연산 값을 사용하는 두 곱셈을 충돌 발생 여부로 확인할 수 있다고 가정한다. 이 충돌 발생 여부는 연산 순서와 상관없이 동일한 값이 사용이 되었다면 상관관계로 인해 공격자는 인지할 수 있다 [16]. 또한 공격자는 Algorithm2 의 단계-5.2와 단계-6의 곱셈 연산에서 사용된 사전 연산 값을 알아낸다면 비밀 지수를 복원할 수 있다. 따라서 우리는 단일 파형 공격을 이용하여 사전 연산된 값을 복원하는데 초점을 맞춘다.

Sim 등이 제안한 지수승 알고리즘에서 사전 연산 테이블 구성은 순차적으로 이루어진다는 점에 주목할 필요가 있다. 이러한 규칙성으로 인하여 다음과 같은 취약점이 발생한다. 예를 들어, 윈도우 사이즈 $w=2, \Phi'_{k-1}=$일 때 Algorithm2 의 단계-4.1은 항상 4번씩 사전 연산 값을 생성하는 연산이 일어난다. 공격자는 파형에서 사전 연산 테이블이 생성되는 부분을 추출해서 곱셈 연산이 일어나는 부분으로 나눈다. 이렇게 나눈 부분을 4개씩 그룹 지으면 한 그룹이 알고리즘의 각 i번째 단계라는 점을 알 수 있다. 따라서 공격자는 첫 번째 그룹이 R^3, R^4, R^5, R^6가 생성되는 부분이라는 것을 알 수 있으며, 두 번째 그룹은 R^3, R^4, R^5, R^6 호출되어 메시지 M과 곱셈 연산이 일어나 사전 연산 값 MR^3, MR^4, MR^5, MR^6 이 생성된다는 사실을 알 수 있다. 공격자는 비밀 지수 값을 알아내기 위해 파형에서 실제 지수 연산이 수행되는 부분인 단계5-2를 각 라운드 별로 추출한다. 공격자는 단계 5-2의 파형과 단계4-1의 부분파형들과의 상관관계를 구해 비교를 한다. 예를 들어 단계 5-2의 i번째 부분 파형과 단계 4-1의 5

번째 파형과 상관관계가 가장 높게 나타났다면, 공격자는 단계 4-1의 5번째 파형은 R^3이 호출되어 MR^3이 생성되는 부분파형이라는 것을 알고 있기 때문에 공격자의 가정에 의해 i번 째의 단계 5-1 파형은 R^3이 사전 연산 테이블에서 호출되었다는 사실을 알 수 있다. 이를 통해 공격자는 i 번째의 비밀값은 $(r_i', \hat{d}_i', \Phi'_{i-1}) = (0,0,0)$ 임을 알 수 있다.

이를 확장 시켜 임의의 윈도우 사이즈 w와 Φ'_{k-1}에 대한 공격 시나리오는 다음과 같다. 공격자는 개인키 $d = (d'_{k-1}, d'_{k-2}, ..., d'_0)_W$와 메시지 M에 대해 Algorithm2를 수행한 파형을 획득한다. 공격자는 획득한 파형을 Algorithm2의 단계-5.2 와 단계-6 부분 k개로 분리하고 각각의 파형을 O_i라 한다. 공격자는 획득한 파형에서 Algorithm2의 단계-4.2.1.1을 연산하는 부분 T_j $(0 \le j < W(2W-1))$에 대해 과 O_i와 상관관계 구한다. 각 T_j에 대해 O_i $(0 \le i \le k-1)$ 과 상관관계값을 계산한 것 중 제일 높은 값을 가질 때 사용한 T_j가 공격자가 찾는 $(r_i', \hat{d}_i', \Phi'_{i-1})$로 하여 지수를 복원한다. 이를 통해 공격자는 단일 파형으로 개인키 d를 복원할 수 있다.

공격자는 $i = 2W^2 - W$일 때 사전 연산 값 W개를 테이블에 저장한 후 바로 Algorithm2의 5단계로 넘어가기 때문에, W에 대한 사전 연산값에 대해서는 전수조사를 할 수 밖에 없다. 따라서 공격 복잡도는 $W!$라 할 수 있다. 이는 기존 논문에서 제시한 $(2W-1)(W-1)(2W^2-W-1)!$에 비해 더 작은 공격 복잡도를 갖는다.

V. 대응 기법

Sim 등의 기법은 사전 연산 시 저장되는 레지스터가 랜덤하게 바뀜에도 불구하고 순서대로 연산되어 저장된다는 점과 사전 연산 테이블을 구성하기 위해 연산되는 피연산자가 지수승 연산에서 재사용됨으로 인해 충돌이 발생한다는 취약점을 가진다. 이러한 취약점을 보안하고, 높은 공격 복잡도를 갖는 지수승 알고리즘을 소개한다.

제안하는 알고리즘은 Sim 등의 기법에서 사용된 지수 분할 기법, 메시지 블라인딩 기법, 윈도우 기법을 적용해 사전 연산 테이블 참조하는 방식을 유지한다. 따라서 우리는 사전 연산 테이블을 구성하는 부분만을 수정한다. 기본적인 아이디어는 지수승 연산에 참조되지 않는 값들만으로 사전 연산 테이블을 구성하는 것이다.

Algorithm2는 사전 연산 테이블을 구성 할 때 $M^i R^j (0 \le i \le 2W-2, W-1 \le j \le 2W-2)$ 값을 계속 이용하기 때문에 충돌 공격에 취약함을 보였

Algorithm 3

Input : $M, d = (d_{t-1} d_{t-2} ... d_0)_2$

$\Phi(n) = (\Phi_{t-1} \Phi_{t-2} ... \Phi_0)_2, \Phi_{t-1} = 1$

RSA modulus n

Output : $M^d \bmod n$

1. If $M=1$, return 1; if $M=-1$, return $1-2d_0$
2. Generate random numbers $R \in \{1,2,...,n-1\}$, r (t-length),
   $v \in \{0,1,...,2W^2 - W - 1\}$
   $v' \in \{0,1,..., W-3\}$
3. Compute $\hat{d} = d - r$
4. Pre-computation
   4.1 $i = 0$ up to $W-3$
      4.1.1 $T_{v' \oplus i} = R^{i+1} \bmod n$
   4.2 $T' = R^{W-2}$
   4.3 $i = -W$ up to $2W^2 - 3W$ by W
      4.3.1 $j = 0$ up to $W-3$
         4.3.1.1 $S_{v \oplus (i+j+W)} = T_{v' \oplus j} \times T'$
      4.3.2 $S_{v \oplus (i+2W-2)} = S_v \times T'$
      4.3.3 $S_{v \oplus (i+2W-1)} = S_{v \oplus 1} \times T'$
      4.3.4 $T' = T' \times M$
5. For $i = k-1$ down to 1
   5.1 $S = S^W \bmod n$
   5.2 $S = S \times S_{v \oplus (W(r'_i + \hat{d}'_i) + \Phi'_{i-1})} \bmod n$
6. Return $S \times S_{v \oplus (W(r'_0 + \hat{d}'_0) + \Phi'_{k-1})} \bmod n$

다. 따라서 사전 연산 테이블에 저장된 값과 사전 연산 테이블을 구성하기 위하여 사용되는 피연산자들이 중복되지 않도록 설계한다. 즉, 지수승 연산에서 사용되지 않는 M^iR^{W-2} $(0 \le i \le 2W-2)$ 과 $R^i(1 \le i \le W-2)$ 를 이용하여 사전 연산 테이블을 구성함으로써 충돌공격을 방지한다.

충돌 공격을 방지하기 위한 사전 연산 테이블 구성은 다음과 같다. 우선 Algorithm3의 단계-4.1에서 사전 연산 테이블 구성하기 위한 고정 값 $R, R^2, ..., R^{W-2}$을 $W-2$개의 레지스터 T에다가 저장한다. 단계-4.2에서 앞으로 사전 연산 테이블 구성에 있어서 사용될 M^iR^{W-2}을 위한 R^{W-2}를 레지스터 T'에 한 번 더 저장한다. 단계-4.3에서는 실제 사전 연산 테이블에 저장될 값들이 연산된다. $i=-W$일 때 각 T에 저장된 $R, R^2, ..., R^{W-2}$에 T'에 저장된 R^{W-2}을 곱해 $R^{W-1}, R^W, ..., R^{2W-2}$를 만들어 사전 연산 테이블에 저장한다. 지수승 연산에서 사용되는 $R^j(W-1 \le j \le 2W-2)$ 보다 사용하지 않은 $R, R^2, ..., R^{W-2}$가 2개 적기 때문에 R^{W-1}, R^W는 중복 사용될 수밖에 없다. 따라서 단계-4.3.2와 단계-4.3.3 에서는 R^{W-1}, R^W에 T'에 저장된 R^{W-2}을 곱해 사전 연산 테이블에 저장된다. 단계-4.3.4의 T'은 메시지 M이 곱해져 MR^{W-2}로 업데이트 한다. $i=0$일 때 다시 T에 저장된 $R, R^2, ..., R^{W-2}$에 T'을 곱해 사전 연산 테이블에 저장하고, R^{W-1}, R^W은 중복 사용하여 T'을 곱해 사전 연산 테이블에 저장한다. 그 후 단계-4.3.4에 T'에 메시지 M을 곱해 M^2R^{W-2}로 업데이트 한다.

제안하는 알고리즘은 2개의 사전 연산 값 R^{W-1}, R^W를 제외하고 나머지 사전 연산 값들은 사전 연산 테이블을 구성하는 과정에서 반복 사용되지 않기 때문에 Sim 등의 기법에서 발생한 취약점에 대해 안전하다. 윈도우 사이즈 w에 대해 $W=2^w$라 하고 개인키 $d=(d'_{k-1}, d'_{k-2}, ..., d'_0)_W$일 때, 공격자는 전체 $W(2W-1)$가지의 사전 연산 값 중 두 개의 사전 연산 값만 알 수 있으므로 개인키의 W비트에 대해서 $W(2W-1)-2$개의 전수 조사를 실시해야 한다. 개인키 전체는 W비트 씩 k개만큼 존재하므로 총 $((W(2W-1)-2)!)^k$의 공격 복잡도를 가진다. 아래의 〔Table 5〕는 1024비트 RSA에서 윈도우 사이즈에 따른 Sim 등의 기법과 제안하는 기법의 공격 복잡도를 비교하였다. Table 5의 세 번째 줄은 Sim 등의 기법에 비해 제안하는 알고리즘의 공격 복잡도의 증가량을 나타낸 수치이다. 이것은 윈도우 사이즈가 증가할수록 더 높은 공격 복잡도 향상을 보인다.

Table 5. Comparison of attack complexity for various window size in RSA 1024

w	4	5	6
Sim's method	$465 \times 495!$	1953×201	8001×812
Proposed method	$(494!)^{256}$	$(2014!)^{203}$	$(8126!)^{171}$
≒increased rate	10^{285060}	$10^{1167908}$	$10^{11056170}$

제안하는 알고리즘은 Sim 등이 기법과 비교하여 단계-4.3.4의 곱셈 연산이 추가된다. 따라서 Sim 등의 기법은 사전 연산 테이블을 생성하기 위하여 총 $(2W-1)W$의 곱셈 연산이 필요한 반면 제안하는 알고리즘은 총 $(2W-1)(W+1)$의 곱셈 연산이 필요하다. 구체적으로 윈도우 4, 5, 6, 7에 대해 각각 106%, 103%, 101%, 100.7% 만큼의 추가적인 비용이 발생한다. 그러므로 제안하는 알고리즘은 Sim 등의 기법과 거의 동일한 효율성을 가지면서 훨씬 더 높은 공격 복잡도를 제공한다.

VI. 결 론

본 논문은 단일 파형 충돌 공격에 대해 높은 공격 복잡도를 갖는 지수승 알고리즘의 취약점을 분석하였다. 분석 결과, 사전 연산 테이블을 구성하는 과정에서 순차적으로 연산이 진행되어 테이블에 저장된다는 점과 사전 연산 테이블을 구성하기 위해 연산되는 피연산자가 지수승 연산에서 재사용된다는 점을 이용하면 단일 파형 충돌 공격에 대해 취약함을 밝혔다. Sim 등의 기법은 $(2W-1)(W-1)(2W^2-W-1)!$의 공격 복잡도를 가졌다고 소개되었으나, 취약점으로 인해 $W!$로 공격 복잡도가 감소함을 확인하였다.

또한 우리는 취약점을 보완한 새로운 지수승 알고리즘을 제안하였다. 사전 연산 테이블에 저장된 값과 사전 연산 테이블을 구성하기 위하여 사용되는 피연산자들 사이에 충돌이 발생하지 않도록 설계하였다. 그 결과 $((W(2W-1)-2)!)^k$만큼의 공격 복잡도를 가지며 이것은 기존 Sim 등이 기대했던 공격 복잡도보다 훨씬 높은 수치이다.

References

[1] P. Kocher, J. Jaffee, and B. Jun, "Differential power analysis," CRYPTO '99 LNCS 1666, pp. 388-397, 1999

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

[3] C.D. Walter, "Sliding windows succumbs to big mac attack," LNCS, pp. 286-299, 2001

[4] H, Kim, D. Han, S. Hong, J. Ha, " Message Blinding Method Requiring No Multiplicative Inversion of RSA," ,ACM vol. 13 no. 4, Article 80, Feb. 2014.

[5] Bauer, A., Jaulmes, E., Prouff, E., Wild, J, "Horizontal collision correlation attack on elliptic curves," Selected Areas in Cryptography, vol. 8282, pp. 553-570, 2013

[6] B. Sim, Y. Won, D. Han, "Study for improving attack complexity against RSA collision analysis," Journal of the Korea Institute of Infomation Security & Cryptology, vol.25, no. 2, April. 2015

[7] Clavier. C, Feix, B. Gagnerot, G. Roussellet, M., Verneuil, V, "Horizontal correlation analysis on exponentiation," ICICS, Lecture Notes in Computer Science, vol. 6917, pp. 46-61,2010.

[8] E. Brier, Christophe Clavier, and Francis Olivier, "Correlation Power Analysis with a Leakage Model," Cryptographic Hardware and Embedded Systems - CHES, Lecture Notes in Computer Science vol.3156 pp16-29, 2004

[9] P. Kocher, "Timing Attacks on Implemntations of Diffie-Hellman, RSA, DSS and Other Systems," Proc. CRYPTO '96, LNCS vol 1109, pp. 104-113, 1996.

[10] I. Biehl, B. Meyer, and V. Muller, "Differential Fault Attacks on Elliptic Curve Cryptosystems," CRYPTO, vol.1880, pp131-146, May 2000.

[11] Coron, J.S , " Resistance against differential power analysis for elliptic curve cryptosystems:" ASIACRYPT '98, LNCS, vol. 1514. pp. 51-65, 1998

[12] Marc F. Witterman, Jasper G.J. can Woudenberg, Federico Menarini, "Defeating RSA multiply-always and message blinding countermeasure," CT-RSA 2011, LNCS 6558, pp. 77-88, 2011

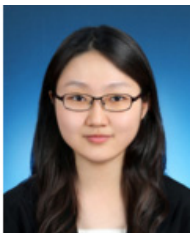
[13] H. Kim, T. Kim, J. Youn, S. Hong, "Pratical Second-Order Correlation Power Analysis on the Message Blinding Method and Its Novel Countermeasure for RSA," ETRI, vol.32 no.1, pp. 102-111, Feb 2010

[14] N. Hanley, H. Kim, M. Tunstall, " Exploiting Collision in Addition Chain-Based Exponentiation Algorithms Using a Single Trace," CT-RSA, LNCS 9048, pp 431-448, 2015

[15] K. Itho, T. Izu, M. Takenaka, "Address-Bit Differential Power Analysis of Cryptographic Schemes OK-ECDH and OK-ECDSA," CHES 2002, LNCS 2523, pp. 129-143, 2003.

[16] A. Bauer, E. Jaulmes, E. Prouff, J. Wild "Horizontal Collision Correlation Attack on Elliptic Curves," SAC 2013, pp. 553-570, 2014.

〈저자소개〉



김 수 리 (Suhri Kim) 학생회원
2014년 2월: 고려대학교 수학과 학사
2014년 8월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 부채널 공격

김 태 원 (Taewon Kim) 학생회원
2010년 2월: 광운대학교 수학과 학사
2012년 8월: 고려대학교 정보보호대학원 석사
2012년 8월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 부채널 공격, 스마트 카드 보안, 암호시스템 안전성 분석 및 고속구현

조 성 민 (Sungmin Jo) 학생회원
2008년 2월: 광운대학교 수학과 학사 졸업
2011년 8월: 고려대학교 정보경영공학전문대학원 석사 졸업
2011년 8월~현재: 고려대학교 정보보호대학원 정보보호학과 박사과정
〈관심분야〉 부채널 공격, 공개키 암호 알고리즘, 암호구현

김 희 석 (HeeSeok Kim) 정회원
2006년: 연세대학교 수학과 학사
2008년: 고려대학교 정보보호대학원 공학석사
2011년: 고려대학교 정보보호대학원 공학박사
2011년 9월~2012년 12월: Bristol University 박사후 연구원
2013년~현재: 한국과학기술정보연구원 (KISTI) 과학기술정보보호실 선임연구원
〈관심분야〉 부채널 공격, 암호시스템 안전성 분석 및 고속구현, 암호칩 설계 기술, 보안관제, 네트워크 보안

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식