Journal of The Korea Society of Computer and Information
Vol. 17 No. 5 , May 2012

2012-17-5-3-4

오류주입 공격에 강건하며 병렬연산이 가능한 RSA-CRT

은 하 수*, 오 희 국*, 김 상 진**

Hardware Fault Attack Resistant RSA-CRT with Parallel Support

Hasoo Eun*, Heekuck Oh*, Sangjin Kim**

요 약

RSA-CRT는 RSA의 속도를 개선하기 위한 가장 대표적인 기법이다. RSA-CRT는 RSA에 사용되는 두 비밀소수의 법에서 각각 연산을 수행하기 때문에 RSA에 비해 약 4배가량 빠른 속도로 연산할 수 있다. 하지만 RSA에서 법 생성 후 바로 파기할 수 있었던 비밀 소수를 연산에 직접 사용함으로서 오류 주입공격 시 이를 노출하게 되는 문제가 있다. 이를 해결하기 위한 가장 대표적인 기법이 오류 확산에 기반을 둔 기법이다. 이 기법은 주입된 오류가 암호문 전체에 영향을 미치기 때문에 공격자가 비밀 소수를 얻기 힘들지만 독립적으로 진행되었던 연산을 순차적으로 해야 하며, 여전히 오류주입 공격에 취약하다는 문제점이 있다. 본 논문에서는 오류주입 공격에 강건하며 병렬처리가 가능하도록 공통법을 이용한 RSA-CRT 기법과 메시지를 각각의 법에서 연산한 RSA-CRT기법을 제안한다. 제안하는 기법은 최대 병렬연산을 통해 2회의 지수연산 시간밖에 소요되지 않기 때문에 빠른 연산속도를 제공하면서 오류주입 공격으로부터 비밀 소수의 노출을 보호할 수 있다.

▸ Keyword : RSA-CRT, 오류주입 공격, 병렬 연산

Abstract

RSA-CRT is one of the commonly used techniques to speedup RSA operation. Since RSA-CRT performs its operations based on the modulus of two private primes, it is about four times faster than RSA. In RSA, the two primes are normally thrown away after generating the public key pair.

• 제1저자 : 은하수 • 교신저자 : 김상진
• 투고일 : 2012. 03. 20, 심사일 : 2012. 04. 26, 게재확정일 : 2012. 05. 04.
* 한양대학교 컴퓨터공학과(Dept. of Computer Science and Engineering, Hanyang University)
** 한국기술교육대학교 컴퓨터공학부(School of Computer Science and Engineering, Korea University of Technology and Education)
※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2012-H0301-12-1002)
※ 이 논문은 2010년도 한국기술교육대학교 교수교육연구진흥비 지원에 의하여 연구되었음

However, in RSA-CRT, the two primes are directly used in RSA operations. This led to hardware fault attacks which can be used to factor the public modulus. The most common way to counter these attacks is based on error propagation. In these schemes, all the outputs of RSA are affected by the infected error which makes it difficult for an adversary to use the output to factor the public modulus. However, the error propagation has sequentialized the RSA operation. Moreover, these schemes have been found to be still vulnerable to hardware fault attacks. In this paper, we propose two new RSA-CRT schemes which are both resistant to hardware fault attack and support parallel execution: one uses common modulus and the other one perform operations in each prime modulus. Both proposed schemes takes about a time equal to two exponentiations to complete the RSA operation if parallel execution is fully used and can protect the two private primes from hardware fault attacks.

▸ Keyword : RSA-CRT, Fault injection, Parallel processing

I. 서 론

RSA는 대표적인 공개키 암호시스템으로 인수분해의 어려움에 기반하고 있다[1]. 1978년 제안된 이래로 보안과 관련된 많은 분야에서 사용되었고 현재까지 사용하고 있으나 속도가 느리다는 단점이 있다. 이를 개선하기 위해 다양한 기법이 제안되었으며, 이들 중 RSA-CRT라 불리는 중국인 나머지 정리(CRT, Chinese Remainder Theorem)를 이용한 기법이 가장 대표적이다.

1982년 Couvreur와 Quisquater가 제안한 RSA-CRT는 법 $n(=p \cdot q)$에서 연산하는 기존 RSA 시스템을 두 소수 법 p와 q로 나누어 연산한 후 CRT로 재결합한다[2]. 법이 작아지기 때문에 멱승을 빠르게 처리할 수 있으며, 이로 인해 빠른 연산을 가능케 한다. 현재 OpenSSL, Java, .Net 등에서는 라이브러리의 형태로 RSA-CRT를 지원하고 있으며 주로 서명과 복호화에 사용된다. 하지만 초기의 RSA-CRT는 오류 주입 공격에 취약한 문제점이 있었다. 오류주입 공격은 광학적 기법, 비정상 전원 인가 기법 등 시스템이 오동작 하는 상황을 유발하여 시스템을 공격하는 기법을 말하며, RSA-CRT에 오류주입 공격을 하면 비밀 소수 p 또는 q를 추출할 수 있다[11, 12].

Couvreur 등의 RSA-CRT는 법의 크기가 작은 것 이외에 s_p와 s_q를 병렬적으로 연산할 수 있다는 특징이 있었다. 하지만 2003년 Yen 등이 오류확산을 위해 s_p와 s_q 연산이 순차적으로 연산되는 형태를 제안한 이후 많은 기법들이 이와 같은 형태를 취하고 있다[7, 8, 9, 10]. 본 논문에서는 s_p와 s_q 연산을 독립적으로 연산함으로써 병렬 특성을 유지하고, 오류가 주입된 경우에도 이를 보완할 수 있는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 기존의 RSA 시스템과 CRT를 설명하고 CRT 재결합 기법을 서술한다. 3장에서는 2장에서 서술한 수식을 통해 기존 연구들이 오류주입 공격에 취약함을 분석하고, 4장에서 제안하는 RSA-CRT 기법을 소개한다. 5장에서는 제안하는 기법이 오류주입 공격에 안전함을 인자별로 보이고, 6장에서 결론을 맺는다.

II. 연구 배경

RSA-CRT는 빠른 속도를 제공한다는 장점이 있지만, 오류주입 공격에 취약하다는 단점 있다. 기존의 RSA는 법 n에서 연산이 이루어지기 때문에 p와 q를 유지할 필요가 없다. 하지만 RSA-CRT는 p와 q를 연산에 직접 사용하기 때문에 이 값들을 유지해야 하며, 오류주입 후 인수분해를 통해 소수를 얻을 수 있다는 문제점이 있다[3, 4]. 본 단락에서는 RSA와 CRT 및 CRT 재결합 기법을 수식을 통해 알아봄으로써 CRT 인자의 의미와 RSA-CRT의 인자 p와 q 유지의 필요성을 확인하고, 추후 분석을 위해 사용할 CRT 재결합 형태를 알아본다.

1. 표기법

본 논문에서는 관련 연구 및 제안하는 기법의 서술에 다음과 같은 표기법을 사용한다.

표 1. 표기법
Table 1. System Environment

표기	의미
p, q	서로소인 임의의 소수
r	임의의 정수
(e, n)	공개키
(d, n)	개인키
m	메시지
s	정상적인 서명
$\hat{s}$	오류 서명
t	오류에 의해 가감된 임의의 값
$CRT(x, y)$	x와 y의 CRT 재결합
$\phi(x)$	오일러 함수

2. 기존 RSA 시스템

기존 RSA 시스템의 키 생성은 다음과 같이 진행된다.

STEP 1. 두 소수 p와 q를 선택하여 다음을 계산한다.

$$n = p \cdot q,\ \phi(n) = (p-1)(q-1)$$

STEP 2. $\gcd(e, \phi(n)) = 1, e \in Z^*_{\phi(n)}$을 만족하는 임의의 e를 선택한다.

STEP 3. $ed \equiv 1 (\mathrm{mod}\phi(n)), d \in Z^*_{\phi(n)}$을 만족하는 임의의 d를 선택한다.

키 생성의 결과로 공개키 (e, n)과 개인키 (d, n)를 얻는다. STEP 1 이후 p와 q는 더 이상 사용되지 않으므로 파기해도 무방하다. 공개키 e로 생성한 암호문 c가 있을 때 RSA의 복호화는 c에 d승을 취함으로써 메시지를 얻게 된다.

3. 중국인 나머지 정리 (CRT, Chinese Remainder Theorem)

앞서 기술했듯이 중국이 나머지 정리는 연립 선형 합동식의 해를 구하기 위한 방법이다. 즉 서로소인 양의 정수 $n_1, \ldots, n_t$와 0이 아닌 정수 $a_1, \ldots, a_t$가 존재할 때 법 $n = n_1 n_2 \ldots n_t$에서 $x \equiv a_1 (\mathrm{mod} n_1), \ldots, x \equiv a_t (\mathrm{mod} n_t)$는 유일한 해를 갖는다. 중국인 나머지 정리의 연산 과정은 다음과 같다.

$$x = \sum_{j=1}^{t} \frac{n}{n_j} b_j a_j \tag{1}$$

$n_1, \ldots, n_t$는 서로소이며 n은 이들의 곱으로 구성되어 있다. 이때 법 n_j에서 n/n_j의 역수인 b_j가 있을 때 $b_j \cdot n/n_j$는 1과 합동이 되며 $b_j a_j \cdot n/n_j$는 법 n_j에서 a_j와 합동이 된다. 식(1)과 같이 $x = \sum_{j=1}^{t} \frac{n}{n_j} b_j a_j$가 존재할 때, $1 \le i \le t$를 만족하는 임의의 법 n_i에서 n_i를 포함한 모든 $b_j a_j \cdot n/n_j$는 0과 합동이 되므로 $x \equiv b_i a_i \cdot n/n_i (\mathrm{mod}\, n_i)$가 된다. 이때의 x는 a_i와 합동이며 선형 합동식의 해가 된다.

III. 관련연구

Lenstra[3]와 Boneh등은[6] Couvreur와 Quisquater에 의해 제안된 RSA-CRT[2]가 오류 주입 공격에 취약함을 증명하였다. 1997년 Shamir는 임의의 값 r을 이용하여 법 p, q에서 연산한 s_p, s_q를 공통된 법으로 변환하고, 이들의 비교를 통해 오류주입을 검사하는 방법을 제안하였다[5, 6]. 이에 대해 2003년 Yen 등은 안전한 $|r|$의 선택에 따른 성능 저하와 비교연산의 위험성을 들며 오류 확산에 기반을 둔 해결책을 제시하였다[7]. 하지만 2006년 동일한 저자에 의해 제안한 기법도 오류주입 공격에 취약함이 증명되었으며 이후 오류주입 공격에 대응하기 위한 여러 개선책들이 제안되었다[8, 9, 10]. 본 단락에서는 이와 같은 기존 연구들을 자세히 분석하며 제안하는 기법을 위한 요구사항을 도출한다.

1. Couvreur와 Quisquater의 RSA-CRT 기법[2]

RSA-CRT는 1982년 Couvreur와 Quisquater에 의해 처음으로 제안되었다. 메시지와 개인키를 두 법으로 나누어 각각 연산한 후 Garner의 CRT 재결합 기법을 이용하여 서명을 생성한다. 각각의 법에 대해서는 독립적으로 연산이 가능하며 CRT 재결합 시 각각 계산된 값을 이용하여 최종 서명을 생성한다.

$$m_p \equiv m (\mathrm{mod}\, p) \qquad m_q \equiv m (\mathrm{mod}\, q) \tag{7}$$

$$s_q \equiv m_q^{d_q} (\mathrm{mod}\, q) \equiv s (\mathrm{mod}\, q) \Rightarrow s = h \cdot q + s_q \tag{8}$$

$$s_p \equiv (h \cdot q + s_q) (\mathrm{mod}\, p) \tag{9}$$

$$h \equiv q_{inv} \cdot (s_p + p - s_q) (\mathrm{mod}\, p) \tag{10}$$

$$\therefore s = s_q + h \cdot q = s_q + [q_{inv} \cdot (s_p + p - s_q) (\mathrm{mod}\, p)] \cdot q \tag{11}$$

개인키를 두 법으로 나누는 것은 식(2)와 동일하다. 추가적으로 메시지를 각각의 법에서 계산한 후 개인키로 멱을 취하면 s_p, s_q값을 얻을 수 있으며 이들은 자신이 속한 법에서 s

값과 합동이다. s_q는 식(8)과 같이 표현할 수 있으며 이를 법 p에 대해 계산하면 s_p를 얻을 수 있다. s_q를 이항해서 s_p와 뺄 때 $s_q > s_p$ 인 경우 음수가 되므로, 양수로 변환하기 위하여 p를 더한다. 식(10)을 식(8)에 대입하면 Garner의 CRT 재결합 기법의 최종 형태인 식(11)을 얻는다.

2. Couvreur와 Quisquater의 RSA-CRT 기법에 대한 오류주입 공격

Lenstra와 Boneh 등은 최초 제안된 RSA-CRT가 오류 주입 공격에 취약함을 보였다. 본 단락에서는 제안된 공격 방법들을 소개하고 이러한 공격이 가능한 이유에 대해 분석한다. 시기적으로 Lenstra의 공격 방법이 먼저 제안되었으나 이해의 편의를 돕기 위하여 Boneh 등의 공격 방법부터 소개한다.

2.1. 선택 암호문 공격[6]

Boneh 등은 공격자가 동일한 메시지의 서명과 오류 서명을 모두 가지고 있는 경우 오류주입 공격이 가능함을 보였다. 이 공격 방법은 정상적인 서명과 오류 서명의 차를 이용하여 소수 값을 추출하는 것으로, 식(6)을 이용하여 다음과 같이 전개할 수 있다.

$$s \equiv \left(q^{p-1} \bullet s_p + p^{q-1} \bullet s_q\right) (\mathrm{mod}\ n) \tag{12}$$

$$\hat{s} \equiv \left(q^{p-1} \bullet \hat{s}_p + p^{q-1} \bullet \hat{s}_q\right) (\mathrm{mod}\ n) \tag{13}$$

만일 s_p를 계산할 때만 오류가 발생하였다고 가정하면 $s_q = \hat{s}_q$이므로 $s - \hat{s}$는 식(14)와 같이 계산된다.

$$s - \hat{s} \equiv \left[q^{p-1}(s_p - \hat{s}_p)\right] (\mathrm{mod}\ n) \tag{14}$$

이때 $p \nmid (s_p - \hat{s}_p)$라면 식(15)와 같이 소수 q를 얻을 수 있다.

$$\gcd(s - \hat{s}, n) = \gcd([q^{p-1}(s_p - \hat{s}_p)] (\mathrm{mod}\ n), p \bullet q) = q \tag{15}$$

위 공격 방법은 정상적인 서명과 오류서명 사이의 차를 이용하여 p로 묶인 항을 제거한다. 결과 값에 q로 묶인 항만 남으므로 최대공약수를 통해 q를 얻을 수 있다. $p | (s_p - \hat{s}_p)$인 경우 최대공약수 값이 n이 나오게 되므로 소수를 추출할 수 없지만 그 확률이 매우 낮다.

2.2. 선택 평문 공격[3]

Lenstra는 알고 있는 메시지와 그에 대한 오류 서명을 가지고 있을 경우 소수 값을 얻을 수 있음을 보였다. Ⅲ.2.1과 같은 공격 환경에서 오류 서명을 얻고 s_p를 계산할 때만 오류가 발생하였다고 가정한다.

$$m - \hat{s}^e = s^e - \hat{s}^e$$

$$= (s - \hat{s})(\sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}) \tag{16}$$

식(16)의 $(s - \hat{s})$는 식(14) ~ 식(15)과 같이 유도 가능하며 다음과 같이 표현할 수 있다.

$$q | (s - \hat{s}) \wedge p \nmid (s - \hat{s}) \tag{17}$$

식(16)의 $\sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}$는 밑이 p와 q인 항들의 덧셈으로 이루어져 있으므로 p 또는 q로 나누어떨어지지 않는다. 이것을 식으로 표현하면 다음과 같다.

$$\left\{q \nmid \sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}\right\} \wedge \left\{p \nmid \sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}\right\} \tag{18}$$

식(17)과 식(18)을 p와 q의 경우로 나누어보면 다음과 같은 결과를 얻을 수 있다.

$$\{p \nmid (s - \hat{s})\} \wedge \left\{p \nmid \sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}\right\} \Rightarrow p \nmid (m - \hat{s}^e) \tag{19}$$

$$\{q | (s - \hat{s})\} \wedge \left\{q \nmid \sum_{i=1}^{e} s^{e-i} \hat{s}^{i-1}\right\} \Rightarrow q | (m - \hat{s}^e) \tag{20}$$

$$\therefore \gcd(m - \hat{s}^e, n) = q \tag{21}$$

3. 비교 연산을 통한 RSA-CRT의 오류주입 대응 기법[5, 6]

1997년 Shamir는 3.2와 같은 오류 주입 공격에 대응하기 위해 s_p와 s_q를 비교하는 기법을 제안하였다. 서로 법이 다른 s_p와 s_q를 비교하기 위하여 임의의 값 r을 선택하고 s_p와 s_q를 법 r의 값으로 변환한다. 비교의 결과 두 값이 같다면 오류가 없다고 판단하고 각각의 법에서 계산한 값을 CRT 재결합하여 서명 s를 구한다. 본 단락에서는 Shamir의 기법을 소개하며 해당 기법이 법 r에서 같음을 보이고, 연산의 결과가 기존의 CRT와 같은 값을 가짐을 수식을 통해 설명한다.

$$\begin{aligned} s_p &\equiv m^{d(\mathrm{mod}\,\phi(pr))} (\mathrm{mod}\ pr) \\ &= m^{\phi(pr) \bullet i_p + d} (\mathrm{mod}\ pr) \\ &\equiv m^d (\mathrm{mod}\ pr) \\ &= (pr) \bullet j_p + m^d \\ &= (pj_p) \bullet r + m^d \\ &\equiv m^d (\mathrm{mod}\ r) \end{aligned} \tag{22}$$

사용자는 임의의 r을 선택하여 지수의 법과 서명의 법에 각각 곱한다. 이 값을 수식으로 풀면 오일러함수와 곱해진 부분은 소거되고 d승 부분만 남게 된다. 이 수식을 법 r에 대한 식으로 표현하면 두 값이 법 r에서 합동임을 알 수 있다. 따라서 두 곳 모두 오류가 나지 않으면 $s_p \equiv s_q \pmod r$을 만족하게 된다. 이 상태를 오류가 없다고 정의하며 식(23)과 같이 CRT 재결합을 계산한다.

$$s = CRT(s_p \pmod p, s_q \pmod q) \tag{23}$$

인자의 생성 과정에서 s_p와 s_q는 각각 법 pr, qr에서 생성되었지만 식(22)와 같은 방식으로 변환하면 각각의 법에서 계산한 값과 같아지므로 식(23)의 값은 기존 RSA-CRT와 동일한 값을 갖는다.

4. 비교 연산을 통한 대응 기법의 문제점과 오류 확산에 기반을 둔 대응 기법[7]

2003년 Yen 등은 Shamir의 기법이 비교연산을 사용하기 때문에 오류 주입공격에 취약하다는 것과 안전한 $|r|$ 선택에 따른 효율성 저하가 생긴다는 문제를 제기하였다. 비교연산의 결과는 True/False 둘 중 하나를 취하게 되며 이를 회로 상에 구현하게 되면 1bit Flip-flop을 사용하게 된다. 따라서 여기에 오류를 주입하면 1/2의 확률로 비교연산을 무시할 수 있기 때문에 초기 RSA-CRT와 동일한 문제가 발생한다. 또한 $|r|$이 작으면 전사공격에 취약해지므로 적당한 길이의 r을 선택해야 하지만, r의 크기가 커지면 그만큼 연산 비용이 증가하게 되는 문제가 있다. 이에 대하여 Yen 등은 오류 확산 기법에 기반을 둔 두 가지 대응책을 제시하였다. 이 둘은 안전성과 공격 방법이 동일하여 편의상 하나만 소개한다.

$$d_r = d - r \quad \gcd(d_r, \phi(n)) = 1 \quad e_r \equiv d_r^{-1} \pmod{\phi(n)} \tag{24}$$

$$k_p = \left\lfloor \frac{m}{p} \right\rfloor \qquad k_q = \left\lfloor \frac{m}{q} \right\rfloor \tag{25}$$

$$s_p \equiv m^{d_r \pmod{\phi(p)}} \pmod p \tag{26}$$

$$m' \equiv \left[s_p^{e_r} \pmod p + k_p \cdot p \right] \pmod q \tag{27}$$

$$s_q \equiv m'^{d_r \pmod{\phi(q)}} \pmod q \tag{28}$$

$$m'' = s_q^{e_r} \pmod q + k_q \cdot q \tag{29}$$

$$s = CRT(s_p, s_q) \cdot m''^r \tag{30}$$

x와 y를 p와 q로 나누었을 때의 나머지라 하면 $k_p \cdot p$ 와 $k_q \cdot q$는 다음과 같이 표현할 수 있다.

$$m = k_p \cdot p + x \qquad k_p \cdot p = m - x \equiv m - (m \bmod p) \tag{31}$$

$$m = k_q \cdot q + y \qquad k_q \cdot q = m - y \equiv m - (m \bmod q) \tag{32}$$

이들은 각각의 법에서 m값을 제거하는데 사용된다. 식(27)은 식(4)와 같은 이유로 내부 인자를 $s_p \equiv m^{d_r} \pmod p$과 같이 표현할 수 있다. 이 값과 식(31)의 $k_p \cdot p$를 식(30)에 대입하면 다음과 같은 결과를 얻을 수 있다.

$$\begin{aligned} m' &\equiv \left[m^{d_r \cdot e_r} \pmod p + k_p \cdot p \right] \pmod q \\ &\equiv \left[m \pmod p + m - (m \bmod p) \right] \pmod q \\ &\equiv m \pmod q \end{aligned} \tag{33}$$

위 식의 결과를 식 (28)에 대입하면 s_q는 다음과 같다.

$$s_q \equiv m^{d_r \pmod{\phi(q)}} \pmod q \equiv m^{d_r} \pmod q \tag{34}$$

따라서 s_p와 s_q는 CRT 재결합이 가능하다. 이 값을 식(29)에 대입하면 법 q에서의 m값이 소거되며 $m'' = m$이 된다. 이 값을 식(30)에 대입하면 다음과 같이 정리되며 m^r이 소거되어 기존의 RSA-CRT와 동일한 값을 얻을 수 있다.

$$\begin{aligned} s_p &\equiv m^{d_r} \pmod p \equiv m^{d-r} \pmod p \\ s_q &\equiv m^{d_r} \pmod q \equiv m^{d-r} \pmod q \\ s &= CRT(s_p, s_q) \cdot m''^r \\ &= \left[q^{p-1} \cdot m^{d_r} \pmod p + p^{q-1} \cdot m^{d_r} \pmod q \right] \pmod n \cdot m^r \\ &= \left[q^{p-1} \cdot m^d \pmod p + p^{q-1} \cdot m^d \pmod q \right] \pmod n \end{aligned} \tag{35, 36}$$

이 기법은 오류 확산을 이용하여 s_p 또는 s_q에 오류가 주입된 경우 서명 값 생성에 영향을 주어 소수 추출을 막는다. 하지만 기존과는 다르게 이 기법은 $s_p \rightarrow m' \rightarrow s_q \rightarrow m'' \rightarrow s$의 순서대로 순차적으로 처리해야 하기 때문에 서명 생성 시 s_p와 s_q를 병렬적으로 처리할 수 없다. 따라서 기존의 기법에 비해 안전성은 높아졌지만 수식의 복잡도가 증가했고, 선형적으로 처리해야 값을 얻을 수 있게 되어 효율성이 낮아졌다.

5. 오류 확산에 기반을 둔 대응 기법의 문제점[8]

이 문제점은 2006년 FDTC에서 동일 저자에 의해 발표된

것으로 s_p와 s_q 이외의 인자에 오류를 주입했을 때도 소수 추출의 위험이 있음을 보여주고 있다. 공격의 방법은 다음과 같다.

Ⅲ.4의 기법에서 k_q 계산 시 오류가 발생했다고 가정하고 t만큼 변화가 생겼을 때 이를 k_q'이라 하면 다음과 같이 표현할 수 있다.

$$k_q' = k_q + t \tag{37}$$

k_q'를 이용하여 생성한 m''을 m'''이라 하면 다음과 같이 전개된다.

$$\begin{aligned} m''' &= (s_q^{e_r} \bmod q) + k_q' \cdot q \\ &= (s_q^{e_r} \bmod q) + (k_q + t) \cdot q \\ &= (s_q^{e_r} \bmod q) + k_q \cdot q + t \cdot q \\ &= m'' + t \cdot q \end{aligned} \tag{38}$$

식(38)의 값을 이용하여 오류 서명을 만들면 기존 서명 값과 소수 q를 포함한 항이 덧셈으로 연결된 결과를 얻을 수 있다. 여기서 a_i $(0 \le i \le r)$의 값은 파스칼 삼각형 $r+1$번째 줄의 i번째 값이다.

$$\begin{aligned} \hat{s} &= CRT(s_p, s_q) \cdot m'''^r \pmod n \\ &= CRT(s_p, s_q) \cdot (m'' + t \cdot q)^r \pmod n \\ &= CRT(s_p, s_q) \cdot (\sum_{i=0}^{r} a_i m''^{r-i} (tq)^i) \pmod n \\ &= CRT(s_p, s_q) \cdot m''^r + CRT(s_p, s_q) \cdot (\sum_{i=1}^{r} a_i m''^{r-i} (tq)^i) \pmod n \\ &= m^d + CRT(s_p, s_q) \cdot \frac{m''^r}{m''^r} \cdot (\sum_{i=1}^{r} a_i m''^{r-i} (tq)^i) \pmod n \\ &= m^d + \frac{m^d}{m''^r} \cdot (\sum_{i=1}^{r} a_i m''^{r-i} (tq)^i) \pmod n \end{aligned} \tag{39}$$

$x = \frac{m^d}{m''^r} \cdot (\sum_{i=1}^{r} a_i m''^{r-i} t^i q^{j-1}) \pmod n$ 라고 하면...

$$\hat{s} = m^d + x \cdot q \tag{40}$$

위와 같이 생성한 오류 서명을 공개키로 복호화한 값과 메시지와 차를 이용하여 다음과 같이 소수 q를 얻을 수 있다.

$$\begin{aligned} &\gcd(\hat{s}^e - m, n) = \gcd((m^d + x \cdot q)^e - m, n) \\ &= \gcd(\sum_{j=0}^{e} b_j m^{d \cdot (e-j)} (xq)^j - m, n) \\ &= \gcd(m - m + q(\sum_{j=1}^{e} b_j m^{d \cdot (e-j)} x^j q^{j-1}), n) \\ &= \gcd(q(\sum_{j=1}^{e} b_j m^{d \cdot (e-j)} x^j q^{j-1}), n) = q \end{aligned} \tag{41}$$

식(41)의 전개 과정은 식(39)와 유사하며 이들 중 메시지를 제거하여 q로 묶인 항만 남도록 만든다. 이 값과 n의 최대 공약수를 구하면 $p \nmid (\sum_{j=1}^{e} b_j m^{d \cdot (e-j)} x^j q^{j-1})$일 경우, q를 얻을 수 있다.

Ⅳ. 제안하는 기법

본 장에서는 이들의 분석에 도출한 목표사항을 정의하고, 오류주입 공격에 강건하며 병렬연산이 가능한 RSA-CRT를 제안하고자 한다. 제안하는 기법은 CRT에 곱셈의 항등원을 곱하되, 항등원 생성 시 s_p와 s_q를 모두 사용한다. 이를 통해 주입된 오류가 서명 전체에 영향을 주도록 설계한다. 정상적인 서명의 경우 항등원은 1이 되어 CRT만 남도록 하며, 오류 주입이 된 서명은 소수 이외에 잔여 값을 남김으로써 소수를 추출할 수 없도록 설계한다.

본 논문에서는 이와 같은 목표 하에 두 가지 형태의 RSA-CRT기법을 제안한다. 먼저 제안하는 RSA-CRT-PR은 Shamir의 기법을 응용하여 s_p와 s_q가 공통된 법에서 연산하되 오류가 서명에 확산되도록 한 것으로 비교연산을 없애고 위와 같은 조건을 만족시키기 위해 설계되었으나 법의 크기가 커져서 연산 효율이 낮다는 단점이 있다. 이를 개선한 RSA-CRT-PD는 공통된 법을 위해 지수를 늘리는 것이 아니라 같은 법에서 계산한 메시지와 서명을 복호화한 값의 차를 이용하여 오류를 검출한다. 이 기법은 위와 동일한 조건을 만족하면서 Yen 등의 기법에 비해 2배 이상 빠른 연산 시간을 보장한다.

1. 설계 목표

제안하는 기법의 설계 목표는 오류주입 공격에 강건하면서 병렬 연산이 가능한 RSA-CRT를 설계하는 것이다. 이를 위하여 제안하는 기법은 다음과 같은 사항을 만족해야 한다.

- **오류 주입을 통한 소수 추출 불가:** 기 제안된 기법들에 대한 공격 방법은 정상적인 서명 또는 메시지와 오류 서명의

차이를 이용하여 하나의 소수 항을 제거하고 n과 최대공약수를 구하여 소수를 추출한다. 이를 위해 공격받은 수식은 소수 p 또는 q로 묶인 단항식으로 정리되어야 한다. 본 논문에서는 오류 확산 기법을 응용하여 주입된 오류가 서명값에 영향을 주어 최대공약수를 통한 소수 추출을 막을 수 있도록 설계 하고자 한다.

- **s_p와 s_q의 독립적 연산:** 초기 RSA-CRT는 s_p와 s_q를 독립적으로 연산할 수 있기에 병렬 연산을 할 수 있었다. 하지만 오류 주입 공격을 막기 위해 오류 확산 기법이 제안되면서 s_q생성 시 인자로 s_p를 사용하게 되었고 병렬 연산이 불가능해 졌다. 본 논문에서는 s_p와 s_q의 연산을 독립적으로 연산 후 재결합 하여 병렬 연산이 가능하게 하되 오류 주입에도 강건하도록 설계하고자 한다.
- **연산의 단순화:** 초기 RSA는 법 n에서 메시지 m에 d승을 취해 서명을 생성하였다. RSA-CRT는 이 연산을 작은 두 법으로 나누고 멱의 크기를 줄여 빠르게 연산하고자 제안되었다. 하지만 오류 주입 공격에 대한 문제가 제기 되면서 이를 막기 위한 여러 기법이 제안되었고 연산 시간이 증가했다. 본 논문에서는 기 제안된 기법에 비하여 연산 시간은 단축시키고 오류 주입 공격을 막을 수 있는 RSA-CRT 기법을 설계하고자 한다.

2. RSA-CRT-PR

제안하는 첫 번째 기법은 임의의 값 r을 사용하여 공통된 법을 만드는 방법이다. s_p와 s_q에서 각각 r에서 합동인 s를 만든 후 이 둘을 이용하여 곱셈의 항등원을 생성한다. 이 기법은 III.3의 기법과 사용하는 인자는 동일하지만 비교연산이 없으며, s_p와 s_q의 값이 서명에 직접 영향을 주어 오류가 서명으로 확산되도록 만든다. 이를 위해 임의의 값 r을 사용한다. 생성 단계는 식(42)의 형태로 이루어진다.

$$d_p \equiv d(\bmod \phi(pr)) \qquad d_q \equiv d(\bmod \phi(qr)) \tag{42}$$

이후 사용자로부터 받은 메시지 m으로 각각의 법에서 서명하여 s_p와 s_q를 만든다.

$$s_p = m^{d_p}(\bmod pr) \qquad s_q = m^{d_q}(\bmod qr) \tag{43}$$

$$s_p \equiv s_q (\bmod r) \tag{44}$$

$$\therefore (s_p(\bmod r) - s_q(\bmod r) + 1) = 1 \tag{45}$$

식(43)의 두 값은 식(22)에서 보인바와 같이 법 r에서 합동이다. 이 두 값을 이용하여 식(45)과 같이 곱셈의 항등원을 만들고 이를 $CRT(s_p(\bmod p), s_q(\bmod q))$에 곱하여 서명을 생성한다.

$$\begin{aligned} s &= CRT(s_p(\bmod p), s_q(\bmod q)) \cdot (s_p(\bmod r) - s_q(\bmod r) + 1) \quad (46) \\ &= CRT(s_p(\bmod p), s_q(\bmod q)) \end{aligned} \tag{47}$$

이 기법은 메시지에서 각 법에 대한 서명을 생성하는 과정이 독립적이므로 병렬적으로 연산 가능하다. 서명은 독립적으로 생성된 s_p와 s_q각각의 결과를 이용하여 항등원을 만들고 $CRT(s_p(\bmod p), s_q(\bmod q))$와 곱하여 생성한다. 수학적인 의미에서 곱셈의 항등원과의 곱셈은 항상 자기 자신과 같은 값을 반환하지만, 제안하는 기법에서는 정상적인 경우 서명을 얻고, 오류가 주입된 경우 전혀 다른 값을 반환하며, 소수 추출을 막는 역할을 한다. 제안하는 기법에 오류주입을 통한 선택 암호문 공격 또는 선택 평문 공격을 하는 경우 별도의 상수항이 존재하게 되어 인수분해로 소수 값을 얻을 수 없다.

3. RSA-CRT-PD

RSA-CRT-PR은 공통된 법을 생성하기 위하여 r을 사용하였다. 본 단락에서는 r을 사용하지 않고 동일한 효과를 낼 수 있는 기법을 제안한다. 제안하는 기법은 서명 값에서 다시 메시지를 추출한 후 각각의 법에서 계산한 메시지 값 m_p와 m_q를 뺀다. 키는 초기 RSA-CRT와 동일하게 법 $\phi(p)$와 $\phi(q)$에서 각각 생성한다.

$$d_p \equiv d(\bmod \phi(p)) \qquad d_q \equiv d(\bmod \phi(q)) \tag{48}$$

$$e_p \equiv d_p^{-1}(\bmod \phi(p)) \qquad e_q \equiv d_q^{-1}(\bmod \phi(q)) \tag{49}$$

이후 사용자로부터 받은 메시지 m으로 각각의 법에서 서명하여 m_p, m_q, s_p, s_q를 만든다.

$$m_p \equiv m(\bmod p) \qquad m_q \equiv m(\bmod q) \tag{50}$$

$$s_p \equiv m^{d_p}(\bmod p) \qquad s_q \equiv m^{d_q}(\bmod q) \tag{51}$$

식(50)은 메시지를 각각의 법에 맞도록 변환하는 과정이며 식(51)은 각각의 법에서 메시지에 서명을 하는 과정이다. 이 때 m_p의 값이 s_p의 생성에 사용되면 m_p에 오류를 주입하는 경우 $s_p^{e_p}$에서 오류가 주입된 m_p와 같은 값을 반환하기 때문에 m_p와 s_p, m_q와 s_q의 생성과정을 분리하여 진행해야 한다.

$$s_p^{e_p}(\bmod p) \equiv m^{d_p \cdot e_p}(\bmod p) \equiv m(\bmod p)$$

$$s_q^{e_q}(\bmod q) \equiv m^{d_q \cdot e_q}(\bmod q) \equiv m(\bmod q) \quad (52)$$

$$\therefore ((s_p^{e_p}(\bmod p) + s_q^{e_q}(\bmod q)) - (m_p + m_q) + 1) = 1 \quad (53)$$

$$s = CRT(s_p, s_q) \cdot \left((s_p^{e_p}(\bmod p) + s_q^{e_q}(\bmod q)) - (m_p + m_q) + 1\right) \quad (54)$$

$s_p^{e_p} \equiv m(\bmod p), s_q^{e_q} \equiv m(\bmod q)$ 이므로 식(53)의 값은 1이 된다. RSA-CRT-PR과 마찬가지로 메시지에서 각 법에 대한 서명을 생성하는 과정이 독립적이므로 병렬적으로 연산 가능하다. 독립적으로 생성된 s_p와 s_q 각각의 결과를 이용하여 항등원을 만들고 서명 생성 시에 곱하기 때문에 연산에 주입된 오류가 확산되어 서명에 영향을 미치게 된다. 생성한 항등원을 $CRT(s_p, s_q)$에 곱하면 정상적으로 서명된 경우 식(53)의 값이 1이 되지만, 오류가 주입된 경우 전혀 다른 값을 주며 소수 추출을 막는 역할을 한다. 제안하는 기법에 오류주입을 통한 선택 암호문 공격 또는 선택 평문 공격을 하는 경우 별도의 상수항이 존재하게 되어 인수분해로 소수 값을 얻을 수 없다.

V. 분석

이 장에서는 제안하는 기법이 오류 주입 공격에 안전함을 분석하고, 기 제안된 기법들과 효율성을 분석한다. 안전성 분석은 수식에 사용되는 각각의 인자에 오류를 주입하여 선택 암호문 공격과 선택 평문 공격을 가한다. 이 식을 전개 후 분석하여 제안하는 기법이 오류주입 공격에 강건함을 보인다. 효율성 분석은 각각 인자의 크기를 가정하고 연산했을 때 지수 연산의 크기와 연산 횟수를 비교한다.

1. 안전성 분석

현재 제안된 공격 방법은 광학적 기법, 비정상 전원 인가 기법 등이 제안되었지만, 해당 공격을 통해 오류를 조절할 수 있는지에 대한 분석은 되어있지 않다[11, 12]. 본 논문에서는 이러한 상황을 고려하여 공격자가 주입하는 오류를 조작해 원하는 값을 만들 수 없는 것으로 가정한다. 또한 RSA-CRT는 시스템에 p와 q 등의 비밀 값을 저장하기 위하여 조작 방지 하드웨어(TRH, Tamper Resistant Hardware)를 사용한다. 따라서 이와 같은 시스템 환경에서 공격자가 임의로 내부 인자에 접근하여 값을 획득할 수 없는 것으로 가정한다. 이상의 내용을 기반으로 다음과 같은 시스템 가정을 도출할 수 있다.

(시스템 가정 1) 오류 주입 공격을 받은 인자는 임의의 값으로 변한다.

(시스템 가정 2) 공격자는 시스템 내의 인자에 임의로 접근할 수 없다.

본 단락에서는 위와 같은 시스템 가정 하에 제안하는 기법을 선택 암호문 공격과 선택 평문 공격으로 나누어 분석한다.

1.1 RSA-CRT-PR에 대한 선택 암호문 공격

$s_p \equiv m^{d_p}(\bmod pr)$ 연산 시 주입된 오류를 t라고 하면, $s - \hat{s}$은 다음과 같이 전개된다.

$$\begin{aligned} & s - \hat{s} \\ & \equiv s - CRT(\hat{s}_p(\bmod p), s_q(\bmod q)) \cdot (\hat{s}_p(\bmod r) - s_q(\bmod r) + 1) \\ & \equiv s - ((s + q^{p-1}t(\bmod p))(\bmod n)) \cdot (t(\bmod r) + 1) \\ & \equiv -q^{p-1}t(\bmod p) \cdot (t(\bmod r) + 1) - s \cdot t(\bmod r) \end{aligned} \quad (55)$$

식(55)의 $CRT(s_p(\bmod p), s_q(\bmod q))$는 식(9)에서 보인 바와 같이 $(q^{p-1}s_p(\bmod p) + p^{q-1}s_q(\bmod q))$로 표현할 수 있으며, $(s_p(\bmod r) - s_q(\bmod r) + 1)$ 은 앞서 설명한 바와 같이 합동일 경우 1이 된다. 최종적으로 p항이 제거되었으나, 공격자가 서명 값 s를 알아도 시스템 가정 2에 의해 t 값을 알 수 없으며, 임의의 값인 r 또한 알 수 없으므로 $-s \cdot t(\bmod r)$을 제거할 수 없다. 상수항이 남아있으므로 $s - \hat{s}$는 소수 q의 배수가 되지 않으며 그 결과 최대공약수를 통해 소수 값을 추출할 수 없다. $s_q(\bmod qr)$에 대해서도 이와 동일하게 식이 전개된다.

1.2 RSA-CRT-PR에 대한 선택 평문 공격

$s_p \equiv m^{d_p}(\bmod pr)$ 연산 시 오류를 주입하여 선택 평문 공격을 하는 경우 다음과 같이 식이 전개된다. 이때 상수 i와 j의 범위는 $0 \le i, j \le e$이다.

$$\begin{aligned} & \hat{s}^e \\ & \equiv (CRT(\hat{s}_p(\bmod p), s_q(\bmod q)) \cdot (\hat{s}_p(\bmod r) - s_q(\bmod r) + 1))^e \\ & \equiv CRT(\hat{s}_p(\bmod p), s_q(\bmod q))^e \cdot (\hat{s}_p(\bmod r) - s_q(\bmod r) + 1)^e \\ & \equiv (q^{p-1}(s_p + t)(\bmod p) + p^{q-1}s_q(\bmod q))(\bmod n)^e \\ & \quad \cdot ((s_p + t)(\bmod r) - s_q(\bmod r) + 1)^e \\ & \equiv (m^d + q^{p-1}t(\bmod p))^e(\bmod n) \cdot (t(\bmod r) + 1)^e \\ & \equiv \left(\sum_{i=0}^{e} a_i m^{d \cdot (e-i)} (q^{p-1}t(\bmod p))^i\right) \cdot \left(\sum_{j=0}^{e} b_j (t(\bmod r))^{e-j}\right) \end{aligned}$$

$$\equiv m(\sum_{j=0}^{e} b_j (t \bmod r))^{e-j})$$

$$+q(\sum_{j=0}^{e} a_j m^{d \cdot (e-j)} q^{p-2} t^j \bmod p)) \cdot (\sum_{j=0}^{e} b_j (t \bmod r))^{e-j}) \quad (56)$$

식(56)은 식(55)와 같이 $\hat{s}_p = s_p + t$를 대입하여 생성한 식이다. 이를 인수분해 공식에 따라 전개한 후, m과 q^{p-1}로 묶는다. 파스칼 삼각형의 모든 줄의 첫 항과 끝항은 1이므로 $a_0 = a_e = 1$가 되고 끝항과 곱해진 m이 $-m$에 의해 소거된다. 하지만 mt와 어떤 수의 곱으로 이루어진 상수항이 남는다. 공격자는 $\sum_{j=0}^{e} b_j t^{e-j-1} \pmod r$을 알 수 없으므로 메시지를 알고 있어도 소수를 추출 할 수 없다. $s_q \pmod{qr}$에 대해서도 이와 동일하게 식이 전개된다.

1.3 RSA-CRT-PD에 대한 선택 암호문 공격

$s_p \equiv m^{d_p} \pmod p$ 연산 시 오류를 주입하여 선택 암호문 공격을 하는 경우 다음과 같이 식이 전개된다. 이때 상수 i의 범위는 $0 \le i \le e_p$이다.

$$s - \hat{s}$$
$$\equiv s - CRT(\hat{s}_p, s_q)$$
$$\cdot (\hat{s_p}^{e_p} \bmod p) + s_q^{e_q} \bmod q)) - (m_p + m_q) + 1)$$
$$\equiv s - (s + q^{p-1}(t \bmod p)) \bmod n)$$
$$\cdot ((s_p + t)^{e_p} \bmod p) - m_p + 1)$$
$$\equiv s - s((\sum_{i=0}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p) - m_p + 1)$$
$$-((q^{p-1} t \bmod p)) \bmod n))$$
$$\cdot ((\sum_{i=0}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p) - m_p + 1)$$
$$\equiv -s(m \bmod p) - m_p + (\sum_{i=1}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p))$$
$$-((q^{p-1} t \bmod p)) \bmod n))$$
$$\cdot ((\sum_{i=0}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p) - m_p + 1)$$
$$\equiv -st(\sum_{i=1}^{e_p} a_i s_p^{e_p - i} t^{i-1}) \bmod p)$$
$$-((q^{p-1} t \bmod p)) \bmod n)) \cdot ((\sum_{i=1}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p) + 1) \quad (57)$$

$s_p \equiv m^{d_p} \pmod p$, $s_q \equiv m^{d_q} \pmod q$ 이므로 e_p와 e_q를 각각 멱으로 취하면 각 법에서의 m값을 얻게 된다. 이 값은 뒤에 더해진 m_p와 m_q에 의해 소거된다. $\hat{s}_p = s_p + t$ 라 하고 식을 전개하면 정상적인 서명 s는 제거되고 q항과 st항만 남게 된다.

$q \mid (-st(\sum_{i=1}^{e_p} a_i s_p^{e_p - i} t^{i-1}) \pmod p)$ 라면 소수를 추출 수 있지만 이와 같을 확률은 $\frac{1}{q}$로 소수를 찾기 위해 전사공격을 하는 것과 같은 확률이다. $s_q \equiv m^{d_q} \pmod q$에 오류가 주입된 경우도 위와 같은 이유로 소수 추출이 어렵다.

$m_p \equiv m \pmod p$ 연산 시 오류를 주입하여 선택 암호문 공격을 하는 경우 다음과 같이 식이 전개된다.

$$s - \hat{s} \equiv s - s \cdot (s_p^{e_p} \bmod p) + s_q^{e_q} \bmod q)) - (\widehat{m_p} + m_q) + 1)$$
$$\equiv s(1 - (s_p^{e_p} \bmod p) - s_q^{e_q} \bmod q) + (m_p + t) + m_q - 1)$$
$$\equiv st \quad (58)$$

s_p의 생성에 m_p가 아닌 입력 값 m을 그대로 사용하므로 m_p에 주입된 오류는 s_p에 영향을 주지 않는다. 따라서 $\hat{s}$의 CRT연산은 정상적으로 수행된다. 오류 주입으로 인한 m_p의 변화량을 t라고 할 때 s_p, s_q를 대입하면 지수가 1이 되면서 m_p, m_q와 함께 소거된다. 식(58)는 소거되는 값들을 정리한 결과이며 이 식에서는 소수를 추출할 수 없다. $m_q \equiv m \pmod q$에 오류가 주입된 경우도 동일하게 식이 적용된다.

1.4 RSA-CRT-PD에 대한 선택 평문 공격

$s_p \equiv m^{d_p} \pmod p$ 연산 시 오류를 주입하여 선택 평문 공격을 하는 경우 다음과 같이 식이 전개된다.

$$\hat{s}^e - m$$
$$\equiv CRT(\hat{s}_p, s_q)^e$$
$$\cdot (\hat{s}_p^{e_p} \bmod p) + s_q^{e_q} \bmod q)) - (m_p + m_q) + 1)^e - m$$
$$\equiv (q^{p-1}(s_p + t) \bmod p) + p^{q-1} s_q \bmod q))^e$$
$$\cdot ((s_p + t)^{e_p} \bmod p) - m_p + 1)^e - m$$
$$\equiv (s + q^{p-1} t \bmod p))^e \cdot ((\sum_{i=0}^{e_p} a_i s_p^{e_p - i} t^i) \bmod p) - m_p + 1)^e - m$$
$$\equiv (\sum_{j=0}^{e} b_j s^{e-j} (q^{p-1} t \bmod p))^j)$$
$$\cdot ((t(\sum_{i=1}^{e_p} a_i s_p^{e_p - i} t^{i-1})) \bmod p) + 1)^e - m$$

$$k=(t(\sum_{i=1}^{e_p}a_i s_p^{e_p-i}t^{i-1}))(\mathrm{mod}\ p)\text{라 하면}$$

$$\equiv q^{p-1}(\sum_{j=1}^{e}b_j s^{e-j}q^{(p-1)\cdot(j-1)}t^j(\mathrm{mod}\ p))\cdot(k+1)^e$$
$$+m(k+1)^e-m$$
$$\equiv q^{p-1}(\sum_{j=1}^{e}b_j s^{e-j}q^{(p-1)\cdot(j-1)}t^j(\mathrm{mod}\ p))\cdot(k+1)^e$$
$$+m\sum_{i=0}^{e-1}c_i k^{e-i} \quad (59)$$

s_p에 대한 오류는 재결합 부분과 항등원 부분 두 곳에 모두 영향을 준다. $\hat{s}_p\equiv s_p+t$라 하자. b_0와 b_e는 1이고 $s^e=m$이므로 이를 대입하고 m과 q^{p-1}로 식을 나눈다.

$m(k-1)^e=\sum_{i=0}^{e}c_i m k^{e-i}$이며 $c_0=c_e=1$이므로 $-m$이 소거되고 남은 부분을 m으로 묶는다. 결과 식은 소수 q^{p-1}로 묶인 항과 m으로 묶인 항으로 구성되므로 $q|m\sum_{i=0}^{e-1}c_i k^{e-i}$라면 소수 값을 얻을 수 있으나 이와 같을 확률은 $\frac{1}{q}$로 소수를 찾기 위해 전사공격을 하는 것과 같은 확률이다. $s_q\equiv m^{d_q}(\mathrm{mod}\ q)$에 오류가 주입된 경우도 위와 같은 이유로 소수 추출이 어렵다.

$m_p\equiv m(\mathrm{mod}\ p)$ 연산 시 오류를 주입하여 선택 평문 공격을 하는 경우 다음과 같이 식이 전개된다.

$$\hat{s}^e-m$$
$$\equiv s^e\cdot(s_p^{c_p}(\mathrm{mod}\ p)+s_q^{c_q}(\mathrm{mod}\ q))-(\widehat{m_p}+m_q)+1)^e-m$$
$$\equiv m\cdot(s_p^{c_p}(\mathrm{mod}\ p)+s_q^{c_q}(\mathrm{mod}\ q))-(m_p+t+m_q)+1)^e-m$$
$$\equiv m\cdot(1-t)^e-m \quad (60)$$

m_p에 주입된 오류는 s_p에 영향을 주지 않으므로 오류 서명의 복호화와 메시지의 차이는 식(60)의 첫 번째 행과 같이 표현할 수 있다. $\widehat{m}_p=m_p+t$라고 했을 때 소거되는 값을 지우면 식(60)과 같은 결과를 얻는다. 식(60)에는 p와 q가 존재하지 않으므로 최대공약수를 이용하여 소수를 추출할 수 없다. $m_q\equiv m(\mathrm{mod}\ q)$에 오류가 주입한 후 선택 평문 공격을 하는 경우도 동일한 결과를 얻을 수 있다.

2.효율성 분석

<표 2>는 관련 연구와 제안하는 시스템의 동작 및 오류주입 공격에 대한 안전성을 비교하여 보여준다. 제안하는 기법은 기존 기법들에 비하여 오류주입 공격을 방지하면서 병렬 연산을 통해 동일한 법에서 연산을 효율적으로 처리한다. 소수의 크기는 $n=2048bit$, $p=q=1024bit$, $r=512bit$으로 가정하고, 병렬 연산이 가능한 경우 곱셈의 형태로 표시하였다. 키 생성 등 미리 연산할 수 있는 값의 경우 연산 횟수에 포함시키지 않았다. 지수 연산 크기의 증가는 연산 시간의 Exponential한 증가를 의미하며, 횟수의 증가는 연산 시간의 의 Linear한 증가를 나타낸다. RSA-CRT 기법의 연산 과정에서는 지수연산 이외에 모듈러 연산과 곱셈 연산 등으로 컴퓨팅 자원을 사용하지만 지수연산에 비해 그 영향이 미미하여 본 논문에서는 지수연산의 크기 및 횟수 비교에 초점을 맞추었다. Couvreur와 Quisquater의 기법은 각 법에서 서명을 생성하기 위해 1번의 지수연산을 수행한다. Shamir의 기법도 각 법에서 서명을 생성하는데 이때 사용되는 r의 크기는 임의로 $512bit$으로 정하였으며 이후 사용되는 모든 r값은 이 값을 이용한다. 이 경우 $1536bit$의 연산을 병렬적으로 1회 수행한다. Yen등의 기법은 병렬 연산이 불가하여 $1024bit$의 연산을 4회 수행해야 하며 그 결과 값에 다시 $512bit$연산을 해야 한다. 이에 반해 제안하는 기법 중 RSA-CRT-PR은 연산의 크기 및 횟수가 Shamir의 기법과 동일하다. RSA-CRT-PD는 같은 크기의 지수 연산을 하는 Couvreur와 Quisquater의 기법에 비해 2배 시간이 걸리지만 Yen등의 기법에 비해 2배 이상 빠른 연산이 가능하다.

표 2. 관련 연구와의 비교
Table 2. Comparative analysis

구분	지수연산		병렬 연산	오류주입 공격 방지
	크기(bit)	횟수(회)		
RSA[1]	2048	1	X	O
Couvreur[2]	1024	1x2	O	X
Shamir[5]	1536	1x2	O	X
Yen[7]	1024	4	X	X
	512	1		
RSA-CRT-PR	1536	1x2	O	O
RSA-CRT-PD	1024	2x2	O	O

n=2048bit, p=q=1024bit, r=512bit

VI. 결론

본 논문에서는 오류주입 공격에 강건하며 병렬연산이 가능한 두 가지 RSA-CRT기법을 제안하였다. 제안하는 기법은 CRT값에 곱셈의 항등원을 곱하되, 항등원 생성 시 각각 법에서 생성한 서명 값 s_p, s_q를 사용한다. 공격자가 인자로 사용

된 서명들에 오류주입 공격을 하게 되면 오류 값이 서명 전체로 확산되어 상수항을 남기게 되고, 소수를 추출할 수 없도록 한다. 기 제안된 기법들은 초기 RSA-CRT의 오류주입 공격 취약점의 대안으로 제시되었으나 여전히 오류주입 공격에 취약하며 연산시간 또한 증가하였다. 반면 제안하는 기법은 오류 주입 공격에 강건하며, 초기 RSA-CRT 기법에 비해 2배의 연산 시간이 들지만 병렬 연산을 통하여 그 외 기법들에 비해 2배 이상 빠른 연산속도를 제공한다.

NIST의 권고에 따르면, 국내에서 이용 중인 RSA 1024 알고리즘은 2013년 이후 안전성을 보장하기 어려울 것으로 전망되고 있다[13]. 이에 행정안전부에서는 2012년 1월부터 공인인증서 암호체계 고도화를 본격적으로 시행할 것을 발표하였다. 하지만 한국인터넷진흥원은 전자서명키 길이 증가에 따라 전자거래 업체의 서버 측 부하인 전자서명 검증시간이 1024비트 키 대비 3~4배 증가한다는 테스트 결과를 발표하였다. 이와 관련하여 본 연구의 결과는 산업적 측면에서 서버의 부하를 줄일 수 있으므로 암호체계 고도화사업에 필수적으로 사용될 것으로 판단되며, 학술적 측면에서 연구들의 흐름을 쉽게 파악하고 RSA-CRT의 취약점 및 보완점을 쉽게 학습할 수 있는 학습 자료로서의 가치가 있다고 판단된다.

RSA-CRT는 많은 연구팀에서 진행하고 있는 만큼 발표된 논문이 많지만, 진행됨에 따라 속도의 개선보다 오류주입공격에 대한 보호에 치중한 결과 성능이 점점 낮아지고 있는 추세이다. 이에 본 논문은 초기의 연산속도가 빠른 기법들에 대해 비교하였으나, 그 결과 다양한 접근 방법을 다루지 못했다. 향후에는 이를 직접 물리적으로 구현하고 오류주입을 통해 실험적으로 안전성을 분석하고, 다양한 접근방법과의 분석을 통해 안전하며 성능이 좋은 RSA-CRT 기법을 연구할 것이다.

참고문헌

[1] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, 21(2), pp. 129-126, Feb. 1978.

[2] C. Couvreur, and J. Quisquater, "Fast Decipherment Algorithm for RSA Public-Key Cryptosystem," Electronics Letters, 18(21), pp. 905-907, Oct. 1982.

[3] A. Lenstra, "Memo on RSA Signature Generation in the Presence of Faults," Manuscript, Sep. 1996.

[4] D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Fault," EUROCRYPT '97, LNCS 1233, pp. 37-51, May 1997.

[5] A. Shamir, "How to Check Modular Exponentiation," EUROCRYPT '97 Rump Session, May 1997.

[6] A. Shamir, "Method and Apparatus for Protecting Public Key Schemes from Timing and Fault Attacks," United States Patent 5991415, Nov. 1999.

[7] S. Yen, S. Kim, S. Lim, and S. Moon, "RSA Speedup with Chinese Remainder Theorem Immune Against Hardware Fault Cryptoanalysis," IEEE Transactions on Computers, 52(4), pp. 461-472, Apr. 2003.

[8] S. Yen, D. Kim, and S. Moon, "Cryptoanalysis of Two Protocols for RSA with CRT Based on Fault Infection," Fault Diagnosis and Tolerance in Cryptography 2006, LNCS 4236, pp. 53-61 Oct. 2006.

[9] S.K. Kim, T.H. Kim, D.G. Han, Y.H. Park, and S.H. Hong, "Secure RSA with CRT Protected Against Fault Attacks without using Checking Procedure," Journal of The Korea Institute of Information Security & Cryptology, vol. 18, no. 4, pp. 17-25, Aug. 2008.

[10] Y.R. Baek and J.C. Ha, "Chosen Message Attack on the RSA-CRT Countermeasure Based on Fault Propagation Method," Journal of The Korea Institute of Information Security & Cryptology, vol. 20, no.3, pp. 135-140, Jun. 2010.

[11] J.H. Park, S.J. Moon, and J.C. Ha, "Experimental Analysis of Optical Fault Injection Attack for CRT-RSA Cryptosystem," Journal of The Korea Institute of Information Security & Cryptology, vol. 19, no.3, pp. 51-59, Jun. 2009.

[12] J.H. Park, S.J. Moon, and J.C. Ha, "An Experimental Fault Injection Attack on RSA

Cryptosystem using Abnormal Source Voltage," Journal of The Korea Institute of Information Security & Cryptology, vol. 19 no.5, pp. 195-200, Oct. 2009.

[13] E. Barker and A. Roginskey, "Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths," NIST Special Publication 800-131A, Jan. 2011.

저 자 소 개

은 하 수

2010: 한양대학교 컴퓨터공학과 공학사.
2012: 한양대학교 컴퓨터공학과 공학석사.
현 재: 한양대학교 컴퓨터공학과 박사과정.
관심분야: 모바일 보안, NFC, 암호학
Email : hseun@hanyang.ac.kr

오 희 국

1983: 한양대학교 전자공학과 학사.
1989: 아이오와주립대학교 전자계산학과 석사.
1992: 아이오와주립대학교 전자계산학과 박사.
현 재: 한양대학교 컴퓨터공학과 교수.
관심분야: 네트워크 보안, 암호프로토콜
Email : hkoh@hanyang.ac.kr

김 상 진

1995: 한양대학교 컴퓨터공학과 공학사.
1997: 한양대학교 컴퓨터공학과 공학석사.
2002: 한양대학교 컴퓨터공학과 공학박사.
현 재: 한국기술교육대학교 컴퓨터공학부 부교수.
관심분야: 프라이버시 보호, 애드혹 네트워크 보안, 클라우드 컴퓨팅 보안
Email : sangjin@koreatech.ac.kr