Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.4, Aug. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.4.951

삭제와 오류로부터 RSA 개인키를 복구하는 알고리즘

백 유 진†‡
우석대학교

Recovering RSA Private Key Bits from Erasures and Errors

Yoo-Jin Baek†‡
Woosuk University

요 약

현재 가장 많이 사용되고 있는 공개키 암호 알고리즘인 RSA에 대하여, 만약 암·복호문 이외의 부가 정보가 주어진 경우 이를 이용해 RSA 시스템의 안전성을 분석하는 것은 부채널 공격, 격자 기반 공격 등에서 많이 다루어지고 있다. 최근에는 전원이 차단된 DRAM의 데이터 유지 성질을 이용한 Cold Boot Attack에서도 이러한 부가 정보를 이용한 RSA 개인키 복구 방법이 많이 연구되고 있다. 본 논문에서는 전체 비트 중 일부 비트는 삭제가 되고 동시에 일부 비트에는 오류가 있는 RSA 개인키가 주어진 경우 원래의 개인키를 복구하는 문제를 다루며, 구체적으로는 이전에 제안된 Kunihiro 등의 알고리즘과 비교하여 그 성능이 향상된 새로운 RSA 개인키 복구 알고리즘을 제안한다.

ABSTRACT

Under the assumption that there is available some additional information other than plaintext-ciphertext pairs, the security of the RSA cryptosystem has been analyzed by the attack methods such as the side-channel attacks and the lattice-based attacks. Recently, based on the data retention property of the powered-off DRAMs, the so called cold boot attack was proposed in the literature, which is focusing on recovering the various cryptosystems' key from some auxiliary information. This paper is dealing with the problem of recovering the RSA private key with erasures and errors and proposes a new key recovery algorithm which is shown to have better performance than the previous one introduced by Kunihiro et al.

Keywords: RSA Key Recovery, Side-Channel Attack, Lattice-Based Attack, Cold Boot Attack, Hoeffding's Inequality

I. 서 론

RSA 암호시스템은 서로 다른 두 소수 p, q와 그 곱 $N = pq$ 그리고 $ed = 1 \bmod (p-1)(q-1)$을 만족시키는 e, d가 주어졌을 때, 임의의 정수 m에 대하여 $(m^e)^d = m \bmod N$이 성립한다는 성질을 이용하여 암·복호화 연산을 수행하며 일반적으로 d 또는 p, q, d와 특정 관계를 만족시키는 $(p, q, d, d_p, d_q, qInv)$를 시스템의 개인키로 사용한다[1].

RSA 시스템의 평문·암호문 쌍이 주어진 경우 이에 대응하는 평문을 복구하는 문제는 일반적으로 모듈러스 N을 소인수분해하는 것만큼 어려울 것으로 예측이 되고 있다. 하지만 만약 암·복호문 이외에 다른 부가 정보가 주어지는 경우에는 소인수분해 문제보다 쉬울 수 있음이 알려져 있는데, 이러한 안전성 분석 방법에는 대표적으로 격자 기반 분석 방법과 부채널 분석 공격 방법이 있다.

먼저 격자 기반 분석 방법은 RSA 시스템의 개인키 사이에 존재하는 다양한 대수적 관계와 이를 나타내는 방정식을 격자 상의 문제로 변환하고 변환된 문제는 격자 문제 (근사) 해결 알고리즘을 이용해

Received(06. 12. 2017), Modified(07. 17. 2017), Accepted(07. 17. 2017)
† 주저자, yoojin.baek@gmail.com
‡ 교신저자, yoojin.baek@gmail.com(Corresponding author)

RSA의 개인키를 복구한다. 이러한 방법론을 이용해서 Wiener는 만약 비밀 지수 d가 $d < \frac{1}{3}N^{1/4}$을 만족시키면 N을 다항식 시간 안에 인수 분해할 수 있음을 보였고[2], Boneh 등은 이를 확장하여 만약 $d < N^{0.292}$이면 RSA 시스템이 안전하지 않음을 보였다[3]. 또한 Coppersmith는 p 또는 q의 최상위 혹은 최하위 비트 중 절반 이상이 주어지면 다항식 시간 안에 RSA 모듈러스 N을 소인수 분해할 수 있음을 보여주었는데[4], 이러한 논문들은 공통적으로 RSA 개인키에 해당하는 d 또는 $(p, q, d, d_p, d_q, qInv)$의 특정 위치에 있는 연속된 비트가 주어졌음을 가정한다.

부채널 분석 공격은 암호 기기의 연산 과정 중에 발생하는 다양한 부가 정보를 이용하여 암호 기기 내에 저장된 비밀 정보를 복구해내는 공격방법이며, 사용되는 부가 정보의 예로는 연산시간, 전력소비량, 전자기장 등이 있다 [5,6]. 최근에는 전원이 차단된 DRAM(Dynamic Random Access Memory)의 내용이 상당 시간 동안 유지된다는 특성을 이용한 Cold Boot Attack이라는 새로운 부채널 공격 방법이 제안되었는데[7], 특히 이 공격 방법은 Coppersmith 등이 제안한 이전의 격자 기반 분석 방법과는 달리 주어진 부가 정보가 꼭 연속된 비트임을 가정하지 않는다. 이와 관련하여 Heninger 등은 RSA 개인키 (p, q, d, d_p, d_q)의 전체 비트들 중 약 27%가 랜덤하게 주어지면 N을 소인수 분해할 수 있음을 보였고[8], Henecka 등은 (p, q, d, d_p, d_q)의 전체 비트들 중 23.7%에 해당하는 비트들에 오류가 있더라도 (여기서 오류란, 비트 0은 비트 1로, 비트 1은 비트 0으로 바뀌는 것을 의미한다) 원래의 RSA 개인키 전체를 복구할 수 있는 알고리즘을 제안하였다[9]. 또한 Paterson 등은 Cold Boot Attack에 대한 이전 결과를 부호론의 channel capacity 관점에서 재분석한 후 각 알고리즘이 처리할 수 있는 비트 오류율이나 비트 삭제율의 이론적인 상한 값을 제시하였으며[10], Sarkar 등은 Henecka 등의 방법을 이용하여 RSA 시스템의 비밀 지수의 해밍 웨이트가 너무 작으면 해당 시스템이 안전하지 않음을 보였다[11]. 또한 Kunihiro 등은 RSA 개인키 비트들이 일부는 삭제되고 동시에 일부는 오류가 있는 경우 이를 복구하는 방법을 제시하였는데, 구체적으로는 RSA 개인키 (p, q, d, d_p, d_q)의 전체 비트 중 $\delta(0 \le \delta < 1)$의 비율에 해당하는 비트들이 삭제가 되고 또한 $\epsilon(0 \le \epsilon < 1/2)$의 비율에 해당하는 비트들에는 오류가 있는 경우, 만약 δ와 ϵ이

$$1 - \delta - 2\epsilon \ge \sqrt{\frac{2(1-\delta)\ln(2)}{5}} \quad (1)$$

들 만족시키면 매우 높은 성공 확률로 다항시간 안에 RSA 개인키 전체를 복구할 수 있음을 보였다[12].

본 논문에서는 Kunihiro 등이 제안한 알고리즘의 성능을 향상시킴을 목적으로 한다. 구체적으로 RSA 개인키 중 (p, q, d, d_p, d_q)에 대하여 만약 비트 삭제율이 δ이고 비트 오류율이 ϵ인 경우

$$\epsilon \le \frac{1}{2} - \sqrt{\frac{\ln(2)}{10(1-\delta)}} \quad (2)$$

를 만족시키면 매우 높은 성공 확률로 다항시간 안에 RSA 개인키 전체를 복구할 수 있음을 보이고(본문의 따름정리 1 참조), (1)과 (2)가 가리키는 부등식 영역을 비교한 후 본 논문에서 제안한 알고리즘이 Kunihiro 등의 알고리즘보다 성능이 뛰어남을 보인다. 특히 본 논문에서 제안하는 알고리즘은 그 특성상 Heninger 등의 결과와 Henecka 등의 결과를 포함함을 알 수 있는데, 가령 $\epsilon = 0$인 경우는 RSA 개인키 비트 중에서 삭제된 비트만 있는 경우에 해당되며 이 경우 만약 $\delta \le 0.72$이면 RSA 개인키가 다항식 시간 안에 복구될 수 있음을 알 수 있고, $\delta = 0$인 경우는 RSA 개인키 비트에 오류만 있는 경우에 해당되며 이 경우에는 만약 $\epsilon \le 0.24$이면 RSA 개인키를 다항식 시간 안에 복구할 수 있음을 알 수 있다.

본 논문에서는 다음과 같은 표기법을 사용한다.

표기법 ① 양의 정수 x에 대하여 $x[i]$ 또는 x_i는 x의 이진수 표현에서 x의 i번째 비트를 나타낸다. 따라서 $x[0]$ 또는 x_0는 x의 최하위비트(least significant bit)를 나타낸다.
② $\ln(\,)$은 자연로그를 의미한다.

II. 이전 결과

RSA 시스템의 CRT(Chinese Remainder Theorem) 방식 개인키 $(p, q, d, d_p, d_q, qInv)$는 다음 관계를 만족한다[13]:

$$\begin{aligned} N &= pq \\ ed &= 1 \bmod (p-1)(q-1) \\ ed_p &= 1 \bmod (p-1) \\ ed_q &= 1 \bmod (q-1) \\ qInv &= q^{-1} \bmod p. \end{aligned} \tag{3}$$

이 중에서 $qInv$는 이후의 분석에서 사용이 되지 않기 때문에 본 논문에서는 RSA 개인키가 (p, q, d, d_p, d_q)의 형태로 주어짐을 가정한다.

일반적으로 (p, q, d, d_p, d_q)의 한 구성 요소가 주어지면 N의 소인수분해가 가능함이 잘 알려져 있으며, Heninger와 Shacham은 이를 이용해, 만약 (p, q, d, d_p, d_q)의 전체 비트 중 일부 비트가 주어지고 RSA 공개지수 e가 작으면 소수 p, q를 복구할 수 있는 알고리즘 (이하 HS 알고리즘)을 제시하였다[8]. 이를 위해 그들은 먼저 $O(e)$번의 계산을 통해 다음을 만족시키는 정수 k, k_p, k_q의 정확한 값을 계산할 수 있음을 보였다:

$$\begin{aligned} ed &= 1 + k(p-1)(q-1) \\ ed_p &= 1 + k_p(p-1) \\ ed_q &= 1 + k_q(q-1) \end{aligned} \tag{4}$$

그리고 (3)과 (4)로부터 다음과 같은 모듈러 방정식을 유도하였다: $i \geq 1$에 대하여 만약 p', q', d', d_p', d_q'가

$$\begin{aligned} N &= p'q' \bmod 2^i \\ ed' &= 1 + k(p'-1)(q'-1) \bmod 2^i \\ ed_p' &= 1 + k_p'(p'-1) \bmod 2^i \\ ed_q' &= 1 + k_q'(q'-1) \bmod 2^i \end{aligned} \tag{5}$$

을 만족시키면, $p[i], q[i], d[i], d_p[i], d_q[i]$는 다음 식을 만족시킨다.

$$\begin{aligned} &p[i] + q[i] = (N - p'q')[i] \bmod 2 \\ &d[i] + p[i] + q[i] = \\ &\quad (k(p'-1)(q'-1) + 1 - ed')[i] \bmod 2 \\ &d_p[i] + p[i] = \\ &\quad (k_p(p'-1) + 1 - ed_p')[i] \bmod 2 \\ &d_q[i] + q[i] = \\ &\quad (k_q(q'-1) + 1 - ed_q')[i] \bmod 2. \end{aligned} \tag{6}$$

특히 식 (6)은 미지수가 5개이지만 방정식은 4개이기 때문에, 별다른 조건이 주어지지 않으면 $(p[i], q[i], d[i], d_p[i], d_q[i]) \in \{0,1\}^5$에 대한 해가 2개 존재함을 알 수 있다. 따라서 $Slice[i] = (p[i], q[i], d[i], d_p[i], d_q[i]), i \geq 0$ 라고 정의하면 $Slice[0] = (1,1,1,1,1)$ 이고 수식 (5)와 (6)를 이용하면 이로부터 귀납적으로 $Slice[i]$를 계산할 수 있음을 알 수 있다. 그리고 Heninger와 Shacham은, 만약 p, q, d, d_p, d_q 비트 중 일부가 랜덤하게 주어지면 그 주어진 비트와 $Slice[i]$를 비교함으로써 복구된 $Slice[i]$의 유효성을 검사할 수 있고, 만약 주어진 비트의 개수가 충분히 많으면 다항식 시간 안에 RSA 개인키 전체를 알아낼 수 있음을 보였다. 구체적으로, 만약 (p, q, d, d_p, d_q)의 비트 중 27% 이상이 랜덤하게 주어지면 N은 다항식 시간 안에 소인수 분해될 수 있음이 증명되었다.

HS 알고리즘이 RSA 개인키의 일부분이 주어졌을 경우 (또는 동일하게, RSA 개인키의 일부분이 삭제되었을 경우) 개인키 전체를 복구한다면, Henecka 등이 제안한 알고리즘(이하 HMM 알고리즘)은 (p, q, d, d_p, d_q)에 오류가 있는 경우 이를 복구하는 방법을 제시하였다[9]. 즉, HMM 알고리즘은 (p, q, d, d_p, d_q)의 전체 비트 중 오류율 δ에 해당하는 비트들이 반전이 된 형태인 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$로 주어지면 이로부터 원래의 (p, q, d, d_p, d_q)를 복구한다. 이를 위해 HMM 알고리즘은, 1비트씩 RSA 개인키를 복구하는 HS 알고리즘과는 달리, p, q, d, d_p, d_q 각각에 대하여 최하위비트부터 $t(>1)$ 비트를 한꺼번에 복구한다. 즉, HMM 알고리즘에서는 $Slice[0], ..., Slice[(i-1)t]$가 주어지면 (6)을 이용해 $Slice[(i-1)t+1], ..., Slice[it]$를 한꺼

번에 계산하고 이렇게 계산된 $5t$개의 비트 $Slice[(i-1)t+1], ..., Slice[it]$와 이 비트들에 대응되는 위치에 있는 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$의 비트들 사이에 서로 매칭이 되는 비트들의 개수를 계산한다. 그리고 마지막으로 이렇게 계산된 매칭비트들의 개수를 주어진 threshold 값과 비교함으로써 복구된 비트들의 유효성을 검사한다.

HS 알고리즘은 RSA 개인키 비트 중 일부가 삭제된 경우에, 그리고 HMM 알고리즘은 RSA 개인키 비트에 오류가 있는 경우에 각각 적용이 되었다면, Kunihiro 등이 제안한 알고리즘 (이하, Kunihiro 알고리즘)은 RSA 개인키 중 일부분은 삭제되고 동시에 일부분은 오류가 있는 형태인 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$가 주어졌을 때, 원래의 RSA 개인키를 복구한다[12]. 이를 위해 Kunihiro 등은 HMM 알고리즘에서처럼, p, q, d, d_p, d_q 각각에 대하여 여러 비트를 한꺼번에 복구하고, 복구된 비트들 중에서 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$의 삭제된 비트 위치에 해당하는 비트를 제외한 나머지 비트들을 T 비트로 이루어진 블록으로 분해한 후 각 블록에 대하여 HMM알고리즘을 적용한다. 즉 길이가 T인 블록의 비트들과 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$의 대응되는 위치에 있는 비트들 사이에 서로 매칭이 되는 비트의 개수를 조사한 후, 이 값을 주어진 threshold 값과 비교하여 복구된 비트들의 유효성을 검사한다. 이러한 접근법을 바탕으로 Kunihiro 등은, 비트 삭제율 δ와 비트 오류율 ϵ에 따라 변형된 $(\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$가 주어졌을 때 만약

$$1-\delta-2\epsilon \geq \sqrt{\frac{2(1-\delta)\ln(2)}{5}}$$

이면 매우 높은 성공 확률로 다항식 시간 안에 (p, q, d, d_p, d_q)를 복구할 수 있음을 보였다.

본 논문 역시 [12]에서와 마찬가지로, 삭제와 오류가 있는 형태로 RSA 개인키가 주어졌을 때 이를 이용해 원래의 RSA 개인키를 복구하는 새로운 알고리즘을 제안한다. 그리고 제안된 알고리즘의 유효성을 증명하기 위하여 다음의 보조정리를 사용한다.

보조정리 1 (Hoeffding's Inequalities, [14]) $\Pr(X_i = 1) = p$인 베르누이 확률변수 $X_1, ..., X_n$에 대하여 $X = \sum_{i=1}^{n} X_i$라 하면 임의의 $0 < \gamma < 1$에 대하여 다음이 성립한다.

1) $\Pr(X \geq n(p+\gamma)) \leq e^{-2n\gamma^2}$

2) $\Pr(X \leq n(p-\gamma)) \leq e^{-2n\gamma^2}$.

III. RSA 개인키 복구 알고리즘

RSA 개인키의 일부 비트들은 삭제된 형태로, 또 다른 일부 비트들은 오류가 있는 형태로 주어졌을 때 본 논문에서 제안하는 RSA 개인키 복구 알고리즘은 다음과 같다.

Algorithm 1 (RSA 개인키 복구)

입력: 삭제율 $\delta(0 \leq \delta < 1)$, 오류율 $\epsilon(0 \leq \epsilon < 1/2)$, RSA 공개키 (N, e), N의 비트 크기 n, 전체 비트 중 δ 비율만큼 삭제되고 ϵ 비율만큼 오류가 있는 형태로 변형된 RSA 개인키 $\widetilde{sk} = (\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$, 양의 정수 t, threshold 값 $C_i, i = 1, ..., \left\lceil \frac{n/2-1}{t} \right\rceil$, (4)를 만족시키는 정수 k, k_p, k_q

출력: p, q

1. $W_0 \leftarrow \{(1,1,1,1,1)\}$;
2. For $i = 1$ to $\left\lceil \frac{n/2-1}{t} \right\rceil$
   1) $W_i \leftarrow \varnothing$;
   2) 각 $(p', q', d', d_p', d_q') \in W_{i-1}$에 대하여 2^t개의 $(Slice[(i-1)t+1], ..., Slice[it])$를 계산한다.
   3) 2)에서 계산된 $(Slice[(i-1)t+1], ..., Slice[it])$와 그에 대응하는 위치의 $\widetilde{sk}$의 비트들 사이의 매칭 비트의 개수를 조사하고 (만약 $\widetilde{sk}$의 특정 비트가 삭제되어 있으면 그 비트 위치에서는 매칭 되지 않은 것으로 가정한다) 만약 매칭 비트의 개수가 주어진 threshold 값 C_i보다

크거나 같으면 (p', q', d', d_p', d_q')에 $(Slice[(i-1)t+1], ..., Slice[it])$를 이어붙인(concatenation) 값을 W_i에 추가한다.

3. 만약 $p'q' = N$을 만족시키는 $(p', q', d', d_p', d_q') \in W_{\lceil \frac{n/2-1}{t} \rceil}$가 존재하면 p', q'를 출력한다.

Algorithm 1과 관련하여 본 논문에서는 다음과 같은 용어를 사용한다. 먼저 $i \geq 1$에 대하여 $(p', q', d', d_p', d_q') \in W_i$와 RSA 개인키 (p, q, d, d_p, d_q)가

$$\begin{aligned} p' &= p \bmod 2^{it+1} \\ q' &= q \bmod 2^{it+1} \\ d' &= d \bmod 2^{it+1} \\ d_p' &= d_p \bmod 2^{it+1} \\ d_q' &= d_q \bmod 2^{it+1} \end{aligned}$$

을 만족시키면 (p', q', d', d_p', d_q')를 '올바른 부분키 후보'라고 하고, 그렇지 않으면 '틀린 부분키 후보'라고 한다. 따라서 '올바른 부분키 후보'는 RSA 개인키의 각 원소들의 첫 $it+1$ 비트로 이루어진 부분키 후보를 의미한다. 본 논문에서는 또한 Algorithm 1의 유효성을 증명하기 위해서, 논문 [8], [9], [12]에서처럼 다음과 같은 가정을 기반으로 논의를 전개한다.

가정 1 Algorithm 1에서 만약 $(p', q', d', d_p', d_q') \in W_{i-1}$가 '틀린 부분키 후보'이면, 2.2)단계에서 계산된 $(Slice[(i-1)t+1], ..., Slice[it])$의 각 비트를 나타내는 확률변수는 서로 독립이고, 각각은 확률이 $1/2$인 베르누이 확률분포를 따른다.

정리 1. 가정 1 하에서 임의의 $\nu > 0$에 대하여 다음이 성립한다. n비트 크기를 가지는 N에 대하여 (N, e)를 RSA 공개키라 하고 (N, e)에 대응되는 RSA 개인키 $sk = (p, q, d, d_p, d_q)$ 의 전체 비트 중 $\delta (0 \leq \delta < 1)$에 해당하는 비율만큼의 비트는 삭제가 되고, $\epsilon (0 \leq \epsilon < 1/2)$에 해당하는 비율만큼의 비트에는 오류가 있는 형태인 $\widetilde{sk} = (\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d_p}, \tilde{d_q})$가 주어져 있다고 하자. 또한 $t = \left\lceil \frac{\ln(n)}{10(1-\delta)\nu^2} \right\rceil$, $\tau = \left\lceil \frac{n/2-1}{t} \right\rceil$, $1 \leq i \leq \tau$에 대하여 δ_i를 $(Slice[(i-1)t+1], ..., Slice[it])$의 비트 위치에 해당되는 $\widetilde{sk}$의 비트 중 삭제가 되어 있는 비트의 비율이라 하고

$$\gamma_i = \begin{cases} \sqrt{(1+\frac{1}{t})\frac{\ln(2)}{10(1-\delta_i)}} & \delta_i \neq 1 \\ 0 & \delta_i = 1 \end{cases}$$

$$C_i = \begin{cases} 5t(1-\delta_i)(\frac{1}{2}+\gamma_i) & \delta_i \neq 1 \\ 0 & \delta_i = 1 \end{cases}$$

라 하자. 이 때 모든 i에 대하여

$$\epsilon \leq \frac{1}{2} - \gamma_i - \nu \tag{7}$$

이면 Algorithm 1은 적어도 $\left(1 - \sum_{i=1}^{\tau} \left(\frac{1}{n}\right)^{\frac{1-\delta}{1-\delta_i}}\right)(1-\delta^{5t})^{\tau}$의 성공 확률로 $O(n^{2+\frac{\ln(2)}{5\nu^2}})$시간 안에 sk를 복구할 수 있다.

증명) 정리의 증명은 [9]에서 제시한 증명 방법을 바탕으로 다음과 같은 표기법을 사용한다: $1 \leq i \leq \tau$에 대하여

- $X_{c,i}$: Algorithm 1의 2단계의 i 번째 루프에서 '올바른 부분키 후보'와 $\widetilde{sk}$사이의 $5t$개의 비트 중에서 서로 매칭이 되는 비트의 수를 나타내는 확률변수
- $X_{b,i}$: Algorithm 1의 2단계의 i 번째 루프에서 '틀린 부분키 후보'와 $\widetilde{sk}$사이의 $5t$개의 비트 중에서 서로 매칭이 되는 비트의 수를 나타내는 확률변수
- Y_i: Algorithm 1의 2단계의 i 번째 루프에서, 3)단계의 threshold 값 비교를 통과하는 '틀린 부분키 후보'의 개수를 나타내는 확률변수

- $Y = \sum_{i=1}^{\tau} Y_i$
- $Z_{g,i}$: Algorithm 1의 2단계의 i 번째 루프에서, '올바른 부분키 후보'로부터 확장이 되는 '틀린 부분키 후보'의 개수를 나타내는 확률변수
- $Z_{b,i}$: Algorithm 1의 2단계의 i 번째 루프에서, 하나의 '틀린 부분키 후보'로부터 확장이 되는 '틀린 부분키 후보'의 개수를 나타내는 확률변수

이러한 표기법을 바탕으로 우리는 다음을 알 수 있다.

① 모든 $i = 1, ..., \tau$에 대하여 $\delta_i \neq 1$일 확률은 $(1-\delta^{5t})^{\tau}$이며, 이후부터는 모든 $i = 1, ..., \tau$에 대하여 $\delta_i \neq 1$임을 가정하고 논의를 진행한다.

② $X_{c,i} \sim Bin(5t(1-\delta_i), 1-\epsilon)$이고 따라서 $E[X_{c,i}] = 5t(1-\delta_i)(1-\epsilon)$이다. 여기서 $Bin(5t(1-\delta_i), 1-\epsilon)$는 연속된 $5t(1-\delta_i)$번의 독립적 시행에서 각 시행의 확률이 $1-\epsilon$일 때의 이항분포를, 그리고 $E[X_{c,i}]$는 확률변수 $X_{c,i}$ 의 기댓값을 나타낸다.

③ 가정 1에 의하면, $X_{b,i} \sim Bin(5t(1-\delta_i), 1/2)$이다.

④ Algorithm 1의 2단계의 i 번째 루프에서 '틀린 부분키 후보'로부터 생성되는 2^t개의 서로 다른 부분키 후보에 대하여, 확률변수 $Z_{b,i}^j, j = 1, ..., 2^t$를, 만약 j 번째 ($1 \le j \le 2^t$) 후보 키가 Algorithm 1의 2.3) 단계를 통과하면 1을, 그렇지 않으면 0의 값을 가진다고 정의하면, $Z_{b,i} = \sum_{j=1}^{2^t} Z_{b,i}^j$이다. 또한 $Z_{b,i}^j, j = 1, ..., 2^t$는 가정 1에 의하면 확률적으로 서로 독립이기 때문에 임의의 j에 대하여 $E[Z_{b,i}] = 2^t E[Z_{b,i}^j]$이다.

⑤ $E[Z_{b,i}^j] = \Pr(Z_{b,i}^j = 1) = \Pr(X_{b,i} \ge C_i)$이다. 여기서 첫 번째 등식은 기댓값의 정의로부터, 두 번째 등식은 확률변수 $X_{b,i}$의 정의에서 유도가 된다.

⑥
$$\begin{aligned}&\Pr(X_{b,i} \ge C_i)\\ &= \Pr(X_{b,i} \ge 5t(1-\delta_i)(\frac{1}{2}+\gamma_i)\\ &\le e^{-10t(1-\delta_i)\gamma_i^2}\\ &= 2^{-(t+1)}\end{aligned}$$
가 성립한다. 여기서 첫 번째 등식은 C_i의 정의로부터, 다음 부등식은 보조정리 1로부터, 그리고 마지막 등식은 γ_i의 정의로부터 유도된다.

⑦ ④와 ⑤를 결합하면 $E[Z_{b,i}^j] \le 2^{-(t+1)}$임을 알 수 있다.

⑧ ③에 의하면 $E[Z_{b,i}] = 2^t E[Z_{b,i}^j]$이고 ⑥에 의하면 $E[Z_{b,i}^j] \le 2^{-(t+1)}$이기 때문에 $E[Z_{b,i}] \le \frac{1}{2}$이다.

⑨ $E[Y_1] = E[Z_{g,1}]$이고, $i > 1$에 대하여 $E[Y_i] = E[Z_{g,i}] + E[Z_{b,i}]E[Y_{i-1}]$이 성립하기 때문에, 수학적 귀납법에 의하여 $E[Y_i] = \sum_{k=1}^{i} E[Z_{g,k}] \prod_{l=k+1}^{i} E[Z_{b,l}]$이다.

⑩ Algorithm 1의 2단계의 i 번째 루프에서 생성되는 후보 키의 개수는 2^t이기 때문에, $E[Z_{g,i}] \le 2^t$이다.

⑪ ⑧, ⑨, ⑩으로부터
$$\begin{aligned}&E[Y_i] = \sum_{k=1}^{i} E[Z_{g,k}] \prod_{l=k+1}^{i} E[Z_{b,l}]\\ &\le \sum_{k=1}^{i} E[Z_{g,k}] \left(\frac{1}{2}\right)^{i-k}\\ &\le 2^t \sum_{k=1}^{i} \left(\frac{1}{2}\right)^{i-k} < 2^{t+1}\end{aligned}$$
임을 알 수 있다.

⑫ Algorithm 1의 2단계의 i 번째 루프에서 올바른 부분키 후보가 3단계의 테스트를 통과할지 못할 확률은 $\Pr(X_{c,i} < C_i)$이다.

⑬ $\Pr(X_{c,i} < C_i)$

$= \Pr(X_{c,i} < 5t(1-\delta_i)(\frac{1}{2}+\gamma_i))$

$\leq \Pr(X_{c,i} < 5t(1-\delta_i)(1-\epsilon-\nu))$

$\leq e^{-10t(1-\delta_i)\nu^2}$

$\leq e^{-\frac{(1-\delta_i)\ln(n)}{1-\delta}}$

$= \left(\frac{1}{n}\right)^{\frac{1-\delta_i}{1-\delta}}$

임을 알 수 있다. 여기서 첫 번째 등식은 C_i의 정의로부터, 첫 번째 부등식은 부등식 (7)로부터, 두 번째 부등식은 보조정리 1의 Hoeffding's Inequality로부터, 세 번째 부등식은 $t = \left\lceil \frac{\ln(n)}{10(1-\delta)\nu^2} \right\rceil \geq \frac{\ln(n)}{10(1-\delta)\nu^2}$로 부터, 마지막 등식은 로그의 성질로부터 유도된다.

⑭ ⑫와 ⑬으로부터 Algorithm 1의 2단계의 i 번째 루프에서 올바른 부분키 후보가 3)단계의 테스트를 통과할지 못할 확률은 $\left(\frac{1}{n}\right)^{\frac{1-\delta_i}{1-\delta}}$보다 작거나 같음을 알 수 있다. 따라서 Algorithm 1의 2단계에서 올바른 부분키가 모든 i에 대해서 2단계를 통과할 확률 $\Pr$은 다음을 만족시킨다:

$$\Pr = \prod_{i=1}^{\tau}(1-\Pr(X_{c,i} < C_i)$$

$$\geq \prod_{i=1}^{\tau}\left(1-\left(\frac{1}{n}\right)^{\frac{1-\delta}{1-\delta_i}}\right)$$

$$\geq 1-\sum_{i=1}^{\tau}\left(\frac{1}{n}\right)^{\frac{1-\delta}{1-\delta_i}}$$

여기서, 첫 번째 등식은 확률의 정의에 의해서, 첫 번째 부등식은 ⑫와 ⑬으로부터, 두 번째 부등식은 $0 < x_i < 1$에 대하여 $\prod(1-x_i) \geq 1-\sum x_i$가 성립한다는 사실로부터 유도된다.

⑮ [7]의 보조정리 6에 의하면 Algorithm 1이 $O(n^{2+\frac{\ln(2)}{5\nu^2}})$시간 안에 $\widetilde{sk}$를 복구할 수 있다

이상으로부터 정리 1이 증명이 된다. ■

따름정리 1. 가정 1 하에서 다음이 성립한다. n 비트 크기를 가지는 N에 대하여 (N, e)를 RSA 공개키라 하고 n을 충분히 큰 자연수라고 하자. 이때 $t = \left\lceil \frac{\ln(n)}{10(1-\delta)\epsilon^2} \right\rceil$ 에 대하여, 만약 $\delta(0 \leq \delta < 1)$의 비율로 비트가 삭제되고 $\epsilon(0 \leq \epsilon < 1/2)$의 비율만큼 오류가 있는 형태로 변형된 RSA 개인키 $\widetilde{sk} = (\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d}_p, \tilde{d}_q)$가 주어지고

$$\epsilon \leq \frac{1}{2} - \sqrt{\frac{\ln(2)}{10(1-\delta)}} \quad (8)$$

이 성립하면, Algorithm 1은 매우 높은 확률과 n에 대한 다항식 시간 안에 $\widetilde{sk}$로부터 $sk = (p, q, d, d_p, d_q)$를 복구할 수 있다.

증명) 정리 2에서, 만약 n이 충분히 커지면 그에 따라 t의 값도 충분히 커진다. 그에 따라 δ_i와 γ_i는 각각 δ와 $\sqrt{\frac{\ln(2)}{10(1-\delta)}}$ 에 충분히 근접하기 때문에 식 (7)은 다음과 같음을 알 수 있다: 임의의 $\nu > 0$에 대하여

$$\epsilon \leq \frac{1}{2} - \sqrt{\frac{\ln(2)}{10(1-\delta)}} - \nu.$$

따라서 따름 정리가 증명이 된다. ■

논문 [12]의 결과와 본 논문의 결과를 비교해 보면, 먼저 [12]에서는, $\delta(0 < \delta < 1)$의 비율로 비트가 삭제되고 $\epsilon(0 < \epsilon < 1/2)$의 비율만큼 오류가 있는 RSA 개인키 $\widetilde{sk} = (\tilde{p}, \tilde{q}, \tilde{d}, \tilde{d}_p, \tilde{d}_q)$가 주어졌을 때, 만약

$$1-\delta-2\epsilon \geq \sqrt{\frac{2(1-\delta)\ln(2)}{5}}$$

이면 매우 높은 성공 확률로 RSA 개인키를 복구할 수 있음을 보였다. 그러나 본 논문의 따름정리 1에 의하면 만약

$$\epsilon \leq \frac{1}{2} - \sqrt{\frac{\ln(2)}{10(1-\delta)}}$$

를 만족시키면 매우 높은 성공 확률로 RSA 개인키

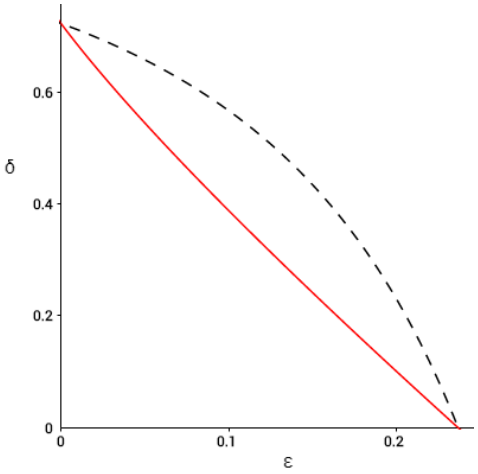

Fig. 1. Graph Comparing the Results of [12] (red solid line) and the New Algorithm (blue dotted line)

를 복구할 수 있음을 알 수 있다. 따라서 두 결과를 비교하기 위해서 각 부등식이 나타내는 영역을 그림으로 표시하면 Fig. 1과 같다.

Figure 1에서 x축은 오류율 ϵ을, y축은 삭제율 δ를 나타낸다. 또한 (붉은 색) 실선에서 원점을 포함하는 왼쪽 아래 영역이 [12]의 결과에 대한 부등식 영역을 나타내고, (검은 색) 점선에서 원점을 포함하는 왼쪽 아래 영역이 본 논문의 결과를 나타낸다. 따라서 [12]의 방법에 비해서 본 논문에서 제안한 방법의 성능이 더 좋음을 알 수 있으며, 이외에도 다음 결과를 도출할 수 있다.

1) 만약 ϵ이 0인 경우, 즉 RSA 개인키 비트 중에서 삭제된 비트만 있는 경우 두 방법 모두 만약 $\delta \leq 0.72$이면 RSA 개인키를 복구할 수 있다.
2) 만약 δ가 0인 경우에는 두 방법 모두 만약 $\epsilon \leq 0.24$이면 RSA 개인키를 복구할 수 있다.

IV. 결 론

본 논문에서는 RSA 개인키 비트들의 일부는 삭제되고 일부는 오류가 있는 경우 이를 복구하는 알고리즘을 제시하였으며, 이 알고리즘은 복구할 수 있는 비트 삭제율과 비트 오류율의 상한이 더 높다는 의미에서 이전의 Kunihiro 등이 제안한 알고리즘에 비해 성능이 개선되었음을 보였다. 또한 본 논문에서 제안한 알고리즘은 Heninger와 Shacham의 결과와 Henecka 등의 결과를 포함하고 따라서 그들의 결과를 일반화한 결과임을 보일 수 있었다.

References

[1] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM vol. 21, no. 2, pp. 120-126, 1978.
[2] M. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Trans. on Information Theory vol. 36, pp. 553 - 558, 1998.
[3] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," IEEE Trans. on Information Theory vol. 46, pp. 1339-1349, 2000.
[4] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," J. Cryptology vol. 10, no. 4, pp. 233 - 260, 1997.
[5] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," CRYPTO '96, LNCS vol. 1109, pp. 104-113, 1996.
[6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," CRYPTO '99, LNCS vol. 1666, pp. 388-397, 1999.
[7] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum and E.W. Felten, "Lest we remember: cold boot attacks on encryption keys," USENIX Security Symposium, pp. 45-60, 2008.
[8] N. Heninger and H. Shacham, "Reconstructing RSA private keys from random key bits," CRYPTO '09, LNCS vol. 5677, pp. 1 - 17, 2009.
[9] W. Henecka, A. May and A. Meurer, "Correcting Errors in RSA Private Keys," CRYPTO '10, LNCS vol. 6223, pp. 351-369, 2010.

〔10〕 K. Paterson, A. Polychroniadou and D. Sibborn, "A Coding-Theoretic Approach to Recovering Noisy RSA Keys," ASIACRYPT 2012, LNCS vol. 7658, pp. 386 - 403, 2012.
〔11〕 S. Sarkar and S. Maitra, "Side channel attack to actual cryptanalysis: Breaking CRT-RSA with low weight decryption exponents," CHES 2012, LNCS vol. 7428, pp. 476-493, 2012.
〔12〕 N. Kunihiro, N. Shinohara and T. Izu, "Recovering RSA Secret Keys from Noisy Key Bits with Erasures and Errors," PKC 2013, LNCS vol. 7778, pp. 180-197, 2013.
〔13〕 RSA Security INc., "Public-Key Cryptography Standards (PKCS) #1 v2.1 RSA Cryptography Standard," 2002.
〔14〕 W. Hoeffding, "Probability inequalities for sums of bounded random variables," Journal of the American Statistical Association vol. 58, no. 301, pp. 13 - 30, 1963.

〈저자소개〉

백 유 진 (Yoo-Jin Baek) 종신회원
1997년 2월: 서울대학교 수학과 졸업
1999년 2월: 서울대학교 수학과 이학석사
2003년 2월: 서울대학교 수리과학부 이학박사
2003년 3월~2003년 6월: KAIST 박사후 연구원
2003년 7월~2013년 3월: 삼성전자 책임연구원
2013년 3월~현재: 우석대학교 정보보안학과 조교수
〈관심분야〉 부채널 공격, 정보 보안, CC 인증