An Improved Attack for Recovering Noisy RSA Secret Keys and its Countermeasure

Noboru Kunihiro

The University of Tokyo, Japan

November 24th, 2015

This research was supported by CREST, JST.

RSA Scheme & PKCS #1 standard

(Textbook) RSA

- $N(=pq), ed \equiv 1 \pmod{(p-1)(q-1)}$
- Public Key (N, e), Secret Key d
- Encryption $C = M^e \bmod N$
- Decryption $M = C^d \bmod N$

Speeding-up via Chinese Remainder Theorem

- Auxiliary Secret Key: $d_p = d \bmod p - 1, d_q = d \bmod q - 1$.
- Compute $M_p = C^{d_p} \bmod p$ and $M_q = C^{d_q} \bmod q$.
- Find M s. t. $M = M_p \bmod p$ and $M = M_q \bmod q$ via CRT.
- Secret Key tuples $(p, q, d, d_p, d_q, q^{-1} \bmod p)$

Secret keys have a redundancy.

Side Channel Attacks against RSA

Extract related values to secret key (p, q, d, d_p, d_q) by physical observation.

$$\boxed{\text{Correct Secret Key}} \xrightarrow[\text{Leakage}]{\text{Observation}} \boxed{\text{Measured Value}}$$

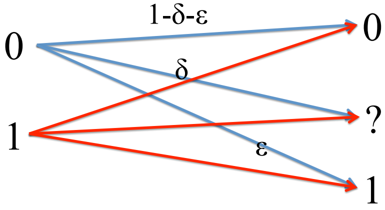
$$\begin{aligned} p &= 110011011 \cdots 1 \\ q &= 100100110 \cdots 1 \\ d &= 1 \cdots 00111 \cdots 1 \\ d_p &= 10111110 \cdots 10 \\ d_q &= 11110110 \cdots 100 \end{aligned} \xrightarrow[\text{Leakage}]{\text{Observation}} \begin{aligned} \tilde{p} &= 100111011 \cdots 1 \\ \tilde{q} &= 100000111 \cdots 1 \\ \tilde{d} &= 1 \cdots 00011 \cdots 1 \\ \tilde{d}_p &= 10111110 \cdots 10 \\ \tilde{d}_q &= 10010110 \cdots 100 \end{aligned}$$

Denote by m the number of involved keys in attacks.

Previous Works for Noisy RSA

Noise Model : Each bit is

- erased with prob. δ. (Heninger–Shacham (CRYPTO2009))
- bit-flipped with prob. ϵ. (Henecka-May–Meurer (CRYPTO2010))
- bit-flipped with asymmetric prob. (Paterson et al. (AC2012))
- erased with prob. δ and bit-Flipped with prob. ϵ. (K–Shinohara–Izu (PKC2013)).



Best Known Results (KSI@PKC2013)

The secret key can be recovered in polynomial time when

$$1-\delta-2\epsilon > \sqrt{\frac{2(1-\delta)\ln 2}{m}}.$$

Theoretical Bound (KSI@PKC2013)

We cannot recover the secret key in polynomial time if

$$(1-\delta)\left(1-H\left(\frac{\epsilon}{1-\delta}\right)\right) < \frac{1}{m}.$$

Open Problem

- KSI pointed out that there is a small gap between the derived condition and the theoretical bound.
- Closing the gap is an open problem.

Our Contributions

Contribution 1

We close the gap by employing Chernoff–Hoeffding Bound.

Contribution 2

- We give a practical countermeasure against the secret-key extraction attack.
- We show the condition so that our countermeasure is valid.

Contribution 3

We give a (provable) bound for asymmetric errors.

Preliminaries

Definition (Binary Entropy)

The binary entropy function $H(x)$ is defined by
$H(x) := -x \log x - (1-x)\log(1-x)$.

Definition (Kullback–Leibler Divergence)

The Kullback–Leibler divergence $D(p, q)$ is defined by

$$D(p,q) := p \log \frac{p}{q} + (1-p)\log \frac{1-p}{1-q}.$$

Useful Inequalities about Binomial Distribution

Proposition (Hoeffding Bound)

Suppose that $X \sim \mathsf{Bin}(n, p)$. For all every $0 < \gamma < 1$, we have

$$\Pr[X \leq n(p - \gamma)] \leq \exp(-2n\gamma^2) \text{ and}$$
$$\Pr[X \geq n(p + \gamma)] \leq \exp(-2n\gamma^2).$$

Proposition (Chernoff–Hoeffding Bound)

Suppose that $X \sim \mathsf{Bin}(n, p)$. For every $0 < \gamma < 1$, we have

$$\Pr[X \leq n(p - \gamma)] \leq \exp(-nD(p - \gamma, p) \ln 2) \text{ and}$$
$$\Pr[X \geq n(p + \gamma)] \leq \exp(-nD(p + \gamma, p) \ln 2).$$

Common Framework

We use Tree-Based approach (proposed by Heninger and Shacham).

$$\mathbf{slice}(i) := (p[i], q[i], d[i+\tau(k)], d_p[i+\tau(k_p)], d_q[i+\tau(k_q)])$$

Assume we obtained a partial secret key up to $\mathbf{slice}(i-1)$.

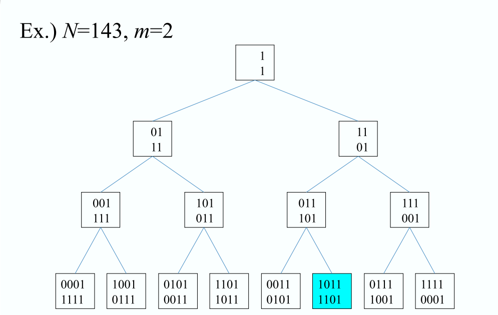
Constraints that each bits satisfies in secret key

$$\begin{aligned} p[i] + q[i] &= c_1 \bmod 2, \\ d[i+\tau(k)] + p[i] + q[i] &= c_2 \bmod 2, \\ d_p[i+\tau(k_p)] + p[i] &= c_3 \bmod 2, \\ d_q[i+\tau(k_q)] + q[i] &= c_4 \bmod 2. \end{aligned}$$

Each bits in $\mathbf{slice}(i)$ have four constraints for five variables.
$\Rightarrow$ There are two candidates.

Tree-based Approach

- Represent $\mathbf{slice}(i)$ by binary tree.
- Once the public key is fixed, the whole binary tree is uniquely determined. The number of leafs in the tree is $2^{n/2}$.
- One of leafs corresponds to the correct secret key.
- Determine with an adequate rule whether each node is discarded or remained by using observed sequence and candidate sequence.



KSI Algorithm@PKC2013

Expansion Phase

- Parameter T.
- We divide the sequence into a T-bit subsequence skipping erasure bits in $\overline{\mathbf{sk}}$.

Rule in Pruning Phase

- Threshold C.
- The Hamming distance between the observed and candidate sequences is larger than C, discard the candidate.

- KSI chose T and C based on the Hoeffding Bound.
- We use Chernoff–Hoeffding bound for improving the success condition.

Success Condition for the Attack

Success Condition

Suppose that we obtain a noisy RSA secret key with error rate (ϵ, δ) satisfying

$$(1-\delta)\left(1-H\left(\frac{\epsilon}{1-\delta}+\zeta\right)\right) \geq \left(1+\frac{1}{t}\right)\frac{1}{m}.$$

Parameter Setting

Suppose that the number of erasure bits is Δ for each block. We choose

$$T=\left\lceil \frac{\log n}{D(\epsilon+\zeta,\epsilon)} \right\rceil \text{ and } C=T\left(\frac{1}{2}+\gamma'\right),$$

where γ' is the solution of the equation of x:
$(1-\delta)\left(1-H\left(\frac{1}{2}-x\right)\right)=\left(1+\frac{1}{t}\right)\frac{1}{m}$.

Computation Time and Success Probability

Our algorithm recovers the correct secret key in average time

$$O\left(n^{2+\frac{2}{mD(\epsilon+\zeta,\epsilon)}+\delta t\frac{\ln 2}{\ln n}}\right)$$

with success probability at least

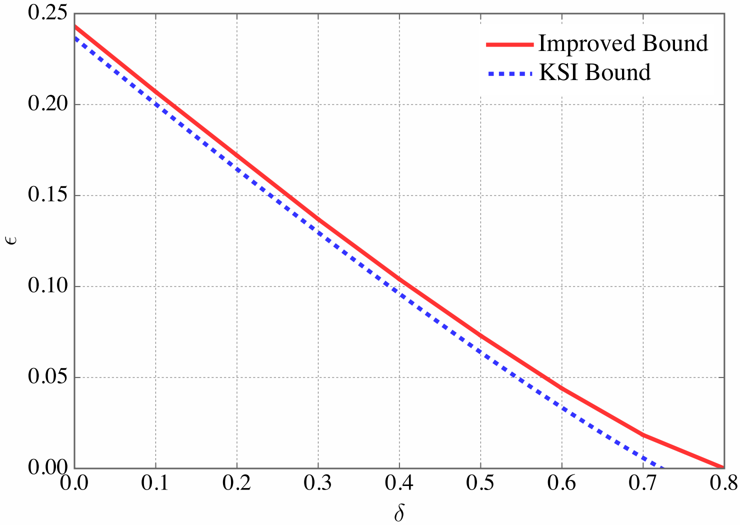
$$1-\left(\frac{mD(\epsilon+\zeta,\epsilon)}{\log n}+\frac{1}{n}\right).$$

Remark

For sufficiently large n, t goes to the infinity, and the success probability is close to 1. Ignoring the term "ζ", we just write the success condition as

$$(1-\delta)\left(1-H\left(\frac{\epsilon}{1-\delta}\right)\right)\geq\frac{1}{m}.$$

Comparison between KSI and our Bounds



(Practical) Countermeasure

Definition ((ϵ, δ)-Adversary)

Extract the secret key with error rate ϵ and erasure rate δ from the storage.

Key idea

The legitimate decryptor intentionally adds small random errors to the original secret key. The error rate is carefully chosen:

- He can performa a fast decryption even if the error is added.
- The attacker cannot reconstruct the correct secret key due to the added errors and his ability.

Experimental results in HMM10 and PPS12 show

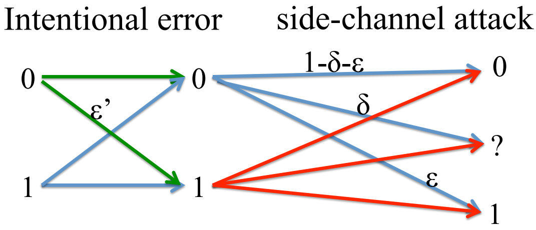
- The secret key can be reconstructed rather fast (less than one second) with high prob. if $\epsilon' \leq 0.15$.
- It is practical to set $\epsilon' = 0.15$ for fast decryption.

Countermeasure

- Setup Phase: (done only once)
  1. Estimate ϵ and δ, which corresponds to the ability of attackers.
  2. Choose ϵ'. Ex.) $\epsilon' = 0.15$ (moderate setting) or $\epsilon' = 0.24$ (aggressive setting)
  3. Store the degraded secret key: each bit in the original secret key is intentionally bit-flipped with probability ϵ'.
  4. Discard the original secret key.
- Decryption Phase: (done for each actual decryption)
  1. Reconstruct the original secret key from the stored secret key.
  2. Decrypt the ciphertext by using reconstructed secret key.

Success Condition for Attack

Total Transition Probability



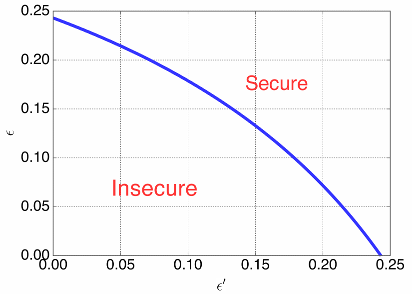
Success Condition for Attack

$$(1-\delta)\left(1-H\left(\frac{\epsilon+\epsilon'-2\epsilon\epsilon'-\epsilon'\delta}{1-\delta}\right)\right)>\frac{1}{m}.$$

Whole Secret Key is Revealed with Errors: $\delta = 0$

The condition ϵ' that the countermeasure is valid is given by

$$\frac{0.243 - \epsilon}{1 - 2\epsilon} < \epsilon' < 0.243.$$



Secure/Insecure Region for (ϵ, ϵ')

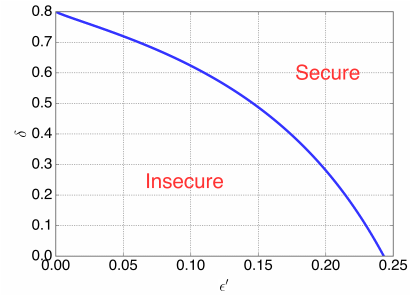
Observation:

When setting $\epsilon' = 0.15$, the countermeasure is valid against the $(\epsilon, 0)$-adversary with $\epsilon > 0.13$.

A Random Fraction is Revealed without any Error

The condition ϵ' that the countermeasure is valid is given by

$$(1-\delta)(1-H(\epsilon')) > \frac{1}{m}.$$



Secure/Insecure Region for (δ, ϵ')

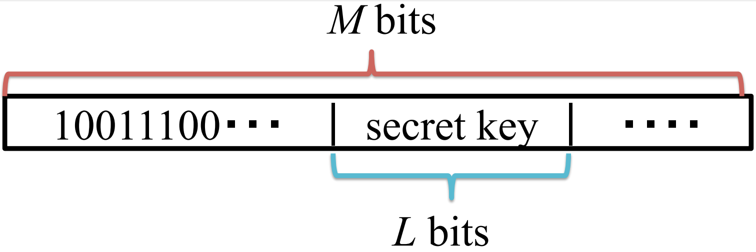
Observation:

- When setting $\epsilon' = 0.15$, the countermeasure is valid against the $(0, \delta)$-adversary with $\delta > 0.49$.
- When setting $\epsilon' = 0.24$ (aggressive setting), more than a 0.976 fraction is necessary for recovering the secret key.

Implications to the Heartbleed Bug

Attacking Situation

The attacker steals only one bit at a random position in a storage.



The average number of trials for obtaining αL-bit ($\alpha \leq 1$) of secret key is given by

$$\frac{M}{L} + \frac{M}{L-1} + \cdots + \frac{M}{L(1-\alpha)}.$$

Bounded by

$$M\left(\frac{1}{L}+\frac{1}{L-1}+\cdots+\frac{1}{L(1-\alpha)}\right)<M\frac{\alpha}{1-\alpha}.$$

Ex.) If $\alpha = 0.2$, upper bounded by $0.25M$.

It is not so tight if α is close to 1. From the so-called coupon collectors argument, the (tighter) upper bound is given by

$$M\left(\frac{1}{L}+\frac{1}{L-1}+\cdots+\frac{1}{1}\right)<M(\ln L+0.5772).$$

- For typical 2048-bit RSA, it is evaluated by $9.12M$.
- The attacker needs about $36(= 9.12/0.25)$ times harder tasks if our countermeasure with aggressive setting $\epsilon' = 0.24$ is applied.

Conclusions

- We close the gap by employing Chernoff–Hoeffding Bound.

$$(1-\delta)\left(1 - H\left(\frac{\epsilon}{1-\delta}\right)\right) > \frac{1}{m}.$$

- We give a practical countermeasure against the secret-key extraction attack.
  - We show the condition so that our countermeasure is valid:

$$(1-\delta)\left(1 - H\left(\frac{\epsilon + \epsilon' - 2\epsilon\epsilon' - \epsilon'\delta}{1-\delta}\right)\right) < \frac{1}{m}.$$

- We give a provable bound for asymmetric errors. (The details are omitted.)