Correcting Errors in Private Keys Obtained from Cold Boot Attacks

Hyung Tae Lee, HongTae Kim, Yoo-Jin Baek, and Jung Hee Cheon

ISaC & Department of Mathmathical Sciences, SNU

2011. 11. 30.

Contents

Cold Boot Attacks

- Cold boot attacks
  - Halderman *et al.* [USENIX '08] (`http://citp.princeton.edu/memory`)

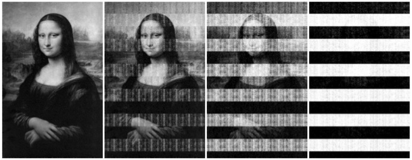

Table: Error rate of cold boot attacks (example)

Temperature	Seconds w/o power	Error rate
Operating Temperature	60	41 %
	300	50 %
-50°C	60	no errors
	300	0.000095 %

Previous Works: RSA-CRT Cryptosystem

- RSA-CRT
  - $C^d \pmod N$
    $\Rightarrow C^{d_p} \pmod p$, $C^{d_q} \pmod q$, and Chinese remainder theorem
    where $d_p \equiv d \bmod (p-1)$, $d_q \equiv d \bmod (q-1)$
  - Private key: (p, q, d, d_p, d_q)

- Previous Results
  - Using equations in variables N, e, p, q, d, d_p, d_q

Table: Previous Results - Recovering Private Keys

Scenario	Reference
Less than 0.73 fraction of p, q, d, d_p, d_q is unknown	Heninger-Shacham (Crypto '09)
Error rate of p, q, d, d_p, d_q is less than 0.237	Henecka-May-Meurer (Crypto '10)

Problem Definition

Definition

Let $\mathbb{G}$ be a finite cyclic group of order q, generated by g. Given an erroneous value $x' \in \mathbb{Z}_q$ with error rate δ, and $y = g^x \in \mathbb{G}$, recover the correct value x .

- Applications
  - DL-based Cryptosystem: $(\mathrm{pk}, \mathrm{sk}) = (g^x, x)$
  - Standard RSA Cryptosystem: $(\mathrm{ct}, \mathrm{msg}) = (C, C^d)$ where $\mathrm{sk} = d$

Previous Works: Splitting System

Definition (Splitting System)

Let n and t be even integers with $0 < t < n$. An (n, t)-splitting system is a pair (X, B) that satisfies the following properties:

1. $|X| = n$ and B is a set of $\frac{n}{2}$-subsets of X called *blocks*.
2. For every $Y \subseteq X$ such that $|Y| = t$, there exists a block $B_i \in B$ such that $|B_i \cap Y| = \frac{t}{2}$.

An (n, t)-splitting system with N blocks is denoted by $(N; n, t)$-splitting system.

Lemma (Coppersmith)

For all even integers n and t with $0 < t < n$, there exists an $(\frac{n}{2}; n, t)$-splitting system for $\mathbb{Z}_n$.

Applications of Splitting System

- Applications
  - Low Hamming Weight Exponent (LWHE) DLP
  - Recovering the private key from an unidirectional erroneous key which has missing bits [Forque *et al.*, CHES '06]

- Idea: when $x = 101011100101$,
  - LHWE DLP: $x' = 000000000000 \Rightarrow x =?$
  - Unidirectional Error: $x' = \cdot 0 \cdot 0 \cdots 00 \cdot 0 \cdot \Rightarrow x =?$
  - Bidirectional Error: $x' = 100010110101 \Rightarrow x =?$

Example I

- Setup
  - $\mathbb{G} = \langle 2 \rangle \subset \mathbb{Z}^*_{2039}$ of order 1019,
  - pk : $y = g^x = 1571$ (sk : $x = 1110101101_2 = 941$),
  - x' : $1010110111_2 \Rightarrow x = \ ?$

- Naive Method (Exhaustive Search):
  Choose t bits from n bits and change corresponding bits (t: the number of error bits)

$$
\begin{array}{c}
1010110111 \\
\downarrow \\
1010110111 \\
\downarrow \\
1110101101 \\
\downarrow \\
2^{1110101101} \equiv 1571 \bmod 2039
\end{array}
$$

Complexity: $\binom{10}{0} + \binom{10}{1} + \binom{10}{2} + \binom{10}{3} + \binom{10}{4} = 386$

Example II

- Our Idea: From $t = 0$ with sequentially increase,
  - When $t = 4$,

$$x' = 1010110111$$

$$U_{1,1} = 0010110000 \qquad U_{1,2} = 1000000111$$

$$\downarrow \qquad \downarrow$$

$$0010110000 \qquad 1000000111$$

$$\downarrow \qquad \downarrow$$

$$\overline{U}_{1,j^*} = 0110100000 \qquad \overline{U}_{2,k^*} = 1000001101$$

$$\downarrow \qquad \downarrow$$

$$2^{0110100000} = 2^{416} \qquad 2^{-1000001101} = 2^{-525}$$

$$\downarrow \qquad \downarrow$$

$$645 \bmod 2039 \qquad 1737 \bmod 2039$$

$$645 \equiv 1541 \cdot 1737 \bmod 2039$$

$$\Rightarrow x = 0110100000 + 1000001101 = 1110101101$$

Complexity: $\binom{5}{0} + 10\left(\binom{5}{1} + \binom{5}{1} + \binom{5}{2} + \binom{5}{2}\right) = 301$

Our Algorithm

Algorithm 1 Recovering private key from the erroneous key x'

INPUT: (g, y, x', n, δ)
OUTPUT: x such that $y = g^x$

for $t = 1$ to $\lfloor n\delta \rfloor$ **do**
  for $i = 0$ to $\lfloor n/2 \rfloor - 1$ **do**
    set $B_{1,i}$ and $B_{2,i}$ to $[i, i + n/2)_n$ and $[i + n/2, i)_n$, respectively
    set $U_{1,i}$ and $U_{2,i}$
    while possible $T_{1,j}$'s **do**
      set $\overline{U}_{1,j}$
      compute $g^{\overline{U}_{1,j}}$ and store $(\overline{U}_{1,j}, g^{\overline{U}_{1,j}})$ in the table Tab
    end while
    while possible $T_{2,k}$'s **do**
      set $\overline{U}_{2,k}$
      compute $yg^{-\overline{U}_{2,k}}$
      find $yg^{-\overline{U}_{2,k}}$ among $g^{\overline{U}_{1,j}}$'s in Tab
      if collision $yg^{-\overline{U}_{2,k^*}} = g^{\overline{U}_{1,j^*}}$ occurs **then**
        return $\overline{U}_{1,j^*} + \overline{U}_{2,k^*}$
      end if
    end while
    initialize the table Tab
  end for
end for

Complexity of Basic Algorithm I

- Computation: $\sum_{t=1}^{\lfloor n\delta \rfloor} n \binom{n/2}{\lceil t/2 \rceil}$
- Storage: $\binom{n/2}{\lceil (\lfloor n\delta \rfloor /2) \rceil}$

Table: Complexity of exhaustive search, Algorithm 1 and unidirectional case ($n = 160$)

Upper bound of error rate (δ)	Complexity		
	Exhaustive search	Algorithm 1	Uni-direction
0.03	$2^{24.69}$	$2^{19.98}$	$2^{17.21}$
0.05	$2^{43.10}$	$2^{28.99}$	$2^{24.65}$
0.10	$2^{71.95}$	$2^{43.24}$	$2^{36.38}$

Complexity of Basic Algorithm II

Table: Complexity of exhaustive search, Algorithm 1 and unidirectional case in RSA ($n = 1024$)

Upper bound of error rate	Complexity		
	Exhaustive search	Algorithm 1	Uni-direction
0.003	$2^{27.42}$	$2^{27.01}$	$2^{24.04}$
0.005	$2^{43.09}$	$2^{34.42}$	$2^{30.49}$
0.010	$2^{78.16}$	$2^{49.08}$	$2^{43.23}$

Applying to Countermeasures: Coron and Kocher's Method I

- Coron and Kocher's Method [Crypto '96, CHES '99]
  - $x \Rightarrow \tilde{x} = x + rq$ where r is a n_r-bit random integer
  - $C^{\tilde{x}} \equiv C^x$ in $\mathbb{G}$
  - Applying our algorithm to Coron and Kocher's method

Table: Lower bound of n_r to provide 2^{80} complexity ($n = 160$)

Upper bound of error rate	0.10	0.15	0.20	0.25	0.30
Lower bound of n_r	155	87	45	24	10

Table: Lower bound of n_r to provide 2^{80} complexity in RSA ($n = 1024$)

Upper bound of error rate	0.005	0.008	0.010	0.015	0.020
Lower bound of n_r	1976	1101	699	243	26

Countermeasures:Clavier and Joye's Method I

- Clavier and Joye's Method [CHES '01]
  - $x = x_1 + x_2$ where x_1 is an random integer, $C^x \equiv C^{x_1} \cdot C^{x_2}$ in $\mathbb{G}$

Algorithm 2 Recovering private key from the erroneous keys x_1', x_2'

INPUT: $(g, y, x_1', x_2', n, \delta)$
OUTPUT: x such that $y = g^x$

for $t_1 = 1$ to $\lfloor n\delta \rfloor$ **do**
  while possible T_1's **do**
    set $\overline{x_1'}$
    compute $g^{\overline{x_1'}}$ and store $(\overline{x_1'}, g^{\overline{x_1'}})$ in the table Tab
  end while
end for
for $t_2 = 1$ to $\lfloor n\delta \rfloor$ **do**
  while possible T_2's **do**
    set $\overline{x_2'}$
    compute $yg^{-\overline{x_2'}}$
    find $yg^{-\overline{x_2'}}$ among $g^{\overline{x_1'}}$'s in the table Tab
    if collision occurs **then**
      return $\overline{x_1'} + \overline{x_2'}$
    end if
  end while

end for

Countermeasures:Clavier and Joye's Method II

- Computation: $2\sum_{t_1=1}^{\lfloor n\delta \rfloor}\binom{n}{t_1}$, Storage: $\sum_{t_1=1}^{\lfloor n\delta \rfloor}\binom{n}{t_1}$

Table: Recovering complexity on Clavier and Joye's method ($n = 160$)

Upper bound of error rate	Complexity		
	Exhaustive search	Algorithm 2	Uni-direction
0.03	$2^{49.30}$	$2^{25.69}$	$2^{21.95}$
0.05	$> 2^{80}$	$2^{44.10}$	$2^{37.05}$
0.10	$> 2^{80}$	$2^{72.95}$	$2^{60.51}$

Table: Recovering complexity on Clavier and Joye's method ($n = 1024$)

Upper bound of error rate	Complexity		
	Exhaustive search	Algorithm 2	Uni-direction
0.003	$2^{55.83}$	$2^{28.42}$	$2^{25.44}$
0.005	$> 2^{80}$	$2^{44.09}$	$2^{39.15}$
0.010	$> 2^{80}$	$2^{79.16}$	$2^{69.39}$

Conclusion

- Provide the algorithm to recover the DL from an erroneous exponent
- Apply to the DL-based cryptosystem and the standard RSA
- Consider breaking countermeasures using our algorithm

Conclusion

- Provide the algorithm to recover the DL from an erroneous exponent
- Apply to the DL-based cryptosystem and the standard RSA
- Consider breaking countermeasures using our algorithm

***** Thank you !! *****