An overview of Cold-Boot Attack, related to RSA and Factorization

Sourav Sen Gupta

Indian Statistical Institute, Kolkata
About this talk

Based on the work “Reconstruction from Random Bits and Error Correction of RSA Secret Parameters”, jointly done with

Santanu Sarkar

&

Subhamoy Maitra

This extends and supplements the work of Heninger and Shacham [Crypto 2009] and that of Henecka, May and Meurer [Crypto 2010].
Contents of this talk

- Cold-Boot attack - a brief introduction

- Application 1: Reconstruction of RSA secret parameters
  - Starting from the LSB side [Heninger and Shacham, 2009]
  - Starting from the MSB side [this work]

- Application 2: Error-Correction of RSA secret parameters
  - Starting from the LSB side [Henecka, May and Meurer, 2010]
  - Starting from the MSB side [this work]

- Implications of Cold-Boot attack on RSA - a summary
Cold-Boot Attack

a brief introduction
Cold-Boot Attack

What happens to your computer memory when the power is down?

Contrary to popular assumption, DRAMs used in most modern computers retain their contents for several seconds after power is lost, even at room temperature and even if removed from a motherboard.

- Halderman et al. [USENIX 2008, Comm. ACM 2009]

Pieces of the puzzle

■ Fact 1: Data remanence in RAM may be prolonged by cooling
■ Fact 2: The memory can be dumped/copied through cold-boot
■ Fact 3: Memory may retain sensitive cryptographic information
Cold-Boot Attack

What happens to your computer memory when the power is down?

Contrary to popular assumption, DRAMs used in most modern computers retain their contents for several seconds after power is lost, even at room temperature and even if removed from a motherboard.

- Halderman et al. [USENIX 2008, Comm. ACM 2009]
Cold-Boot Attack

What happens to your computer memory when the power is down?

Contrary to popular assumption, DRAMs used in most modern computers retain their contents for several seconds after power is lost, even at room temperature and even if removed from a motherboard.

- Halderman et al. [USENIX 2008, Comm. ACM 2009]

Pieces of the puzzle

- Fact 1: Data remanence in RAM may be prolonged by cooling
- Fact 2: The memory can be dumped/copied through cold-boot
- Fact 3: Memory may retain sensitive cryptographic information
Cold Boot Attack

Cold boot attack reads partial information from the memory!
Cold Boot Attack

Cold boot attack reads partial information from the memory!

RSA stores $N, e, p, q, d, d_p, d_q, q^{-1} \text{ mod } p$ in memory (PKCS#1)

Potential information retrieval
- Few random bits of the secret keys $p, q, d, d_p, d_q, q^{-1} \text{ mod } p$
- All bits of secret keys, but with some probability of error
Cold Boot Attack

Cold boot attack reads partial information from the memory!

RSA stores $N, e, p, q, d, d_p, d_q, q^{-1} \mod p$ in memory (PKCS#1)

Potential information retrieval

- Few random bits of the secret keys $p, q, d, d_p, d_q, q^{-1} \mod p$
- All bits of secret keys, but with some probability of error

Question: Does this partial information help the attacker?
Partial Key Exposure attacks on RSA

Rivest and Shamir (Eurocrypt 1985)
$N$ can be factored given 2/3 of the LSBs of a prime.

Coppersmith (Eurocrypt 1996)
$N$ can be factored given 1/2 of the MSBs of a prime.

Boneh, Durfee and Frankel (Asiacrypt 1998)
$N$ can be factored given 1/2 of the LSBs of a prime.

Herrmann and May (Asiacrypt 2008)
$N$ can be factored given a random subset of the bits (small contiguous blocks) in one of the primes.
Partial Key Exposure attacks on RSA

Rivest and Shamir (Eurocrypt 1985)
\( N \) can be factored given 2/3 of the LSBs of a prime.

Coppersmith (Eurocrypt 1996)
\( N \) can be factored given 1/2 of the MSBs of a prime.

Boneh, Durfee and Frankel (Asiacrypt 1998)
\( N \) can be factored given 1/2 of the LSBs of a prime.

Herrmann and May (Asiacrypt 2008)
\( N \) can be factored given a random subset of the bits (small contiguous blocks) in one of the primes.

What if we know random bits?
Reconstruction

of RSA Secret Parameters
Reconstruction of RSA secret parameters

**Situation**

*Cold boot attack provides you with $\delta$ fraction of random bits in each secret parameter $p, q, d, d_p, d_q$, where $0 < \delta < 1$.*

**Problem:** Can one correctly reconstruct these parameters?
Reconstruction of RSA secret parameters

**Situation**

*Cold boot attack provides you with $\delta$ fraction of random bits in each secret parameter $p, q, d, d_p, d_q$, where $0 < \delta < 1$.*

**Problem:** Can one correctly reconstruct these parameters?

- **Heninger and Shacham** (Crypto 2009)
  Reconstruction of secret parameters from the LSB side

- **Maitra, Sarkar and Sen Gupta** (Africacrypt 2010)
  First attempt at reconstruction from the MSB side (known blocks)

- **Sarkar, Sen Gupta and Maitra** (this talk)
  Reconstruction from the MSB side with known random bits
Reconstruction of parameters given $\delta$ fraction of random bits.

Idea: The relation $p[i] \oplus q[i] = (N - p_{i-1}q_{i-1})[i]$ gives a chance for improvised branching and pruning in the search tree.

Either $p[i]$ or $q[i]$ is known

Both $p[i]$ and $q[i]$ are known

\[ \text{Either } p[i] \text{ or } q[i] \text{ is known} \]

\[ \text{Both } p[i] \text{ and } q[i] \text{ are known} \]
Heninger and Shacham (Crypto 2009)

- Reconstruction of parameters given $\delta$ fraction of random bits.

- Idea: The relation $p[i] \oplus q[i] = (N - p_{i-1}q_{i-1})[i]$ gives a chance for improvised branching and pruning in the search tree.

Either $p[i]$ or $q[i]$ is known

Both $p[i]$ and $q[i]$ are known

- Result: One can factor $N$ in time $\text{poly}(e, \log_2 N)$, given
  - $\delta \geq 0.27$ fraction of random bits of $p, q, d, d_p, d_q$, or
  - $\delta \geq 0.42$ fraction of random bits of $p, q, d$, or
  - $\delta \geq 0.57$ fraction of random bits of $p, q$. 

Reconstruction of parameters from the MSB side given small blocks of the parameters are known.

Intuition for primes $p, q$:

\[
\begin{align*}
q_{a-t} & \approx N/p_a \\
p_{2a-t} & \approx N/q_{2a} \\
q_{3a-t} & \approx N/p_{3a}
\end{align*}
\]
Reconstruction of parameters from the MSB side given small blocks of the parameters are known.

Intuition for primes $p, q$:

\[ p_{0} \approx \frac{N}{p_{a}} \]
\[ q_{a-t} \approx \frac{N}{q_{a-t}} \]
\[ p_{2a-t} \approx \frac{N}{q_{2a}} \]
\[ q_{3a-t} \approx \frac{N}{p_{3a}} \]

Result: One can factor $N$ in time $O(\log^2 N)$ with considerable probability of success given $< 70\%$ bits of the primes (together).
Random Bits: Reconstruction of $p, q$

**Context**

- We know $\delta$ fraction of random bits of both primes $p, q$
- The goal is to reconstruct prime $p$ from this knowledge
Random Bits: Reconstruction of $p, q$

**Context**
- We know $\delta$ fraction of random bits of both primes $p, q$
- The goal is to reconstruct prime $p$ from this knowledge

**Step 0. Guess Routine**
- Generate all $2^{a(1-\delta)}$ options for the first window ($a$ MSBs) in $p$
- Pad the remaining by 0’s, and store in an array $A$, say.
Random Bits: Reconstruction of $p, q$

**Step 1.** For each option $\tilde{p}_i \in A$,
- Reconstruct first $(a - t)$ MSBs of $q$ using $\tilde{q}_i = \lfloor \frac{N}{\tilde{p}_i} \rfloor$
- Store these options in an array $B$, say.
- Offset $t$ comes as division is not ‘perfect’

![Diagram of step 1 reconstruction process]
Random Bits: Reconstruction of $p, q$

**Step 2. Filter Routine**

- If for some known bit $q[l]$ of $q$, the corresponding bit in $q_i$ does not match, discard $\tilde{q}_i$ from $B$, and hence $\tilde{p}_i$ from $A$.
- If all the known bits of $q$ match with those of $\tilde{q}_i$, retain $\tilde{p}_i$.

Filtered $A = \{\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_x\}$ where $x = |A| < 2^a(1-\delta)$

**Hope:** Options in $A$ reduce considerably after filtering.
Random Bits: Reconstruction of $p, q$

**Step 3.**
- Each option in $A$ has some correctly recovered block of MSBs.
- Find the initial contiguous common portion out of the options

$$\tilde{p}_1[l] = \tilde{p}_2[l] = \cdots = \tilde{p}_x[l] \quad \text{for all } 1 \leq l \leq c, \text{ not for } c < l \leq a$$
Random Bits: Reconstruction of $p, q$

**Iterate.** Slide the Window

- Take next window of $a$ bits of $p$ starting at the $(c+1)$-th MSB
- Repeat Guess and Filter routines using first $(c+a)$ MSBs of $p$. 

![Diagram showing a spectrum of bits with areas labeled as Recovered, Next block, and Padding of 0's.](image)
Random Bits: Reconstruction of $p, q$

**Iterate.** Slide the Window

- Take next window of $a$ bits of $p$ starting at the $(c + 1)$-th MSB
- Repeat Guess and Filter routines using first $(c + a)$ MSBs of $p$.

![Diagram showing recovery process]

Continue till we get top half of prime $p$.
Then use Coppersmith’s method to factor $N$ efficiently!
Random Bits: Sliding Window Technique

Intuition for the General Algorithm:

1. Fit a window of length $a$ at the top of prime $p$

2. Find out how many bits we know within this window

3. Guess the remaining unknown bits within the window of $a$ bits

4. Filter through the guesses using the partial information known about the bits of all other secret parameters $q, d, d_p, d_q$

5. Slide the window forward and continue the same process
We could factor $N$ with considerable success probability, given

- $\delta \geq 0.38$ fraction of random bits of $p, q, d, d_p, d_q$, or
- $\delta \geq 0.47$ fraction of random bits of $p, q, d$, or
- $\delta \geq 0.62$ fraction of random bits of $p, q$.
Comparison with Heninger-Shacham

Heninger-Shacham: LSB side reconstruction with random bits known

Our work: MSB side reconstruction with random bits known

<table>
<thead>
<tr>
<th>Bits known from</th>
<th>Heninger-Shacham</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
</tr>
<tr>
<td>$p, q$</td>
<td>59%</td>
<td>64%</td>
</tr>
<tr>
<td>$p, q, d$</td>
<td>42%</td>
<td>51%</td>
</tr>
<tr>
<td>$p, q, d, d_p, d_q$</td>
<td>27%</td>
<td>37%</td>
</tr>
</tbody>
</table>
Comparison with Heninger-Shacham

Heninger-Shacham: LSB side reconstruction with random bits known

Our work: MSB side reconstruction with random bits known

<table>
<thead>
<tr>
<th>Bits known from</th>
<th>Heninger Shacham</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
</tr>
<tr>
<td>$p, q$</td>
<td>59%</td>
<td>64%</td>
</tr>
<tr>
<td>$p, q, d$</td>
<td>42%</td>
<td>51%</td>
</tr>
<tr>
<td>$p, q, d, d_p, d_q$</td>
<td>27%</td>
<td>37%</td>
</tr>
</tbody>
</table>

How do you know the bits for sure?