How can we overcome both side channel analysis and fault attacks on RSA-CRT?

2007. Sep. 10.

Chong Hee KIM and

Jean-Jacques QUISQUATER

Content

- Side channel analysis & Fault attacks
- SCA and FA on RSA-CRT
- Approaches to prevent SCA & FA
- Proposed scheme
- Analysis
- Conclusions

SCA & Fault attacks

- *Side channel Analysis* by Kocher, 1996
- Many variants
  - Using timing, power consumption, EM, etc
- *Fault attacks* by Boneh et al., 1997

Side channel Analysis	Fault attacks
Non-invasive	Semi-invasive
Measure timing, power consumption, EM radiation,...	Invoke faults and use the faulty outputs

RSA-CRT

- RSA-CRT (Chinese Remainder Theorem)
  - $N=p \cdot q$: RSA modulus, p and q: large primes
  - $e \cdot d = 1 \bmod (p-1)(q-1)$
  - $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$
  - I_q: inverse of q modulo p

  - Signature S of a message m
    1. $S_p = m^{dp} \bmod p$
       $S_q = m^{dq} \bmod q$
    2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $S_p = m^{dp} \bmod p$

   $S_q = m^{dq} \bmod q$

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $\underline{S_p} = m^{dp} \bmod p$ ⟵ Fault attack

   $S_q = m^{dq} \bmod q$

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $S_p = m^{dp} \bmod p$

   $S_q = m^{dq} \bmod q$

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $S_p = m^{dp} \bmod p$

   $\underline{S_q} = m^{dq} \bmod q$ ⟵ Fault attack

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $S_p = m^{dp} \bmod p$

   $S_q = m^{dq} \bmod q$

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $\underline{S_p} = m^{dp} \bmod p$ ⟵ Fault attack

   $S_q = m^{dq} \bmod q$

2. $S = \mathrm{CRT}(S_p, S_q) = S_q + q \cdot \{(S_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $\underline{S}_p = m^{dp} \bmod p$ ⟵ Fault attack

   $S_q = m^{dq} \bmod q$

2. $\underline{S} = \mathrm{CRT}(\underline{S}_p, S_q) = S_q + q \cdot \{(\underline{S}_p - S_q) \cdot I_q \bmod p\}$

Fault attack on RSA-CRT

– Signature S of message m

1. $\underline{S}_p = m^{dp} \bmod p$ ⟵ Fault attack

   $S_q = m^{dq} \bmod q$

2. $\underline{S} = \mathrm{CRT}(\underline{S}_p, S_q) = S_q + q \cdot \{(\underline{S}_p - S_q) \cdot I_q \bmod p\}$

- Find primes p and q
  - By computing $\mathrm{GCD}(S - \underline{S}, N)$ or $\mathrm{GCD}(\underline{S}^e - m, N)$
    - Proof)
      - $(S - \underline{S}) \neq 0 \bmod p$ and $(S - \underline{S}) = 0 \bmod q$
      - $(S - \underline{S}) = k \cdot q$
      - $\mathrm{GCD}(S - \underline{S}, N) = q$

SCA on RSA-CRT

- RSA-CRT uses two exp. with d_p and d_q
  - Can be the targets for SPA.
- DPA such as MRED (*Modular Reduction using Equidistant Data*, *CHES'02*)
- Safe error attack
  - Be careful to adapt SPA resistant countermeasure

Countermeasure against SPA & FA on RSA-CRT

- Giraud's (FDTC'05)
  - SPA & FA resistant
  - Use the fact that temporary variables (a_0,a_1) are of the form (m^{α}, $m^{\alpha+1}$) in Montgomery Ladder

$$
\begin{aligned}
&a_0 \leftarrow 1 \\
&a_1 \leftarrow m \\
&\text{for } i \text{ from } n-1 \text{ to } 0 \text{ do} \\
&\quad a_{\bar{d}_i} \leftarrow a_{\bar{d}_i} \cdot a_{d_i} \bmod N \\
&\quad a_{d_i} \leftarrow a_{d_i}^2 \bmod N \\
&\text{return } a_0.
\end{aligned}
$$

Giraud's (2005)

- SPA-FA-resistant exponentiation

Input: $m, d = (d_{n-1}, ..., d_0), N$
Output: $m^d \bmod N$

$a_0 \leftarrow m$
$a_1 \leftarrow m^2 \bmod N$
for i **from** $n-2$ **to** 1 **do**
  $a_{\bar{d}_i} \leftarrow a_{\bar{d}_i} \cdot a_{d_i} \bmod N$
  $a_{d_i} \leftarrow a_{d_i}^2 \bmod N$
$a_1 \leftarrow a_1 \cdot a_0 \bmod N$
$a_0 \leftarrow a_0^2 \bmod N$
if (Loop Counter i not modified) & (Exponent d not modified) **then**
  return (a_0, a_1),
else
  return *error*.

$(m^{d-1} \bmod N, m^d \bmod N)$

Giraud's (2005)

- Final scheme

Input: m, p, q, d_p, d_q, I_q
Output: $m^d \bmod N$

1.1 $(S_p^*, S_p) \leftarrow \text{SPA-FA-EXP}(m, d_p, p)$
1.2 $(S_q^*, S_q) \leftarrow \text{SPA-FA-EXP}(m, d_q, q)$
2.1 $S^* \leftarrow \text{CRT}(S_p^*, S_q^*)$
2.2 $S \leftarrow \text{CRT}(S_p, S_q)$
2.3 $S^* \leftarrow m \cdot S^* \bmod (p \cdot q)$
3.1 **if** $S^* = S$ & (Parameters p and q not modified) **then**
3.2 **return** S
3.3 **else**
3.4 **return** *error*

$S^* = m^{d-1} \bmod N$,

$S = m^d \bmod N$

Motivation

- Then how RSA-CRT be secure against DPA as well as SPA, FA?
  - Fumaroli and Vigilant's exp. (FDTC'06) secure against SPA,DPA, and FA.
  - Tried to apply FV's exp. to construct CM on RSA-CRT.
  - But FV's exp. could not detect fault injection.
  - Tried to find another approach.

Fumaroli & Vigilant's exp.

SPA-DPA-FA EXP(m, d, N)

Pick a random $r \in G$

$a_0 \leftarrow r; a_1 \leftarrow rm; a_2 \leftarrow r^{-1}$

for i from $n-1$ to 1 do

$a_{\bar{d}_i} \leftarrow a_{\bar{d}_i} \cdot a_{d_i} \bmod N$

$a_{d_i} \leftarrow a_{d_i}^2 \bmod N$

$a_2 \leftarrow a_2^2 \bmod N$

$a_1 \leftarrow a_1 a_0 \bmod N$

$a_0 \leftarrow {a_0}^2 \bmod N$

$a_2 \leftarrow {a_2}^2 \bmod N$

Return $a_2 a_0$

Fumaroli & Vigilant's exp.

- The following relations holds
  - $(a_0,a_1) = (m^{\alpha}r^{\beta},m^{\alpha+1}r^{\beta})$, $a_2 = r^{-\beta}$
  - $a_2 \cdot (a_0,a_1) = (m^{\alpha},m^{\alpha+1})$

- By using a random number r, it is secure against DPA

Weakness of FV's exp.

- They said in their paper,
  - *If an error occurs on a temporary result or during one of the group operations at any time during the computation, the mutual coherence of a_0,a_1, and a_2 is definitively lost.*

- But if an error occurs during $a_2 \leftarrow a_2 \bmod N$, the mutual coherence of variables is NOT lost.

- Therefore we could not use FV's exp to construct RSA-CRT.

Our approach

- Goal: CM against all known SCA and FA
  - SPA, Safe error attack, DPA, FA
    - Another FA [WISTP'07 by Kim and Quisquater]

New FA in WISTP'07

Step 1. Computation of two exponentiation
- Compute S_p^* and S_q^*

Step 2. CRT combination
- Compute $S^* \leftarrow \mathrm{CRT}(S_p^*, S_q^*)$

Step 3. Fault detection
- Return $\begin{cases} S \leftarrow f(S^*) & \text{if there is no error,} \\ \perp & \text{otherwise.} \end{cases}$

- Attack model
  - First fault on Step 1: Error on one of the exp.
  - Second fault just before Step 3: Skip Step3

New FA in WISTP'07

Step 1. Computation of two exponentiation
- Compute S_p^* and S_q^*

Step 2. CRT combination
- Compute $S^* \leftarrow \mathrm{CRT}(S_p^*, S_q^*)$

Step 3. Fault detection
- Return $\begin{cases} S \leftarrow f(S^*) & \text{if there is no error,} \\ \perp & \text{otherwise.} \end{cases}$

S is stored,

If no error → output S

Else multiply r and output S*r

or output nothing

- Attack model
  - First fault on Step 1: Error on one of the exp.
  - Second fault just before Step 3: Skip Step3

New FA in WISTP'07

Step 1. Computation of two exponentiation
- Compute S_p^* and S_q^*

Step 2. CRT combination
- Compute $S^* \leftarrow \mathrm{CRT}(S_p^*, S_q^*)$

Step 3. Fault detection
- Return $\begin{cases} S \leftarrow f(S^*) & \text{if there is no error,} \\ \perp & \text{otherwise.} \end{cases}$

S is stored,

If no error → output S

Else multiply r and output S*r

or output nothing

To prevent this attack, S*r should be stored,

If no error → remove r and output S

Else output S*r or nothing

Our approach

- Goal: CM against all known SCA and FA
  - SPA, Safe error attack, DPA, FA
    - Another FA [WISTP'07]
- We choose the message randomization
  - to prevent DPA
  - to prevent also FA [WISTP'07]
- Signature randomization until the end of fault detection process

Proposed scheme

0. Initialization step

   - For two co-prime *k*-bit integer t_1, t_2,

   define $p^*=p \cdot t_1$, $q^*=p \cdot t_2$.

   - Compute $d_p = d \bmod \text{phi}(p^*)$, $d_q = d \bmod \text{phi}(q^*)$,

     $e_{ti} = d^{-1} \bmod \text{phi}(t_i)$, where i=1,2.

   - Choose a small random value α (less than one byte) and compute β s.t. $\alpha \cdot \beta = 1 \bmod \text{phi}(N)$.

Proposed scheme

1. Set-up

   - Select a random r in Z^*_N, compute an inverse of r, $a=r^{-1} \bmod N$.

   - Initialize c_1 and c_2 with random values.

   - Compute $b = a^{\beta} \bmod N$.

   - After set-up, r, a, and b are updated each time:

     $r_{i+1} \leftarrow r_i^R \bmod N$,

     $a_{i+1} \leftarrow a_i^R \bmod N$,

     $b_{i+1} \leftarrow b_i^R \bmod N$, R is a random small integer.

Proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

3. Compute $S^* \leftarrow CRT(S^*_p,S^*_q)$.

4. Compute

   $c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$

   $c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

Proposed scheme

- SPA-DPA-EXP

SPA-DPA-EXP$(m, d, N, a, r = a^{-1} \bmod N)$

$C \leftarrow r \bmod N$
$a_0 \leftarrow a \;\&\; a_1 \leftarrow m \cdot a \bmod N$
for i from $n-1$ to 0 do
    $C \leftarrow C^2 \bmod N$
    $C \leftarrow C \cdot a_{d_i} \bmod N$
return C $m^d \cdot r \bmod N$

Analysis of proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

3. Compute $S^* \leftarrow CRT(S^*_p,S^*_q)$.

4. Compute
   $c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$
   $c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

When there is no attack

$S^*_p = m^{dp} \cdot r \bmod p^*$

$S^*_q = m^{dq} \cdot r \bmod q^*$

$S^* = S \cdot r \bmod N^*$

$c_1 = c_2 = 1$

$S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)}$

$= (S \cdot r) \cdot (a^{\beta})^{(c1 \cdot c2 \cdot \alpha)}$

$= S \cdot r \cdot a = S$

Analysis of proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

3. Compute $S^* \leftarrow CRT(S^*_p,S^*_q)$.

4. Compute
   $c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$
   $c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

Bellcore attack

$S^*_p = m^{dp} \cdot r \bmod p^*$

$S^*_q = m^{dq} \cdot r \bmod q^*$

$S^* = S \cdot r \bmod N^*$

$c_1 \neq 1, \; c_2 = 1$

$S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)}$

$= (S \cdot r) \cdot (a^{\beta})^{(c1 \cdot c2 \cdot \alpha)}$

$= S \cdot r \cdot a^{c1} \neq Sp \bmod p$

$\neq Sq \bmod q$

Analysis of proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

Fault attack in WISTP07

$S^*_p = m^{dp} \cdot r \bmod p^*$

$S^*_q = m^{dq} \cdot r \bmod q^*$

3. Compute $S^* \leftarrow CRT(S^*_p,S^*_q)$.

$S^* = S \cdot r \bmod N^*$

2nd attack to skip detection

4. Compute

$c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$

$c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

1. Skip computation c1,c2

→ Already initialized with random numbers

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

Analysis of proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

Fault attack in WISTP07

$S^*_p = m^{dp} \cdot r \bmod p^*$

$S^*_q = m^{dq} \cdot r \bmod q^*$

3. Compute $S^* \leftarrow \mathrm{CRT}(S^*_p, S^*_q)$.

$S^* = S \cdot r \bmod N^*$

2nd attack to skip detection

4. Compute

$c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$

$c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

2. Skip Step 5.

$\rightarrow S^*$

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

Analysis of proposed scheme

2. Compute
   - $S^*_p \leftarrow$ SPA-DPA-EXP(m,d_p,p^*,a,r)
   - $S^*_q \leftarrow$ SPA-DPA-EXP(m,d_q,q^*,a,r).

Fault attack in WISTP07

$S^*_p = m^{dp} \cdot r \bmod p^*$

$S^*_q = m^{dq} \cdot r \bmod q^*$

3. Compute $S^* \leftarrow \text{CRT}(S^*_p, S^*_q)$.

$S^* = S \cdot r \bmod N^*$

2nd attack to skip detection

4. Compute

$c_1 = (m \cdot r^{et1} - (S^*)^{et1} + 1) \bmod t_1$

$c_2 = (m \cdot r^{et2} - (S^*)^{et2} + 1) \bmod t_2$

3. Skip exp. of $b^{(c1 \cdot c2 \cdot \alpha)}$

$\rightarrow S^* \cdot b = (S \cdot r) \cdot a^{\beta} \neq S$

5. Return $S \leftarrow S^* \cdot b^{(c1 \cdot c2 \cdot \alpha)} \bmod N$

Analysis of proposed scheme

- SPA, DPA, Safe-error attack
  - SPA-DPA-EXP(m,d_p,p^*,a,r) has a balanced structure → Immune to SPA
  - No redundant operations → Immune to Safe-error attack
  - All intermediate values are randomized → Immune to DPA

Analysis of proposed scheme

- Efficiency
  - Main bottleneck is two modular exp.
  - Assuming that random number r, a, and b are pre-computed and updated each time (Still there is a problem storing these values).

	Giraud's	Our proposal
Secure against	SPA, FA	SPA, DPA, FA
Memory usage	2 registers (3 registers)	3 registers (4 registers)
Bit operations	$2 \cdot 2n^3$	$2 \cdot 2n(n+k)^2$

	Giraud's	Our proposal
Required word operation $(ex. 1024 - bits, k = 32)$	131,072 1	139,392 1.06
Implementation result	8.531 ms 1	9.578 ms 1.12

Conclusion

- Showed a weakness in Fumaroli and Vigilant's exponentiation against fault attack detection.
- Firstly proposed a countermeasure on RSA-CRT secure against all known SCA and FA's.