Fault attacks on RSA-CRT: new attacks, new results, and new countermeasures

2007. May. 10.

Chong Hee KIM and
Jean-Jacques QUISQUATER

Content

- Fault attacks?
- Fault attacks on RSA-CRT
- Previous countermeasures
- Problem of previous countermeasures
- New attack
- Experimental results
- New countermeasures
- Conclusions

Fault attacks (1/2)

- *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
Passive	Semi-invasive
Measure timing, power consumption, EM radiation,...	Invoke faults and use the faulty outputs

Fault attacks (2/2)

- Methods of invoking faults
  - Variation in voltage, ext. clock, temperature, white light, laser, X-rays, ion beams ...
- We used Glitch in supply voltage
  - Easy to treat (only control timing)
  - Glitch is used to break RSA (2002) and DSA (2005)

Fault attacks on 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 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 attacks 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 attacks 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 attacks 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 attacks 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 attacks 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 attacks 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 attacks 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 attacks 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 $(S - \underline{S}) = 0 \bmod q$
      - $(S - \underline{S}) = k \cdot q$
      - $\mathrm{GCD}(S - \underline{S}, N) = q$

Previous countermeasures

- Simple methods
  1. Compute signature twice and compare them
     - Drawback
       - Double computation time
       - Cannot avoid permanent errors
  2. Verify signature with public exponent *e*
     - Device returns *S* only when $S^e \equiv m \bmod N$
     - Drawback
       - Too costly if *e* is large
       - In some applications (e.g. Javacard), access of *e* is not allowed during signature generation

Previous countermeasures

- Shamir's (1997)
  - Use redundancy in computing S_p and S_q
  - Check correctness before RSA combination
  - Let *r* be a random *k*-bit integer (typically, k=32)

1. $S_p^* = m^d \bmod (p \cdot r)$
   $S_q^* = m^d \bmod (q \cdot r)$

2. $\begin{cases} S = \mathrm{CRT}(S_p^*, S_q^*) \bmod N & \text{if } S_p^* \equiv S_q^* \bmod r, \\ error & \text{otherwise.} \end{cases}$

Previous countermeasures

- Shamir's generalizations (2001)
  - Joye et al. pointed out Shamir's required *d* which is not known in CRT. Only d_p and d_q are known in CRT.
  - Let r_1, r_2 be random *k*-bit integers.

1. $S_p^* = m^{d_p} \bmod (p \cdot r_1)$, $s_1 = m^{d_p \bmod \varphi(r_1)} \bmod r_1$
   $S_q^* = m^{d_q} \bmod (q \cdot r_2)$, $s_2 = m^{d_q \bmod \varphi(r_2)} \bmod r_2$

2. $\begin{cases} S = \mathrm{CRT}(S_p^*, S_q^*) \bmod N & \text{if } S_p^* \equiv s_1 \bmod r_1 \text{ and } S_q^* \equiv s_2 \bmod r_2 \\ error & \text{otherwise.} \end{cases}$

Previous countermeasures

- Aumüller et al.'s (2002)
  - Previous alg.'s cannot detect errors occurred during RSA combination
  - Check errors in each step

Previous countermeasures

1. $p' = p \cdot r$
   $d'_p = d_p + random_1 \cdot (p-1)$
   $S'_p = m^{d'_p} \bmod p'$
   if $\neg(p' \bmod p \equiv 0 \wedge d'_p \bmod (p-1) \equiv d_p)$ then return *error*

   $q' = q \cdot r$
   $d'_q = d_q + random_2 \cdot (q-1)$
   $S'_q = m^{d'_q} \bmod q'$
   if $\neg(q' \bmod q \equiv 0 \wedge q'_p \bmod (q-1) \equiv d_q)$ then return *error*

2. $S_p = S'_p \bmod p$
   $S_q = S'_q \bmod q$
   $S = \mathrm{CRT}(S_p, S_q)$

3. if $\neg(S \bmod p \equiv S_p \wedge S \bmod q \equiv S_q)$ then return *error*

   $S_{pr} = S'_p \bmod r$
   $d_{pr} = d'_p \bmod (r-1)$
   $S_{qr} = S'_q \bmod r$
   $d_{qr} = d'_q \bmod (r-1)$
   if $(S_{pr}^{d_{qr}} \equiv S_{qr}^{d_{pr}})$ then
   return S,
   else
   return *error*.

Previous countermeasures

- Infective computations
  - Yen et al. (2003)
    - Error detection based on decisional tests should be avoid.
    - If an error occurs in one of the exp., then it makes both $\underline{S} \neq S \bmod p$ and $\underline{S} \neq S \bmod q$
    - Shown to be insecure by Yen and Kim (2004)
  - Blömer et al. (2003)
    - Shamir's method + fault infective method
    - Shown to be insecure by Wagner (2004)
    - Variation by Blömer and Otto (2006)

Previous countermeasures

- Blömer et al. (2003)

1. $S_p^* = m^{d_1} \bmod (r_1 p),$
   $S_q^* = m^{d_2} \bmod (r_2 q),$

2. $S^* = \mathrm{CRT}(S_p^*, S_q^*) \bmod (r_1 r_2 N),$

3. $c_1 = (m - S^{*e_1} + 1) \bmod r_1,$
   $c_2 = (m - S^{*e_2} + 1) \bmod r_2,$

   $S = (S^*)^{c_1 c_2} \bmod N.$

Previous countermeasures

- Ciet and Joye's (2005)
  - Generalize Shamir's + fault infective

Input: $m, p, q, d_p, d_q, I_q^*, r_1, r_2$
Output: $m^d \bmod N$

1.1 $S_p^* \leftarrow m^{d_p} \bmod p^*$ and $s_2 = m^{d_q \bmod \varphi(r_2)} \bmod r_2$,
1.2 $S_q^* \leftarrow m^{d_q} \bmod q^*$ and $s_1 = m^{d_p \bmod \varphi(r_1)} \bmod r_1$,
2. $S^* \leftarrow S_q^* + q^* \cdot I_q^* \cdot (S_p^* - S_q^*) \bmod p^*$,
3.1 $c_1 \leftarrow (S^* - s_1 + 1) \bmod r_1$
3.2 $c_2 \leftarrow (S^* - s_2 + 1) \bmod r_2$
3.3 $\gamma \leftarrow \lfloor (r_3 c_1 + (2^l - r_3) c_2)/2^l \rfloor$
3.4 $S \leftarrow (S^*)^\gamma \bmod N$
4. **return** S.

$p^* = r_1 p$,
$q^* = r_2 q$,
$q^* Inv = (q^*)^{-1} \bmod p^*$.

Previous countermeasures

- Giraud's (2005)
  - Use the fact that temporary variables (a_0,a_1) are of the form (m^{α}, $m^{\alpha+1}$) in Montgomery Ladder

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

Previous countermeasures

- 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*.

Previous countermeasures

- Giraud's (2005)

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*

Problem of countermeasures

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

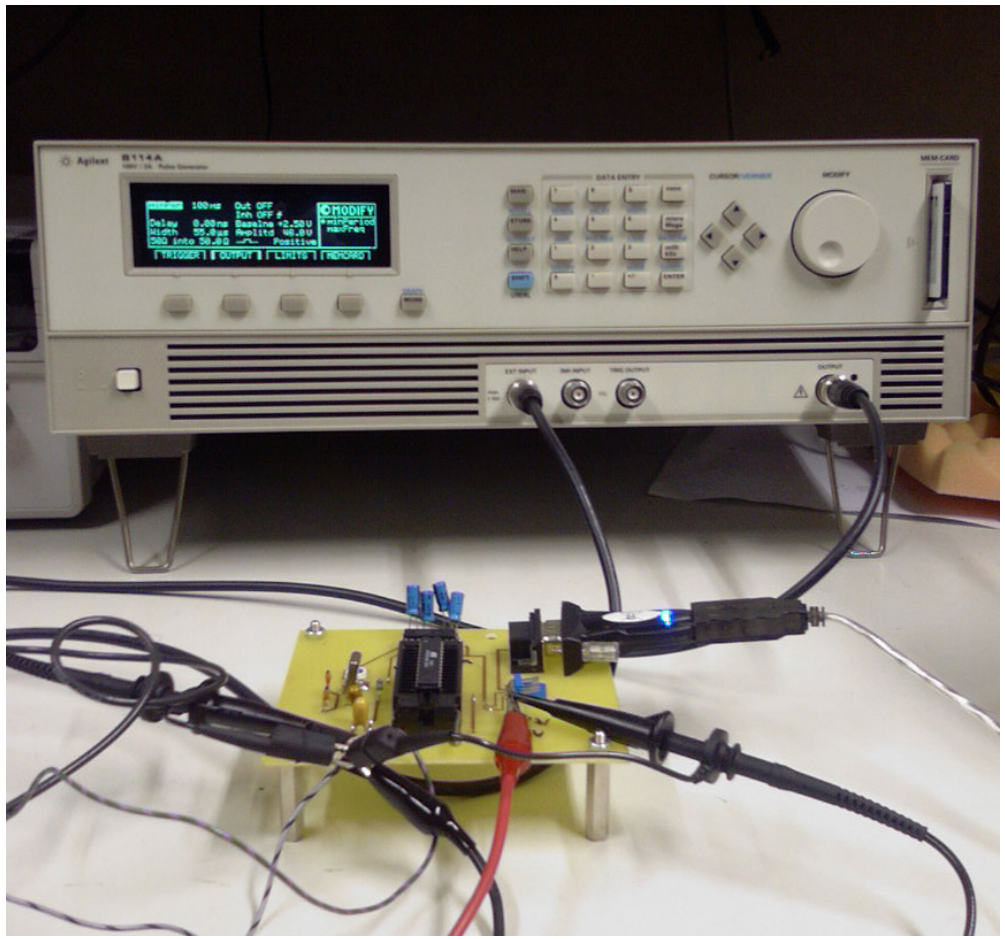
- Attack model
  - First fault on Step 1: Error on one of the exp.
  - Second fault just before Step 3: Skip Step3

New attack

- Proof of the Attack model
  - Implemented 128-bit RSA-CRT with Ciet and Joye's and Giraud's in an Atmel 8-bit AVR microcontroller, ATMega168
  - Implemented program

```
Main() {
        ...
        Set I/O pin low
        Call subroutine RSA-CRT with CJ (as in 2.4)
        Set I/O pin high
        ...
}
```


Experimental results



- Generator for Glitch on supply voltage
- Circuit
- Normally running at 5V
- Glitch ~ 90V 50us

Experimental results

	Target	1st Glitch	2nd Glitch
Exp.1	CJ	Step1.1	Step3.4
Exp.2	CJ	Step1.1	Step3.1~3.3
Exp.3	Gir	Step1.1	Step3.1

Experimental results

- 1st experiment on Ciet and Joye's

Input: $m, p, q, d_p, d_q, I_q^*, r_1, r_2$
Output: $m^d \bmod N$

1.1 $S_p^* \leftarrow m^{d_p} \bmod p^*$ and $s_2 = m^{d_q \bmod \varphi(r_2)} \bmod r_2$,
1.2 $S_q^* \leftarrow m^{d_q} \bmod q^*$ and $s_1 = m^{d_p \bmod \varphi(r_1)} \bmod r_1$,
2. $S^* \leftarrow S_q^* + q^* \cdot I_q^* \cdot (S_p^* - S_q^*) \bmod p^*$,
3.1 $c_1 \leftarrow (S^* - s_1 + 1) \bmod r_1$
3.2 $c_2 \leftarrow (S^* - s_2 + 1) \bmod r_2$
3.3 $\gamma \leftarrow \lfloor (r_3 c_1 + (2^l - r_3) c_2)/2^l \rfloor$
3.4 $S \leftarrow (S^*)^\gamma \bmod N$
4. **return** S.

$p^* = r_1 p,$
$q^* = r_2 q,$
$q^* Inv = (q^*)^{-1} \bmod p^*.$

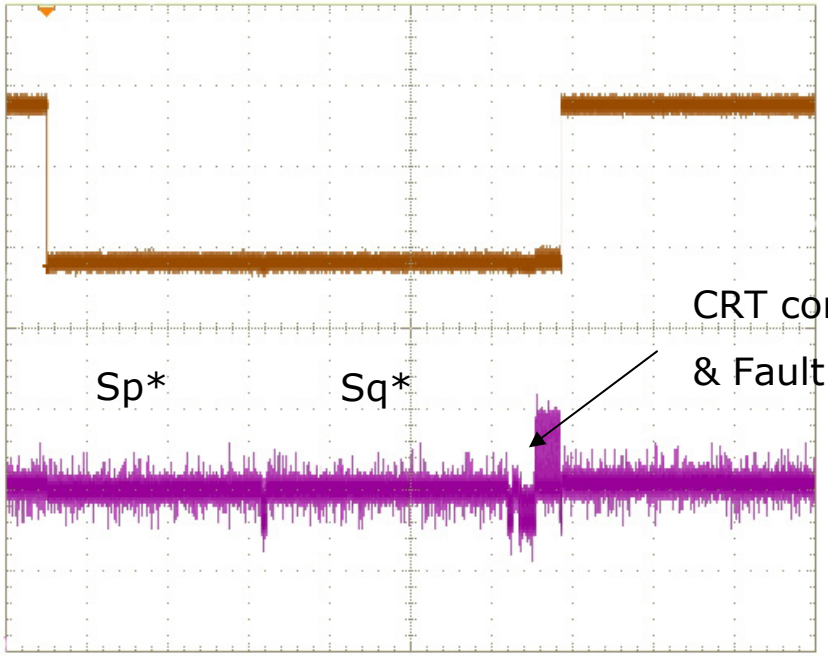


Fig.1 RSA-CRT with CJ without faults

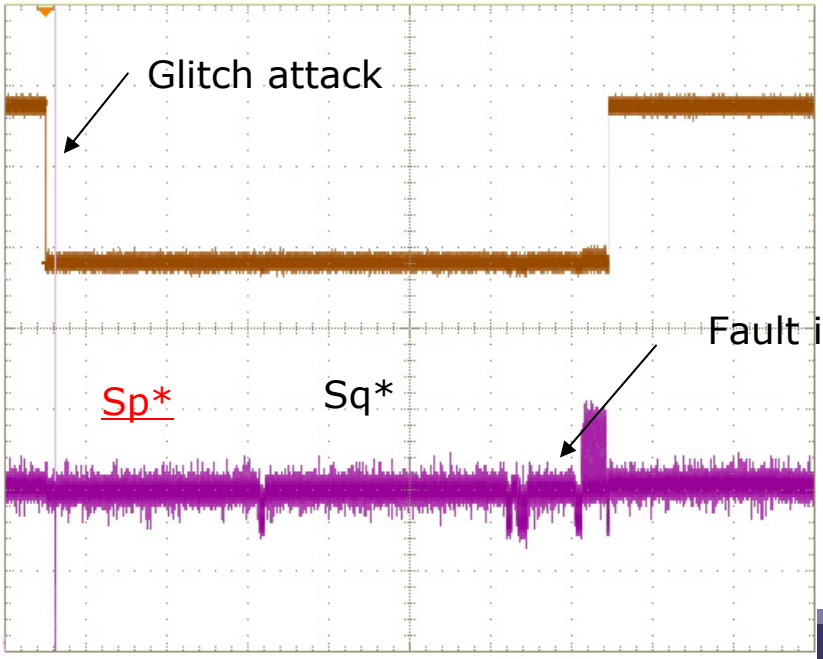


Fig.2 Glitch attack during Sp* exp.

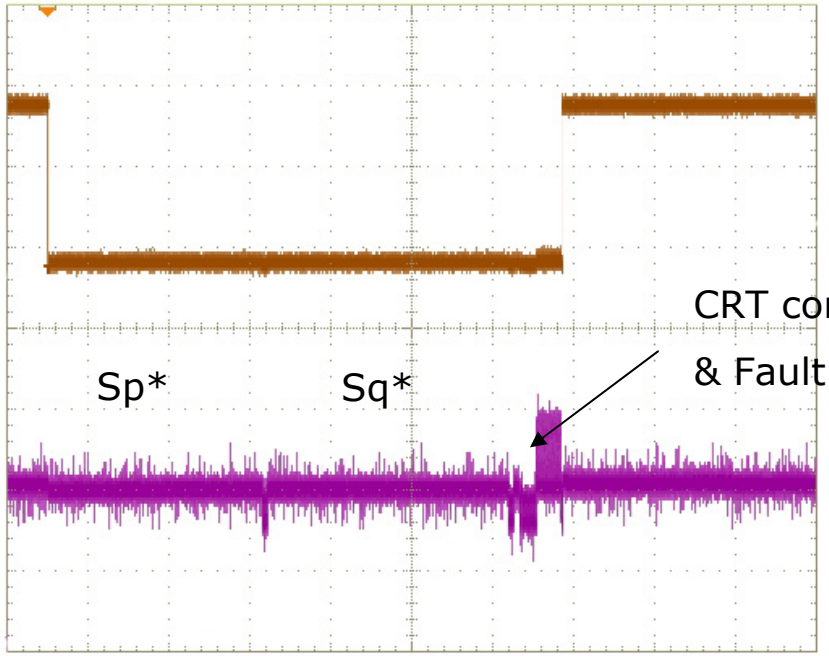
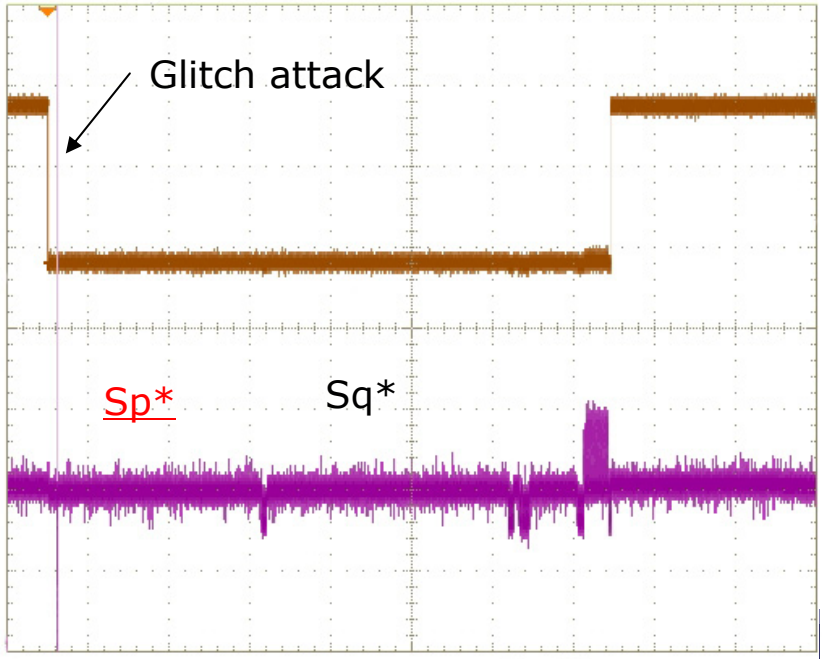
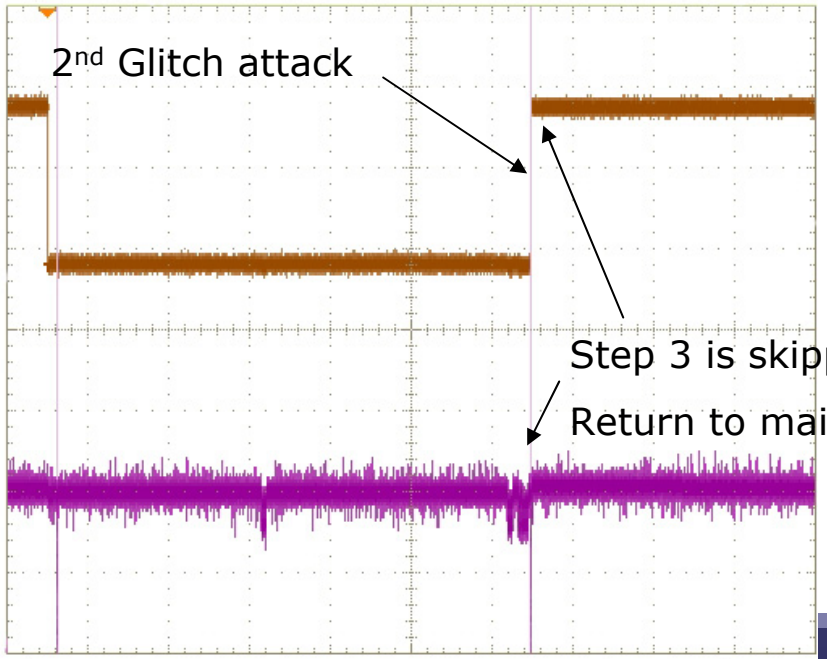


Fig.3 Glitch attack both Sp* and $(S^*)^{\gamma}$





Experimental results

- 2nd experiment on Ciet and Joye's

Input: $m, p, q, d_p, d_q, I_q^*, r_1, r_2$
Output: $m^d \bmod N$

1.1 $S_p^* \leftarrow m^{d_p} \bmod p^*$ and $s_2 = m^{d_q \bmod \varphi(r_2)} \bmod r_2$,
1.2 $S_q^* \leftarrow m^{d_q} \bmod q^*$ and $s_1 = m^{d_p \bmod \varphi(r_1)} \bmod r_1$,
2. $S^* \leftarrow S_q^* + q^* \cdot I_q^* \cdot (S_p^* - S_q^*) \bmod p^*$,
3.1 $c_1 \leftarrow (S^* - s_1 + 1) \bmod r_1$
3.2 $c_2 \leftarrow (S^* - s_2 + 1) \bmod r_2$
3.3 $\gamma \leftarrow \lfloor (r_3 c_1 + (2^l - r_3) c_2)/2^l \rfloor$
3.4 $S \leftarrow (S^*)^\gamma \bmod N$
4. **return** S.

$p^* = r_1 p$,
$q^* = r_2 q$,
$q^* Inv = (q^*)^{-1} \bmod p^*$.

Experimental results

- 3rd experiment on Giraud's

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*

BRNE

Experimental results

	Target	1st Glitch	2nd Glitch	Results
Exp.1	CJ	Step1.1	Step3.4	Skip of Step3.4
Exp.2	CJ	Step1.1	Step3.1~3.3	Skip from Step3.1 ~ 3.3
Exp.3	Gir	Step1.1	Step3.1	Jump to Step3.2 instead of Step3.4

- Error on S_p* is occurred in all experiments by 1st Glitch

New countermeasure

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

- How
  - Randomize *S** until the final error detection

New countermeasure

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

With S*, attacker should not get any useful info.

- How
  - Randomize *S** until the final error detection

New approach to prevent fault attacks

- Modified Ciet and Joye's

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

0. Choose a random integer $\mathbf{a}$ in $Z^*_{r_1 r_2 N}$
   Initialize γ with a random number
1. $S^*_p = (\mathbf{a} + m^{d_p}) \bmod p^*$ and $s_2 = (\mathbf{a} + m^{d_q \bmod \varphi(r_2)}) \bmod r_2$,
   $S^*_q = (\mathbf{a} + m^{d_q}) \bmod q^*$ and $s_1 = (\mathbf{a} + m^{d_p \bmod \varphi(r_1)}) \bmod r_1$,
2. $S^* = S^*_q + q^* \cdot I^*_q \cdot (S^*_p - S^*_q) \bmod p^*$,

3.1 $c_1 = (S^* - s_1 + 1) \bmod r_1$
3.2 $c_2 = (S^* - s_2 + 1) \bmod r_2$
3.3 $\gamma = \lfloor (r_3 c_1 + (2^l - r_3) c_2)/2^l \rfloor$
3.4 $S = (S^* - \mathbf{a}^\gamma) \bmod N$
4. **return** S.

S* = S + a

New approach to prevent fault attacks

- Modified SPA-FA-EXP

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**
  $a_{\bar{d}_i} \leftarrow a_{\bar{d}_i} \cdot a_{d_i} \bmod N$
  $a_{d_i} \leftarrow a_{d_i}^2 \bmod N$
$a_1 \leftarrow (\mathbf{a} + a_1 \cdot a_0) \bmod N$
$a_0 \leftarrow (\mathbf{a} + a_0^2) \bmod N$
if (Loop Counter i not modified) & (Exponent d not modified) **then**
  return (a_0, a_1),
else
  return *error*.

New approach to prevent fault attacks

- Modified Giraud's

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

0. Choose a random integer $\mathbf{a}$ and $\mathbf{b}$ in Z_N^*
1. $(S_p^*, S_p) \leftarrow \text{SPA-FA-EXP}^*(m, d_p, p, \mathbf{a})$
   $(S_q^*, S_q) \leftarrow \text{SPA-FA-EXP}^*(m, d_q, q, \mathbf{a})$
2. $S^* = \text{CRT}(S_p^*, S_q^*)$
   $S = \text{CRT}(S_p, S_q)$
   $S^* = (m \cdot S^* + \mathbf{a}) \bmod (p \cdot q)$
   $S = (S + \mathbf{a} \cdot m) \bmod (p \cdot q)$
3. **if** $(S^* = S)$ & (Parameters p and q not modified) **then**
   return $(S - \mathbf{a} - \mathbf{a} \cdot m + (S^* - S) \cdot \mathbf{b}) \bmod N$
   else
   return *error*

S* = S + a + am

Conclusion

- Pointed out the weakness of previous countermeasures against FA on CRT-RSA
- Show experimental results on two times faults during one execution of algorithm
- Proposed a simple and almost cost-free method to defeat a new attack