Practical Fault Countermeasures for Chinese Remaindering Based RSA

Mathieu Ciet Marc Joye

Gemplus & CIM-PACA
Gardanne, France

FDTC '05

Outline

RSA Cryptosystem

- Setup
  - Let $N = pq$ with p, q prime
  - Let (e, d) satisfying $e \cdot d \equiv 1 \pmod{\varphi(N)}$
- Public parameters: $\{e, N\}$
- Private parameters:
  - Standard mode: $\{d, N\}$
  - CRT mode $\{p, q, d_p, d_q, i_q\}$

Signature on message m

$S = \dot{m}^d \bmod N$ where $\dot{m} = \mu(m)$

Verification

$S^e \stackrel{?}{\equiv} \mu(m) \pmod{N}$

RSA Signatures in Practice

- Deterministic paddings
  - FDH [Bellare & Rogaway, ACM CCS '93]

    $$\mu(m) = H(m) \quad \text{with } H : \{0,1\}^* \to \{0,1\}^{\log_2 N}$$

  - PKCS #1 v1.5 [RSA Labs]
- Probabilistic paddings
  - PSS [Bellare & Rogaway, EUROCRYPT '96]

    $$\mu(m) = 0 \| w \| r^* \| g_2(w)$$

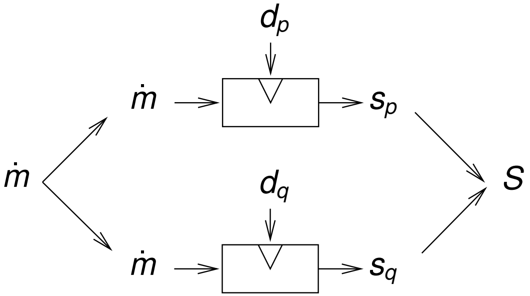
    with $w = h(m, r)$ and $r^* = g_1(w) \oplus r$
  - PKCS #1 v2.1 [RSA Labs]

Chinese Remaindering

- Computation of a signature $S = \dot{m}^d \bmod N$ using CRT
  - $s_p = \dot{m}^{d_p} \bmod p$ with $d_p = d \bmod (p-1)$
  - $s_q = \dot{m}^{d_q} \bmod q$ with $d_q = d \bmod (q-1)$

CRT formula

$$\boldsymbol{S = \mathrm{CRT}(s_p, s_q) = s_q + q[i_q(s_p - s_q) \bmod p]}$$



Flipping-Bit Attack

- Let $d = \sum_{i=0}^{\ell-1} d_i 2^i$
  - Flipping bit: $d_j \to \overline{d_j}$

$$\Rightarrow \hat{d} = \overline{d_j}\, 2^j + \sum_{\substack{i=0 \\ i \neq j}}^{m-1} d_i\, 2^i = (\overline{d_j} - d_j) 2^j + d$$

$$\Rightarrow e \cdot \hat{d} \equiv (\overline{d_j} - d_j) 2^j + 1 \pmod{\varphi(N)}$$

  - For $j = 0$ to $\ell - 1$ check if

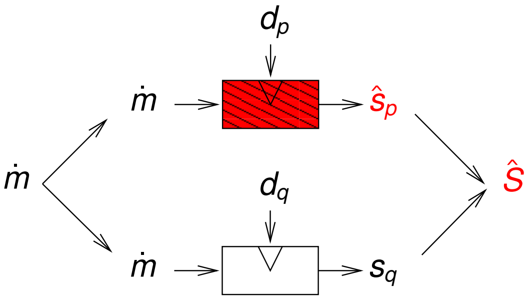
$$\hat{S}^e \equiv \mu(m)^{e \cdot \hat{d}} \equiv \begin{cases} \mu(m)^{2^j+1} \pmod{N} \Rightarrow d_j = 0 \\ \mu(m)^{-2^j+1} \pmod{N} \Rightarrow d_j = 1 \end{cases}$$

Forcing-Bit Attack

- Let $d = \sum_{i=0}^{\ell-1} d_i 2^i$
  - Forcing bit: $d_j \to 0$
  - Check whether S is a valid signature
    - if so, then $d_j = 0$
    - if not, then $d_j = 1$

- Also applies to probabilistic paddings
- Replacing d with $d^* = d + r\,\varphi(N)$ may help to prevent the attack

GCD Attack



$$\gcd(\hat{S}^e - \dot{m} \bmod N, N) = q$$

Proof. $\hat{S}^e - \dot{m} \equiv {\hat{s}_p}^e - x \not\equiv 0 \pmod{p} \iff p \nmid (\hat{S}^e - \dot{m})$
and $\hat{S}^e - \dot{m} \equiv {s_q}^e - \dot{m} \equiv 0 \pmod{q} \iff q \mid (\hat{S}^e - \dot{m})$

- Equally applies if there is a fault on i_q

Known Countermeasures

- Computing the signatures twice
  - Doubles the running time
- Verifying the signatures
  - Efficient (as public verification exponent e is small)
  - …but requires the knowledge of e

Known Countermeasures

Shamir's trick

1. $s_{p^*} = \dot{m}^d \bmod (rp)$
2. $s_{q^*} = \dot{m}^d \bmod (rq)$
3. $S = \mathrm{CRT}(s_{p^*} \bmod p, s_{q^*} \bmod q) \quad \text{iff} \quad s_{p^*} \equiv s_{q^*} \pmod{r}$

- Drawbacks
  - Requires the knowledge of d
  - Does not detect errors on CRT combination
    - e.g., fault on i_q
- Variants
  - Do not detect errors on CRT combination

Infective Computation

- Observation [Yen *et al.*, IEEE TC, 2003]
  - Decisional tests should be avoided
  - Inducing a random fault in the status register flips the value of the zero flag bit with a probability of 50%

Infective computation

Ensure that both half exponentiations are faulty whenever an error is induced:

$$\hat{S} \neq S \pmod{p} \iff \hat{S} \neq S \pmod{q}$$

BOS Algorithm

- J. Blömer, M. Otto, and J.-P. Seifert, ACM CCS 2003
- Initialization
  - For two “appropriate” randoms t_1 and t_2, precompute and store
    1. $p^* = t_1 p$, $q^* = t_2 q$, $N^* = t_1 t_2 N$
    2. $i_{q^*} = (q^*)^{-1} \bmod p^*$
    3. $d_{p^*} = d \bmod \varphi(p^*)$, $e_1 = {d_{p^*}}^{-1} \bmod \varphi(t_1)$
    4. $d_{q^*} = d \bmod \varphi(q^*)$, $e_2 = {d_{q^*}}^{-1} \bmod \varphi(t_2)$

BOS Algorithm

Input: $\dot{m}$
Output: $S = \dot{m}^d \bmod N$
In memory: $\{p^*, q^*, i_{q^*}, N^*, d_{p^*}, e_1, d_{q^*}, e_2\}$

1. Compute
    1. $s_{p^*} \leftarrow \dot{m}^{d_{p^*}} \bmod p^*$ and $s_{q^*} \leftarrow \dot{m}^{d_{q^*}} \bmod q^*$
    2. $S^* \leftarrow \mathrm{CRT}(s_p^*, s_q^*) \bmod N^*$
    3. $c_1 \leftarrow (\dot{m} - S^{e_1} + 1) \bmod t_1$ and $c_2 \leftarrow (\dot{m} - S^{e_2} + 1) \bmod t_2$
2. Return $S = (S^*)^{c_1 c_2} \bmod N$

- Shown to be insecure by D. Wagner, ACM CCS 2004

Our Algorithm

Input: $\dot{m}, \{p, q, d_p, d_q, i_q\}$
Output: $S = \dot{m}^d \bmod N$

1. For two co-prime κ-bit integers r_1 and r_2, define
   $$p^* = r_1\, p\,,\ q^* = r_2\, q\,,\ i_{q^*} = (q^*)^{-1} \bmod p^*\,,\ N = p\, q$$
2. Compute
   1. $s_{p^*} \leftarrow \dot{m}^{d_p} \bmod p^*$ and $s_2 \leftarrow \dot{m}^{d_q \bmod \varphi(r_2)} \bmod r_2$
   2. $s_{q^*} \leftarrow \dot{m}^{d_q} \bmod q^*$ and $s_1 \leftarrow \dot{m}^{d_p \bmod \varphi(r_1)} \bmod r_1$
   3. $S^* \leftarrow s_{q^*} + q^* \left(i_{q^*}(s_{p^*} - s_{q^*}) \bmod p^*\right)$
   4. $c_1 \leftarrow (S^* - s_1 + 1) \bmod r_1$
   5. $c_2 \leftarrow (S^* - s_2 + 1) \bmod r_2$
3. For an ℓ-bit integer r_3, set $\gamma \leftarrow \lfloor (r_3\, c_1 + (2^\ell - r_3)\, c_2)/2^\ell \rfloor$
4. Return $S = (S^*)^\gamma \bmod N$

Security Model

1. Fault model #1: Precise bit errors
   - The attacker can cause a fault in a single bit
   - Full control over the timing and location of the fault
2. Fault model #2: Precise byte errors
   - The attacker can cause a fault in a single byte
   - Full control over the timing but only partial control over the location (e.g., which byte is affected)
     - new faulty value cannot be predicted
3. Fault model #3: Unknown byte errors
   - The attacker can cause a fault in a single byte
   - Partial control over the timing and location of the fault
     - new faulty value cannot be predicted
4. Fault model #4: Random errors
   - Partial control over the timing and no control over the location

Security Analysis

- In Step 1, the computations are supposed to be error-free
  - Quantities available in memory (as in BOS)
  - Checks
- Order is important in Step 2
  - s_p^*, s_2, s_q^* and s_1
  - If s_p^*, s_1, s_q^* and s_2 then a long-lived fault on $\dot{m}$ (after s_1) will be undetected
    - $c_1 = c_2 = \gamma = 1$ but...
    - $\gcd(\hat{S}^e - \dot{m} \pmod{N}, N) = p$
- Heuristic security analysis
  - Only known attacks are considered

Future Work & Open Problems

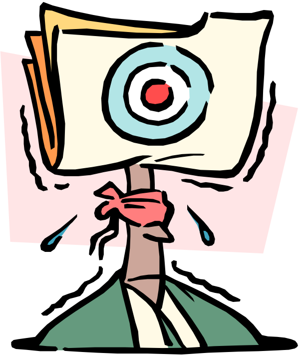
Deterministic RSA signatures

- Cryptanalyze the proposed algorithm
- Improve the proposed algorithm
- Propose a better c/measure

Probabilistic RSA signatures

- Mount a fault attack against RSA-PSS
- Prove the security of RSA-PSS in a given security model

Questions



`http://www.geocities.com/MarcJoye/`