Attacking unbalanced RSA-CRT using SPA

P.A. Fouque*, G. Martinet, G. Poupard

(with the participation of Barnaby Martinet-Fouque)

DCSSI Crypto Lab, France

(*) Ecole Normal Supérieure, France

Our goal

- Power analysis techniques (**SPA**) are very usefull tools to detect some « *events* »
- Lattice reduction algorithm (**LLL**) is a very usefull tool for « *classical cryptanalysts* »
- **SPA + LLL = new threat for smartcards**
- this paper : an example of such a combination

RSA CRT + Garner

- Goal : efficient computation of modular exponentiation with RSA modulus
- Applications :
  - RSA decryption
  - RSA signature generation
- Input :
  - modulus $n = p.q$, exponents $e.d = 1 \bmod \varphi(n)$
  - message (or ciphertext) m
- Output : $S = m^{d} \bmod n$

RSA CRT + Garner

- Pre-computation :
  - $d_p = d \bmod p-1$
  - $d_q = d \bmod q-1$
  - $u = q^{-1} \bmod p$
- Algorithm (assume $q < p$) :
  - $s_p = m^{dp} \bmod p$, $s_q = m^{dq} \bmod q$
  - $t = s_p - s_q$; if $t < 0$ then $t = t + p$
  - S (=$m^{d} \bmod n$) = $s_q + (t \cdot u \bmod p) \cdot q$ (in [0, $n-1$])

Novak's attack

- R. Novak. « *SPA-based Adaptive Chosen-Ciphertext Attack on RSA Implementation* », In PKC 2002
- **SPA** used to detect the « *event* »

```
if t < 0 then t = t + p
```

- requires $\log_2(n)/2$ **chosen** messages m_i
- → **total break** (recovers p and q)

Novak's attack

- Novak's attack applies only when the attacker can **choose** messages m_i
- Application :
  RSA decryption on « *open* » cards
- Limitation :
  Does not apply to RSA signature generation

Our attack

- **SPA** used to detect the « *event* »

```
if t < 0 then t = t + p
```

- requires **known** messages m_i
- → **total break** (recovers p and q)

  if p and q are « *unbalanced* »
- Applies to RSA decryption

  and RSA signature generation

Our attack

More precizely :

- $n = p.q$, $q \approx p / 2^{\ell}$, i.e $|p| - |q| \approx \ell$
- $2^{\ell} . |n| / \ell$ **known** messages
- reduction of $(d{\times}d)$-lattice with $d \approx |n| / \ell$

  (max $d \approx 100 \rightarrow \ell \approx |n| / 100$)

Actual example of attack

- $|n|$=1024,
  $|p|$=516, $|q|$=508 → $\ell = 8$
- observation of 69 323 signatures
  (19 hours if 1 signature / second)
  → **61** signatures with « event $t < 0$ »
- lattice reduction with LLL: 26 minutes
  → recovers $\boldsymbol{p}$ and $\boldsymbol{q}$

Details of the attack

- Garner CRT Algorithm (assume $q < p / 2^{\ell}$) :
  - $s_p = m^{dp} \bmod p$, $s_q = m^{dq} \bmod q$
  - $t = s_p - s_q$; if $t < 0$ then $t = t + p$
  - $S = s_q + (t \,.\, u \bmod p).\, q$ (in $[0, n-1]$)
- if $t < 0$, $s_p - s_q < 0$ so $s_p < s_q < q < p / 2^{\ell}$
- $S = s_p + \lambda.p$ with $s_p < p / 2^{\ell}$ (instead of $< p$)
- Problem : how to factor n if we know such S

Lattice definition

$$L = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_d \\ V_{d+1} \end{pmatrix} = \begin{pmatrix} N & 0 & \cdots & \cdots & 0 \\ 0 & N & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & N & 0 \\ -S_1 & -S_2 & \cdots & -S_d & A \end{pmatrix}$$

Lattice = set of all the linear combinations, with integer coefficients, of basis vectors $V_1, \ldots V_{d+1}$

$$L = \left\{ \sum_{i=1}^{d+1} c_i \times V_i ; (c_1, c_2, \cdots, c_{d+1}) \in Z^{d+1} \right\}$$

Lattice reduction

- *Lattice reduction* = computation of a basis that generates the same lattice and such that
  - the vectors of the basis are « ***short*** »
  - the vectors of the basis are « ***almost orthogonal*** »
- similar to *Gram-Schmidt* reduction but using integer coefficients

LLL

- Lattice reduction algorithm : **LLL** (Lenstra, Lenstra, Lovasz, 1982)
- → Use of **LLL** to find a ***short vector*** in a lattice
- → If we know that a lattice has an « *abnormally short vector* », we can use **LLL** as a « *short vector oracle* »

Attack : lattice problem

- Our problem : we know $S_i = u_i + \lambda_i.p$ with $u_i < p / 2^\ell$ (instead of $< p$)
- Consider the lattice

$$L = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_d \\ V_{d+1} \end{pmatrix} = \begin{pmatrix} N & 0 & \cdots & \cdots & 0 \\ 0 & N & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & N & 0 \\ -S_1 & -S_2 & \cdots & -S_d & A \end{pmatrix}$$

- $V^* = \lambda_1.V_1 + \lambda_2.V_2 + \ldots + \lambda_d.V_d + q.V_{d+1}$ is in L

$V^* = (-q.u_1, -q.u_2, \ldots, -q.u_d, q.A) \rightarrow ||V^*|| < q.A.\sqrt{d+1}$

Attack : lattice problem

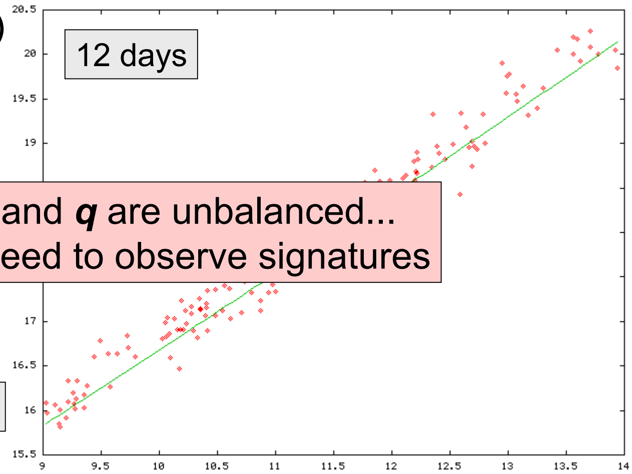
- the vector $V^* = (-q.u_1,-q.u_2,\ldots,-q.u_d,q.A)$ is in the lattice; if it is « *abnormally small* », we can hope LLL will find it
- We use an upper bound of q for A
- The last coordinate of V^* reveals q
- V^* is small if ℓ is large enough... (*proof in appendix of the paper*)
  → this is the reason for « *unbalanced RSA* »

Attack : algorithm

- ```
  d=0
  do
     d=d+1
     Observe signatures of known
          messages with SPA until
          « event t<0 » occurs
     let S_d be this signature
     let Lat be the (d+1)x(d+1)-lattice
          formed using S_1,S_2,…,S_d
  until LLL(Lat) finds V*
  return q = V*_{d+1}/A
  ```
  ```

Results with RSA 1024

log$_2$(#signatures)
(1 sig/sec)



12 days

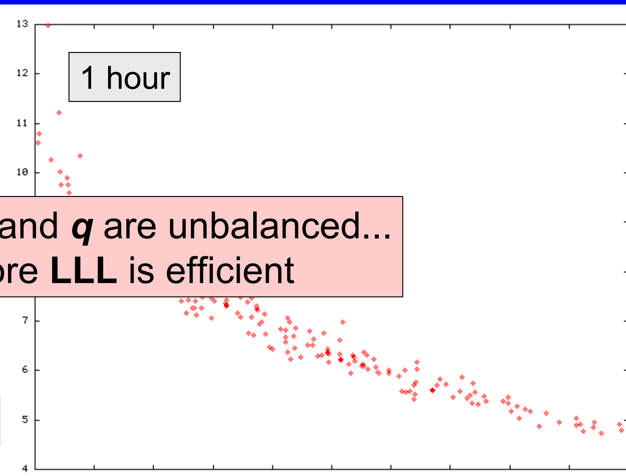
The more ***p*** and ***q*** are unbalanced...
the more we need to observe signatures

18 min

$|p|-|q| = \ell$

Results with RSA 1024

log$_2$(time LLL)



1 hour

The more ***p*** and ***q*** are unbalanced...
the more **LLL** is efficient

30 sec

$|p|-|q| = \ell$

Countermeasures

- Use « *dummy operations* »

```
t1 = t
t2 = t+p
if  t<0  t = t1 else t = t2
```

… but be careful with « *safe errors* »…

- Use « *randomization* » of *m*

  signature(m)=**signature**($m.r^e$) / r

- Use « *full randomization* » of m, p, q, d_p, d_q,...

Conclusion

- RSA implementations using Garner's CRT algorithm and unbalanced modulus must be protected against SPA …

- Lattice reduction techniques are very useful tools when considering hardware security