

# TP 03: algorithmes de chiffrement et de signature

Jean-Sébastien Coron

Université du Luxembourg

<http://www.eleves.ens.fr/home/coron/cours/cours.html>

## 1 RSA signature

1. Implement the RSA FDH scheme using the NTL library available at [www.shoup.net](http://www.shoup.net), with a modulus size of 1024 bits.
2. Implement the signature generation algorithm using the Chinese Remainder Theorem (CRT) : to compute  $s = H(m)^d \pmod N$ , compute

$$s_p = s \pmod p = H(m)^d \pmod{p-1} \pmod p$$

and

$$s_q = s \pmod q = H(m)^d \pmod{q-1} \pmod q$$

Recover  $s \pmod N$  from  $s_p$  and  $s_q$  using the CRT.

3. Assume that an error occurs during the computation of  $s_p$ , that is, an incorrect value  $s'_p \neq s_p$  is computed while  $s_q$  is correctly computed. Show how to recover the factorization of  $N$  from  $s$ . How could such error be detected ? Propose and implement a simple method to detect such error.

## 2 El-Gamal encryption

The El-Gamal encryption scheme works as follows :

Key generation :

- Generate a prime  $p$  and a generator  $g$  of  $\mathbb{Z}_p^*$ .
- Generate a random  $a \in [0, p - 2]$ .
- Let  $h = g^a \pmod p$
- Public-key:  $(g, h)$ . Private-key:  $a$

To encrypt  $m \in \mathbb{Z}_p$ , generate a random  $r \in [0, p - 2]$ . The ciphertext is then :

$$c = (g^r, m \cdot h^r)$$

To decrypt, write  $c = (c_1, c_2)$  and recover  $m = c_2 \cdot (c_1)^{-a} \pmod p$

1. Show that decryption works
2. Implement the El-Gamal encryption scheme
3. Show that the El-Gamal encryption scheme is vulnerable to a simple chosen-ciphertext attack