Niveau : 2 **Orange**

Attaque par faute de la signature RSA

1 Une implémentation particulière de RSA

On va utiliser pour signer, un système RSA, de module n produit des deux nombres premiers p et q, dont la clé publique est e et la clé privée est d. Ainsi

$$ed \equiv 1 \pmod{\lambda(n)},$$

où

$$\lambda(n) = \text{lcm}(p-1, q-1)$$

est la fonction de Carmichael.

Lors de la signature le propriétaire de la clé doit calculer une expression du type :

$$S = c^d \mod n.$$

Pour cela il peut, soit faire un calcul direct, soit calculer :

$$S_1 = c^d \mod p,$$

$$S_2 = c^d \mod q,$$

et reconstituer S grâce au théorème des restes chinois :

$$S = uS_2p + vS_1q \mod n,$$

où (u, v) vérifie :

$$up + vq = 1.$$

Cette dernière méthode est plus rapide. À cause du petit théorème de Fermat on peut écrire :

$$S_1 = c^{d_1} \mod p,$$

$$S_2 = c^{d_2} \mod q,$$

avec :

$$d_1 = d \mod (p-1),$$
$$d_2 = d \mod (q-1).$$

Cette méthode est à peu près 4 fois plus rapide que le calcul direct pour effectuer une signature. Mais elle est sensible à l'attaque par faute suivante.

2 Attaque par faute

Supposons que l'un et un seul des deux calculs de S_1 ou S_2 soit faux. Supposons par exemple que ce soit celui de S_1. Alors le résultat S' faussement calculé avec S'_1 au lieu de S_1 et S_2 vérifie :

$$S' \equiv S_2 \quad (q),$$
$$S' \not\equiv S_1 \quad (p).$$

Si bien que :

$$(S'^e - c) \mod n \equiv 0 \quad (q),$$
$$(S'^e - c) \mod n \not\equiv 0 \quad (p).$$

En conséquene,

$$\gcd(n, (S'^e - c) \mod n) = q.$$

Ainsi, grâce à la signature erronée S' du message connu c, et à la clé publique (e, n), un attaquant est capable de factoriser n. Pour monter cette attaque il suffit de provoquer une erreur de calcul au bon moment. Sur une carte à puce cela est relativement simple.