Master CSI 2 2018-2019

Cryptanalyse — 4TCY902U
Responsable : G. Castagnos

TP 3 — Attaques par réduction de réseaux

Attaque sur RSA par la méthode de Coppersmith

1 Implanter l'algorithme de génération de clefs de RSA avec l'exposant public $e = 3$: il prend un entier k en entrée et ressort la clef publique ($N = pq$, $e = 3$) et la clef privée d telles que p, q sont deux premiers distincts aléatoires de k bits avec $\varphi(n) = (p-1)(q-1)$ premier avec 3 et d l'inverse de 3 modulo $\varphi(n)$.

L'objectif du TP est d'implanter l'attaque de Coppersmith sur les messages dont les bits de poids forts sont connus.

Soit N un module public RSA. On note M un message clair dont on cherche les t bits de poids faibles et C son chiffré par RSA avec la clef publique ($N, e = 3$). On pose $M = M' + M_0$ avec M' connu et M_0 inconnu de t bits. On note f le polynôme de $\mathbf{Z}[x]$, $f(x) = (x + M')^3 - C$.

2 Générer de tels éléments $N, M = M' + M_0, C$ et f pour N de 1024 bits et $t = 240$ bits. Vérifier que M_0 est une (petite) racine de f modulo N.

Nous allons retrouver M_0 par réduction de réseau en suivant la démonstration du théorème de Coppersmith qui calcule une petite racine d'un polynôme f modulo un entier à la factorisation inconnue.

Soit m un paramètre. On note $k = deg\, f$ (pour notre application, $k = 3$). On considère une famille de polynômes de la forme

$$g_{i,j}(x) = N^{m-i} x^j f(x)^i \quad \text{pour} \quad 0 \leqslant i \leqslant m-1 \quad \text{et} \quad 0 \leqslant j \leqslant k-1.$$

On rajoute à cette famille le polynôme f^m, on obtient ainsi $d := mk + 1$ polynômes.

3 Montrer que toute combinaison linéaire à coefficients entiers de cette famille admet la racine M_0 modulo N^m.

Soit X un entier positif avec $M_0 < X$. Si g est un polynôme $g(x) = \sum_{i=0}^{k} g_i x^i$, on note $g(xX)$ le vecteur $(g_0, g_1 X, g_2 X^2, \ldots, g_k X^k)$.

On note L le réseau de dimension d engendré par la matrice carrée G dont les lignes sont formées par les vecteurs $g_{i,j}(x\mathrm{X})$ pour $0 \leqslant i \leqslant m-1$ et $0 \leqslant j \leqslant k-1$ et le vecteur $f^m(x\mathrm{X})$.

4 Montrer que

$$det\,\mathrm{L} = \mathrm{N}^{k\frac{m(m+1)}{2}}\mathrm{X}^{\frac{km(km+1)}{2}}.$$

5 Écrire une fonction qui prend un entrée f, N, X et m et qui ressort la matrice G. Vérifier que l'on obtient bien le bon déterminant sur l'exemple construit en question 2 (on pourra prendre $m = 2$ et $\mathrm{X} = 2^t$).

6 Montrer que si

$$2^{\frac{d-1}{4}}(det\,\mathrm{L})^{\frac{1}{d}} \leqslant \frac{\mathrm{N}^m}{\sqrt{k+1}}, \tag{1}$$

alors on peut retrouver M_0 en cherchant une racine dans $\mathbf{Z}$ du polynôme construit à l'aide du premier vecteur d'une base LLL réduite de L.

7 On pose $\mathrm{X} = \mathrm{N}^\alpha$. On néglige les quantités ne dépendant pas de N. Montrer que l'inégalité (1) est alors vérifiée si

$$\alpha \leqslant \frac{m-1}{km+1}.$$

Cette fraction tendant vers $1/k$ quand m tend vers plus l'infini, on trouvera bien toute racine $|x_0| \leqslant \mathrm{N}^{1/k}$ de f modulo N pour m suffisament grand, comme l'annonce le théorème de Coppersmith. Dans notre cas d'application sur RSA, cela signifie que si on connaît au moins de l'ordre de 2/3 des bits de M alors on pourrait retrouver l'intégralité des bits de M à partir de son chiffré avec $e = 3$ en considérant un réseau suffisamment grand.

8 Retour à l'exemple construit en question 2. Trouver M_0 avec Sage en appliquant LLL sur la matrice G construite en question 5, toujours avec $m = 2$ et $\mathrm{X} = 2^t$. Augmenter ensuite la valeur de $t < 1024/3 \approx 341$, c'est à dire le nombre de bits inconnus et recommencer l'attaque en faisant croître la valeur de m (et par conséquent la dimension du réseau considéré).