Master CSI 2 2019-2020

Cryptanalyse — 4TCY902U
Responsable : G. Castagnos

TP 2 — Attaques par réduction de réseaux

1 Lagrange-Gauss

$\boxed{1}$ Programmer l'algorithme de réduction de Lagrange-Gauss (on ne renverra pas nécessairement une base directe). Application : réduire le réseau de $\mathbf{R}^2$ de base $b_1 = (6,1), b_2 = (10,3)$.

$\boxed{2}$ Soit p un nombre premier tel que $p \equiv 1 \pmod 4$. On rappelle que dans ce cas -1 est un carré modulo p. On note s un entier tel que $s^2 \equiv -1 \pmod p$. On considère le réseau L de $\mathbf{R}^2$ de base

$$\begin{pmatrix} 1 & s \\ 0 & p \end{pmatrix}.$$

Par un résultat dû à Hermite, on sait qu'il existe un vecteur $u = (u_1, u_2) \in \mathrm{L}$ tel que $\|u\| \leqslant \frac{4}{3}\sqrt{det\,\mathrm{L}}$.

En déduire un algorithme polynomial prenant en entrée p et retournant deux entiers a et b tel que $a^2 + b^2 = p$.

Application : Décomposer 1267650600228229401496703205653 (le plus petit premier congru à 1 modulo 4 supérieur à 2^{100}) en somme de deux carrés.

2 Attaque sur Merkle-Hellman

Cette cryptanalyse consiste à résoudre un problème de type sac à dos, le ***subset sum problem***, grâce à l'algorithme LLL. Le *subset sum problem* consiste étant donné $a_1, \dots, a_n$ et s des entiers naturels à déterminer s'il existe un sous-ensemble $\mathrm{I} \subset \{1, \dots, n\}$ tel que

$$s = \sum_{i \in \mathrm{I}} a_i.$$

Ce problème est NP-complet.

Pour utiliser ce problème pour construire un système de chiffrement à clef publique, il faut une instance faible, c'est à dire une instance que l'on sera résoudre connaissant une trappe, la clef secrète. On dit que $a_1, \dots, a_n$ est à **super croissance** si $a_j > \sum_{i=1}^{j-1} a_i$, pour $2 \leqslant j \leqslant n$, Dans ce cas, on peut résoudre efficacement le *subset sum problem* avec l'algorithme glouton.

Algorithme glouton pour le *subset sum problem*

Entrée : $(a_1, \dots, a_n)$ et s
Déterminer le plus grand des a_i tel que $a_i \leqslant s$
$s \leftarrow s - a_i$
Recommencer tant que $s \geqslant min(a_i)$

$\boxed{3}$ Montrer que si $(a_1, \dots, a_n)$ est à super croissance, à la fin de l'algorithme, on a $s \neq 0$ si et seulement si s ne se décompose pas. Si $s = 0$, les a_i utilisés donnent la décomposition. Il ne sera pas utile pour la suite de programmer cet algorithme.

Le système de chiffrement de Merkle-Hellman (1978) consiste à masquer une suite super croissante.

Cryptosystème de Merkle-Hellman

Génération des clefs :
n un paramètre de sécurité
Générer aléatoirement $a_1, \dots, a_n$ à super croissance, avec $2^{n-1} < a_1 < 2^n$ puis $\sum_{i=1}^{j-1} a_i < a_j < 2^{n+j-1}$ pour $j = 2, \dots, n$
Générer aléatoirement N tel que $\sum_{i=1}^{n} a_i < \mathrm{N} < 2^{2n}$
Générer aléatoirement $\mu \in (\mathbf{Z}/\mathrm{N}\mathbf{Z})^\times$
Poser $b_i \equiv a_i\mu \pmod{\mathrm{N}}$ pour $i = 1, \dots, n$
Retourner $(b_1, \dots, b_n)$ comme clef publique et N, μ, $(a_1, \dots, a_n)$ comme clef privée

Chiffrement de $m = (m_1, \dots, m_n) \in \{0, 1\}^n$
Retourner $c = \sum_{i=1}^{n} m_i b_i$.

Déchiffrement de c
Poser $s = c\mu^{-1} \mod \mathrm{N}$.
Résoudre le *subset sum problem* $(a_1, \dots, a_n)$, s avec l'algorithme glouton et retourner la solution

$\boxed{4}$ Montrer que ce chiffrement est correct, c'est à dire que le déchiffrement avec une clef secrète d'un chiffrement de m avec la clef publique associée retourne bien m.

$\boxed{5}$ Implanter les algorithmes de génération des clefs et de chiffrement de Merkle-Hellman. Il ne sera pas utile pour la suite de programmer le déchiffrement.

On considère une attaque, due à Lagarias et Odlyzko, consistant, étant donné un chiffré c et la clef publique $(b_1, \dots, b_n)$ à retrouver le message clair m, c'est à dire retrouver $(m_1, \dots, m_n) \in \{0, 1\}^n$ tel que $c = \sum_{i=1}^{n} m_i b_i$ (c'est à dire une attaque à clairs choisis contre la notion de sens-unique du chiffrement). On doit donc résoudre une instance du *subset sum problem a priori* difficile.

À c, $(b_1, \dots, b_n)$ on associe le réseau $\mathrm{L} \subset \mathbf{R}^{n+2}$ de dimension $n+1$ engendré par

$$\mathrm{A} := \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & \mathrm{K}b_1 \\ 0 & 1 & 0 & 0 & \dots & 0 & \mathrm{K}b_2 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & \mathrm{K}b_n \\ 0 & 0 & \dots & 0 & 0 & 1 & -\mathrm{K}c \end{pmatrix}$$

où K est un entier.

Soit $e_1, \dots, e_{n+1}$ les lignes de A. À une solution du ***subset sum problem***, $c = \sum_{i=1}^{n} m_i b_i$, on fait correspondre le vecteur $m = m_1 e_1 + \dots m_n e_n + e_{n+1} \in \mathrm{L}$.

$\boxed{6}$ Montrer que $\|m\|^2 \leqslant n+1$.

$\boxed{7}$ On pose K tel que $\mathrm{K}^2 > n+1$. Réciproquement, soit $v \in \mathrm{L}$ et $\|v\|^2 \leqslant n+1$. Montrer que les n premières coordonnées de v fournissent une solution du ***subset sum problem*** si $v_{n+1} = 1$.

Les solutions du ***subset sum problem*** donnent donc des vecteurs courts et réciproquement des vecteurs courts peuvent donner des solutions.

$\boxed{8}$ Cryptanalyse : retrouver le message m à partir d'un chiffré c. Pour cela, réduire avec LLL le réseau engendré par la matrice A. Tester pour un paramètre de sécurité $n = 10, 20, 30 \dots$

Pour appliquer LLL sur une matrice M d'entiers avec Sage, utiliser la commande `M.LLL()`.

La densité d'une instance du ***subset sum problem*** est défini par

$$d = \frac{n}{log\, max(b_i)}.$$

On sait résoudre presque tous les ***subset sum problem*** en temps polynomial en utilisant LLL s'ils sont à faible densité ($d < 1/n$, Lagarias-Odlyzko 1985). De plus Lagarias-Odlyzko ont montré que pour $d < 0.645$ alors presque tous les ***subset sum problem*** se réduisent à un problème SVP. Pour le système de Merkle-Hellman on a $d \approx \frac{n}{2n} \approx \frac{1}{2}$.