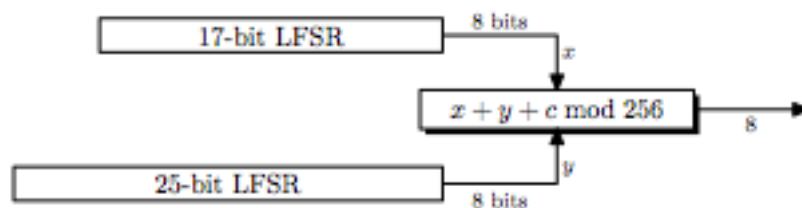


Attaque sur CSS

Dans ce TP vous pouvez réutiliser les fonctions `LFSR_Step` et `LFSR` issues du TP3 (récupérez le fichier `tp5_code.py`).

Le Content Scrambling System (CSS) est utilisé pour sécuriser le contenu des disques DVDs. CSS souffre de plusieurs faiblesses. Nous discutons ici du stream cipher CSS utilisé pour chiffrer les contenus des films. CSS a été conçu dans les années 1980 quand le chiffrement exportable était restreint à 40 bits. En conséquence, CSS utilise des clés secrètes de 40 bits.

Le PRG de CSS utilise deux LFSRs.



Algorithm 1 CSS

Entrée : seed $s \in \{0, 1\}^{40}$

Écrire $s = s_1 || s_2$ où $s_1 \in \{0, 1\}^{16}$ et $s_2 \in \{0, 1\}^{24}$

Charger $1 || s_1$ dans le LFSR de 17 bits

Charger $1 || s_2$ dans le LFSR de 25 bits

$c \leftarrow 0$

▷ carry bit

for $i = 1$ à N **do**

 exécuter les deux LFSRs pendant 8 cycles et obtenir $x, y \in \{0, 1\}^8$

 traiter x et y comme des entiers dans $0 \dots 255$

Renvoyer $z = x + y + c \bmod 256$

 si $x + y > 255$, alors $c \leftarrow 1$, 0 sinon $c \leftarrow 0$

▷ carry bit

end for

Le PRG retourne un octet à chaque itération. L'ajout du bit à 1 pour s_1 et s_2 garantit que les LFSRs ne seront pas initialisés à 0. Les polynômes de rétroaction des LFSR de 17 et 25 bits sont respectivement :

$$\begin{cases} P_{17} = X^{17} + X^3 + 1 \\ P_{25} = X^{25} + X^{22} + X^{21} + X^{13} + 1 \end{cases}$$

Ces deux polynômes sont primitifs.

Question 1. Programmez le PRG de CSS. Pour représenter simplement la sortie, utilisez une liste d'entiers python entre 0 et 255.

Question 2. Soit x_1, x_2, x_3 les 3 premiers octets de sortie du LFSR de 17 bits. Montrez que l'état initial s_2 du second LFSR peut être obtenu en fonction de (z_1, z_2, z_3) (les trois premiers octets de sortie du PRG) et (x_1, x_2, x_3) .

Question 3. *On suppose que l'on connaît les 6 premiers octets z_1, z_2, \dots, z_6 . Montez une attaque du type guess-and-determine de complexité 2^{16} .*

Question 4. *Programmer l'attaque contre ce générateur.*

Question 5. *On suppose maintenant que l'on ne connaît que les 5 premiers octets z_1, z_2, \dots, z_5 . Monter une attaque de complexité 2^{17} .*

Question 6. *Programmer l'attaque contre ce générateur.*