Projet Cryptographie – IN603
Questions théoriques – Id Es Q4 et Q5

AUCLAIR Lucas
MEDIOUNI Soumaya
LDDBI

Question 4 : *Soit x1, x2, x3 les 3 premiers octets de sortie du LFSR de 17 bits. Montrer que l'état initial s2 du second LFSR peut être obtenu en fonction de (z1, z2, z3) et (x1, x2, x3).*

On sait que :

LFSR17 => (x1, x2, x3, …) et LFSR25 => (y1, y2, y3, …), tels que **tous les 8 cycles l'algorithme CSS renvoie z = x + y + c mod 256 avec (z1, z2, z3, …)** les résultats du chiffrement et donc les bits du chiffré ainsi que **c initialisé à 0 pour le premier octet et = 1 si x + y > 255 sinon = 0** ;

Les 2 LFSR sont initialisés avec 40 (16 + 24) bits de la clé secrète et **s1[16] = s2[24] = 1** tels que s1 et s2 sont respectivement les états initiaux du LFSR17 et du LFSR25 ;

Conséquemment :

Pour obtenir s2 à partir de (z1, z2, z3) et (x1, x2, x3), il s'agit globalement de retourner/inverser l'algorithme de chiffrement CSS et son équation principale.

Tel que, **zi – xi – c mod 256 = yi avec i appartenant à {1, 2, 3, …}**, **c = 0 pour i = 1** et permet donc de déterminer en partant de i = 1 la retenue et celles à venir pour chiffrer les octets.
En effet, une fois yi obtenu à l'aide de xi et zi, **il suffit de regarder si xi + yi > 255 pour mettre en place la retenue c à 1 ou à 0 pour les prochains octets** (comme dans le chiffrement classique).

On fait 3 fois ces opérations, donc pour (z1, z2, z3) et (x1, x2, x3) afin d'obtenir s2 **comme 3 x 8 = 24 (s2[0, …, 23]) et que** les 2 LFSR sont initialisés avec 40 (16 + 24) bits de la clé secrète et **s1[16] = s2[24] = 1** tels que s1 et s2 sont respectivement les états initiaux du LFSR17 et du LFSR25.

Ainsi, **il ne reste plus qu'à transformer (y3, y2, y1) en binaire**, à ajouter **1 en position s2[24]** et **on obtient (1, binary(y3), binary(y2), binary(y1)) soit s2 l'état initial du LFSR25** avec binary une fonction quelconque pour transformer un entier en son correspondant en binaire.
Avoir les octets suivants permettrait de vérifier que l'on a bien le bon état initial s2.

Question 5 : *On suppose que l'on connaît les 6 premiers octets z1, z2, . . ., z6. Décrire (dans le document pdf) une attaque de complexité 2^16 qui permet de récupérer l'initialisation du générateur, en exploitant le point 4 ci-dessus.*

On sait que :

cf. Q4) aucun changement

D'après la Question 4, on peut obtenir l'état initial s2 du LFSR25 à partir des 3 premiers octets du LFSR17 (x1, x2, x3) et du CSS (z1, z2, z3). En exploitant, la procédure décrite dans la **Question 4** et **en séparant les 6 premiers octets du CSS en 2 parties**, tel **que (z1, z2, z3) servent à trouver s2 ainsi que la clef secrète** et **(z4, z5, z6) servent à vérifier qu'on a la bonne**, alors **on peut déterminer la clé secrète du générateur du chiffrement CSS** à partir de ses 6 premiers octets **avec une complexité de 2^16** (normalement ce n'est pas commun d'avoir 6 octets d'affilés comme ça).

Pour ce faire :

1- **On devine au hasard l'état initial s1 du LFSR17**
2- **On récupère les 3 premiers octets et on applique la procédure de la Question 4** afin d'avoir les 3 premiers octets du LFSR25 ainsi que son état initial s2
3- On récupère les 3 octets suivants de LFSR17 (x4, x5 ,x6) et LFSR25 (y4, y5, y6) selon le s1 aléatoire, **on applique CSS (fait l'opération pour avoir z4, z5 et z6) et on compare avec les octets (z4, z5, z6) qu'on avait initialement** afin de déterminer si on a bien deviné s1.
4- **Si on a bien deviné s1, alors on a la clef secrète de 40 bits** s $\in$ {0, 1}^40 tel que s = s1$\nparallel$s2 où s1 $\in$ {0, 1}^16 et s2 $\in$ {0, 1}^24. **Sinon, on devine au hasard à nouveau s1 et on refait les étapes décrites précédemment**, et ce jusqu'à avoir une correspondance entre les octets calculés et ceux récupérés.

Complexité :

On connaît s1[16] = 1, donc l'état initial s1 du LFSR17 n'est inconnu que pour 16 bits de positions [0, …, 15]. Dans le pire des cas, nous devons donc tester par brut force 2^16 possibilités pour le LFSR17. Ainsi, l'attaque décrite a bien une complexité de 2^16.