Université Côte d'Azur — L3 Info, L3 MI, L3 ST
Cryptographie et calcul symbolique — 2023-24

TD n° 3

Cryptographie à clef secrète

Exercice 1) Chiffre de Vernam

1. A quelles conditions d'usage ce chiffre est-il un chiffre *parfait* ? Pourquoi ?
2. Comment transposeriez-vous ce chiffre quand l'alphabet comporte 26 lettres ?
3. Le texte chiffré $c = 111\ 101\ 100\ 101\ 001\ 100\ 010\ 110$ a été obtenu en faisant un $\oplus$ du texte clair m avec une suite z de bits générés en utilisant un *générateur congruentiel* basé sur la congruence linéaire :

$$z_0 = 2$$

$$z_{i+1} \equiv 5z_i + 7 \mod 8$$

Les entiers z_i ainsi générés sont interprétés comme des nombres sur 3 bits $b_{i_2}b_{i_1}b_{i_0}$ de façon telle que $z_i = b_{i_2}2^2 + b_{i_1}2 + b_{i_0}$. Ainsi :

$$z = b_{0_2}b_{0_1}b_{0_0}\ b_{1_2}b_{1_1}b_{1_0}\ b_{2_2}b_{2_1}b_{2_0}\ \ ...$$

$$c = m \oplus z$$

Les 3 premiers blocs du texte clair m sont 101 100 000, déterminez le reste.

Exercice 2) LFSR (*Linear Feedback Shift Register*) Les LFSR sont des générateurs pseudo-aléatoires dont la sortie peut être utilisée comme clef secrète. En français ce sont les *registres linéaires à décalage rebouclés*. Ils généralisent les générateurs congruentiels de sorte que :

$$x_n = (a_1x_{n-1} + ... + a_kx_{n-k}) \mod 2$$

avec x_0, ..., x_{k-1} donnés. On les représente sous forme polynomiale :

$$\Pi(X) = X^k - a_1X^{k-1} - ... - a_k$$

Calculez les 10 premiers *bits aléatoires* produits par le LFSR :

$$\Pi = X^4 + X^3 + X^2 + 1$$

sur les valeurs initiales $(0, 1, 1, 0)$.

Exercice 3) Déchiffrement de DES On s'intéresse à la structure de cet algorithme de chiffrement afin de comprendre comment se produit le déchiffrement.

1. Rappelez le schéma de Feistel sur lequel est basé cet algorithme ;
2. Montrez qu'il n'est pas nécessaire d'inverser f pour inverser un *tour* de DES ;
3. En déduire l'algorithme de déchiffrement du DES.

Exercice 4) Blocs et mode de chaînage

1. Représentez sous forme schématique les modes de chaînage ECB et CBC.
2. Une erreur s'est produite lors de la transmisssion du premier bloc chiffré c_1 et c'est le bloc c_1' qui a été reçu à la place. Dites, pour chaque mode, quels seront les blocs qui seront correctement déchiffrés.

Exercice 5) Mots de passe UNIX

1. Quelle est la raison profonde à l'ajout de *sel* dans le calcul de l'empreinte d'un mot de passe ?
2. A quoi servent les 25 itérations de DES pour chiffrer un mot de passe ?

Exercice 6) Partage de secret Comment partager un secret s en 3 morceaux m_1, m_2 et m_3 de sorte qu'il est nécessaire de réunir ces 3 informations pour reconstituer le secret ? *(indication : recourir à l'arithmétique modulaire).*

Exercice complémentaire

Exercice 7) BBS L'algorithme BLUM BLUM SHUB (1968), du nom de ses inventeurs, permet d'engendrer des *nombres pseudo-aléatoires*. C'est même un générateur cryptographiquement sûr, compte tenu que le problème de la factorisation est un problème difficile. Il repose sur la suite :

$$x_{n+1} = (x_n)^2 \mod M$$

— choisir deux nombres premiers p et q congrus à $3 \mod 4$ puis calculer l'entier $M = p.q$;
— choisir un entier $x < M$ aléatoire et premier avec M ;
— on pose $x_0 = x^2 \mod M$ comme graine pour le générateur ;
— le i^{e} bit b_i pseudo-aléatoire est le bit de poids faible de x_i, il vérifie $b_i = x_i \mod 2$.

L'avantage de cette méthode est que le calcul peut se faire sans générer tous les bits. Ce générateur reste trop lent pour être utilisé pour chiffrer mais sert à la génération de clefs.

Application numérique : prenons $p = 7$ et $q = 11$ avec $x = 5$, quels sont les 4 prochains bits engendrés ?