UPHF - INSA
Cryptographie

3^e année ICy
Année 2022 - 2023

TP 1 : chiffrement DES

F. Goichot

Le langage est imposé : Python. L'environnement fortement conseillé est Jupyter. La clarté de votre compte rendu sera un élément déterminant d'évaluation : dans des cellules `markdown` (de préférence en utilisant le codage LaTeX pour les objets et formules mathématiques) et/ou sous forme de chaînes de documentation (*doc-string*) et commentaires dans le code, expliquez ce que vous faites.

Votre travail sera à déposer sur `Moodle` impérativement en fin de séance. Le fichier aura pour nom `TP1-DES-NOM1-NOM2`, où `NOM1` et `NOM2` sont les noms des deux étudiant(e)s constituant le binôme.

Le but *final* de la séance est de créer, tester et vérifier une fonction `des(k,p)` dont les paramètres sont `k`, clé de session (au sens de DES bien sûr) et `p`, message en clair. Cette fonction retourne le résultat du chiffrement de `p` avec la clé `k`. Il faudra aussi la fonction de déchiffrement.

Le message `p` est simplement un bloc de 64 bits : on ne s'intéresse pas à la conversion d'un message texte en tels blocs. Par contre, comme on aura beaucoup de messages (cryptés ou non) et de clés (de même format, ou de 48 bits pour les clés de tours) à afficher, il serait utile de commencer par créer une fonction qui affiche un bloc d'octets de façon lisible : un octet par ligne.

Il vous est vivement conseillé de décomposer le problème en autant de fonctions que nécessaire. Par exemple pour créer un message en clair (aléatoire), pour créer une clé de session (aléatoire aussi, mais qui soit bien une clé DES), etc. Peut-être n'aurez-vous pas le temps de tout programmer, il est plus important d'avoir la décomposition complète, avec le rôle de chaque fonction.

Vous trouverez dans `Moodle` le document Pour_TP_DES aux formats Jupyter (`ipynb`) et Python simple. Il contient les différentes constantes de DES (permutation initiale, etc) que vous n'aurez donc qu'à copier/coller dans votre propre carnet. Le document contient aussi un test global, sur lequel vous pourrez contrôler votre programme, et un test intermédiaire portant uniquement sur la première clé de tour et le résultat de la première fonction de chiffrement.

S'il vous reste du temps après avoir ainsi programmé et testé DES, vous êtes invités à tester la résistance de DES à la cryptanalyse différentielle : si on modifie un seul bit du message en clair, comment est modifié le message crypté ?

Voilà. Bon travail !