Algorithmes et structures de données avancés : TD 11

L'algorithme Merkle-Hellman

Dans ce TP, vous allez élaborer des programmes qui sont capables d'encrypter et de décrypter des messages à l'aide de l'algorithme de Merkle-Hellman. Cet algorithme est basé sur le principe des clés asymmetriques. Il est remarquable que la sécurité de cet algorithme Merkle-Hellman est basé sur le fait que les problèmes NP complet ne peuvent pas etre résolu pour de grandes longueur de données. La longueur de données dans le programme ci-dessous est spécifié par la constante *nn*, est pour assurer un bon niveau de sécurité, il faut choisir un *nn* supérieur à 2^20 (on dit aussi dans le programme ci-dessous).sécurité des algorithmes

Exercice 11.1 *Compréhension*

1. Essayer de comprendre le programme suivant que vous pouvez télécharger sur http://www.labri.fr/∼preuter/asda2009/tp11.py
2. Notez que vous disposez des fonctions `ord` pour passer d'un caractère à un nombre entre 0 et 255, et de la fonction `chr` pour passer d'un nombre de 0 à 255 à un caractère.
3. Les fonctions int2bin et bin2int permettent de convertier un nombre du système décimal en binaire, et l'inverse, respectivement.

Exercice 11.2 *Crypter un caractère*

1. Appeler la fonction crypter pour crypter le caractère "A".
2. Ecrire un programme qui demand à l'utilisateur de saisir un mot, et cryptez-le lettre par lettre en appelanbt la fonction crypter.

Exercice 11.3 *Algorithme de decryptage pour un code*

1. Elargir le programme et créer une fonction `def decrypter(chiffre, a, Ainv, N):` qui permet de decrypter un chiffre avec la clé publique `a`, `Ainv`, et `N`. La fonction renverra une chaîne de caractère.

Exercice 11.4 *Décrypter un mot*

1. Elargir le programme pour que l'utilisateur peut entrer un message, et que le programme affiche le message crypté à l'écran.

Exercice 11.5 *Trouver mon message.*

Décryptez le message suivant avec la clé privé :

```
4059 5041 3585 2461 2751 2992 4448 5041 2461 3091
```

Exercice 11.6 *Trouver un message sans clé privé.*

Imaginez vous etes en possession de la clé publique, mais pas de la clé privé, et vous trouvez un message crypté. Ecrire une fonction qui retrouve le message en décryptant le messgae crypté. *Remarque :* Cette fonction doit tester tout les permutations possibles du problème sac à dos de la clé publique ... Ceci est possible pour une petite clé (n ¡)