(Cryptographie)

Exercice 1 : Sac à dos de Merkle Hellman

Partie 1

Alice et Bob utilisent le cryptosystème du sac à dos de Merkle-Hellman pour communiquer. L'alphabet utilisé est l'alphabet usuel, chaque lettre étant codée par un entier écrit en binaire avec cinq composantes $(\varepsilon_4\varepsilon_3\varepsilon_2\varepsilon_1\varepsilon_0)_2$ (on a A=$(00000)_2$, · · · , Z=$(11001)_2$). Les unités de message sont des mots de trois lettres. Alice choisit la suite d'entiers A=(4, 5, 12, 23, 45), ainsi que *m = 400 et t = 381*.

1. Vérifier que ces données sont conformes au principe d'utilisation de ce cryptosystème.
2. Déterminer la clé publique et la clé privée d'Alice.
3. Bob veut envoyer à Alice le message ***OUI***. Indiquer le procédé qu'il doit suivre et comment Alice retrouve-t-elle le message.

Partie 2

La clé privée de Bob est [3, 4, 9, 19, 38, 77], avec t = 27 et m = 155.

1) Calculer la clef publique de Bob.
2) Comme la clef publique de Bob comporte 6 nombres, Alice regroupe les bits par paquets de 6, et au besoin elle peut ajouter des bits aléatoires pour obtenir un nombre de bits multiple le 6.
   Donner la chaîne binaire à chiffrer pour le mot "***merci***" :
3) Calculer l'inverse de t modulo m.
4) Alice transmettre le message chiffré suivant : [193, 200, 349, 389].
   Donner le message clair.

a	00000	g	00110	m	01100	s	10010	y	11000
b	00001	h	00111	n	01101	t	10011	z	11001
c	00010	i	01000	o	01110	u	10100	,	11010
d	00011	j	01001	p	01111	v	10101	.	11011
e	00100	k	01010	q	10000	w	10110	?	11100
f	00101	l	01011	r	10001	x	10111		11101

Partie 3

1) Écrire une fonction ***int somme_tab(int A[], int n)*** qui permet de retourner la somme des n premiers termes d'un tableau d'entiers A.
2) Ecrire une fonction ***void super_croissante(int A[],int n)*** qui permet de créer une suite d'entiers super-croissante dans le tableau A.
3) Ecrire une fonction ***void decomposition(int A[],int C[],int n,int k)*** qui reçoit en paramètre un ensemble A super-croissante de taille n et un entier k et qui permet de remplir le tableau C par la décomposition binaire de k dans A.
4) Ecrire une fonction ***int composition(int A[],int C[],int n)*** qui reçoit en paramètre un ensemble A super-croissante de taille n et le tableau C contenant la décomposition binaire de k dans A et qui renvoi la valeur de k.