TD2 — Chiffrement symétrique I

Ex1. Sécurité parfaite L'adjudant Durand du 28^{ieme} RT propose d'améliorer le chiffrement par masque jetable. En effet, il a observé que lorsque le masque comporte de grandes plages de 0, le message circule en clair ! Il propose plutôt que de tirer la clé uniformément parmi les clés de n bits, de ne garder que les clés qui contiennent au moins 25% de bits à 1.

1. Formaliser le chiffrement du masque jetable avec cette nouvelle contrainte, en utilisant les notations du cours ($\mathcal{M}$, Gen, Enc, Dec).
2. Cette nouvelle manière de chiffrer est-elle parfaitement sûre ?
3. Démontrer votre affirmation en utilisant la notion de schéma parfaitement indistinguable.

Ex2. Chiffrement par flot Dans cet exercice, on s'intéresse aux schémas de chiffrement par flot utilisant un état de taille fixe (comme RC4 ou les algorithmes de type FSR).

1. Rappeler les définitions d'un algorithme de chiffrement par flot, en supposant qu'un état est codé sur k bits.
2. Démontrer que la suite de bits générée est toujours ultimement périodique et donner une borne sur cette période et la longueur de la transitoire.
3. Comparer cette façon de chiffrer d'une part au chiffrement par XOR et d'autre part à l'utilisation d'un masque jetable.

Ex3. Mini RC4

Dans cet exercice, les messages sont constitués d'une suite d'entiers de 0 à 7 codés sur 3 bits. Chaque caractère est codé par deux entiers successifs selon le codage décrit à la fin de la fiche. Ainsi le mot `LICORNE` est codé 1, 3, 1, 0, 0, 2, 1, 6, 2, 1, 1, 5, 0, 4. L'algorithme RC4 décrit en cours est modifié pour travailler avec des mots de 3 bits (boucles de 0 à 7, calculs modulo 8).

1. Écrire les deux algorithmes Init et GetBits pour des mots de 3 bits.
2. Calculer les 6 premiers mots de 3 bits générés à partir de la clé 1,4,2,9.
3. Chiffrer le message `OAI` avec la clé 1,4,2,9.
4. Déchiffrer le message PPx avec la clé 4,2.

Bonus Qui a écrit ce texte ?

```
KPTS1tVnJuHZMyrbclQHiD5IiJZrbxGw.9dBk8A9vZoezf A p3ZtlRlH.uemwi2VTQDwL4lv8SO9KT9
iFk9.8VoBn  sMhnFYX7JDavi19cVSKKTZFc7Xy4zXIUV.uznNcQQVwRw3RANC.b.3ruNAlkCQEF95Wj
kCHc5EpuBgTyjNZHZNRIWz2fymKKAnkQYL4n7E7g9UuicsQdmoIX.ixh2QYMCHzp5LVr0W7Gm41Yo8KA
ePUr5lhAiOSmjOtCb3R9fvi9p4q1lQtYcSw hOOF6eGfwQyiS9hrbpNjp6E3jrbln7jKZnuLsIHBMK2X
tuDMDkVCR.S8V2UNiydhqzd731T71taDcyVrJ0upr5jU51CmDPiRFAqho6IVaXiplCmROfIuvjhap77v
wa1lWw0LtjT3n766MDeIGkc SeoTciojgWA5VuD.km7WX5qK rf9Lv3DqgMcGQ rj5QZWqmEGcn0rw4I
Q6kez8a1VxYL95BUPtCVa6Xqj2QwhDNUD7P6r3Mno49eHgSd6v iSAozCO1ZQ.fA..JAOFtte8AzW6sf
4vMfYrl5vZPoLgQvSsV 1D0t3MSs9U1KONp0Zbu2VEzv t3ZWNm5TbpoVAdFP3bP Nh6uy74oxp 7
```

Annexes

Un codage des caractères sur 6 bits (ainsi le caractère U est codé, en octal, 24) :

	0	1	2	3	4	5	6	7
0	A	B	C	D	E	F	G	H
1	I	J	K	L	M	N	O	P
2	Q	R	S	T	U	V	W	X
3	Y	Z	a	b	c	d	e	f
4	g	h	i	j	k	l	m	n
5	o	p	q	r	s	t	u	v
6	w	x	y	z	0	1	2	3
7	4	5	6	7	8	9	.	