Cryptanalyse fréquentielle

- Tableau de la fréquence des digrammes en anglais

First Letter	Second Letter																											
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Space	*Total*
A	2	144	308	382	1	67	138	9	322	7	146	664	177	1576	1	100	-	802	683	785	87	233	57	14	319	12	50	7086
B	136	14	-	-	415	-	-	-	78	18	-	98	1	-	240	-	-	88	15	7	256	1	1	-	13	-	36	1417
C	368	-	13	-	285	-	-	412	67	-	178	108	-	1	298	-	1	71	7	154	34	-	-	-	9	-	47	2053
D	106	1	-	37	375	3	19	-	148	1	-	22	1	2	137	-	-	83	95	3	52	5	2	-	51	-	2527	3770
E	670	8	181	767	470	103	46	15	127	1	35	332	187	799	44	90	9	1314	630	316	8	172	106	87	189	2	4904	11612
F	145	-	-	-	154	86	-	-	205	-	-	69	3	-	429	-	-	188	4	102	62	-	-	-	4	-	110	1561
G	94	1	-	-	289	-	19	288	96	-	-	55	1	31	135	-	-	98	42	6	57	-	1	-	2	-	686	1901
H	1164	-	-	-	3155	-	-	1	824	-	-	5	1	-	487	2	-	91	8	165	75	-	8	-	32	-	715	6733
I	23	7	304	260	189	56	233	-	1	-	86	324	255	1110	88	42	2	272	484	558	5	165	-	15	-	18	4	4501
J	2	-	-	-	31	-	-	-	9	-	-	-	-	-	41	-	-	-	-	-	56	-	-	-	-	-	-	139
K	2	-	-	-	337	-	-	-	127	-	-	10	1	82	3	1	-	-	50	-	3	-	-	-	8	-	309	933
L	332	4	6	289	591	59	7	-	390	-	38	546	30	1	344	34	-	11	121	74	81	17	19	-	276	-	630	3900
M	394	50	-	-	530	6	-	-	165	-	-	4	28	4	289	77	-	-	53	2	85	-	-	-	19	-	454	2160
N	100	2	98	1213	512	5	771	5	135	8	63	80	-	54	349	-	3	2	148	378	49	3	2	2	115	-	1152	5249
O	65	67	61	119	34	80	9	1	88	3	123	210	417	598	336	138	-	812	195	415	1115	136	398	2	47	5	294	5776
P	142	-	1	-	280	1	-	24	97	-	-	169	-	-	149	64	-	110	48	40	68	-	3	-	14	-	127	1337
Q	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	66	-	-	-	-	-	-	66
R	289	10	22	133	1139	13	59	21	309	-	53	71	65	106	504	9	-	69	318	190	89	22	5	-	145	-	1483	5124
S	196	9	47	-	626	-	1	328	214	-	57	48	31	16	213	107	8	-	168	754	175	-	32	-	34	-	2228	5292
T	259	2	31	1	583	1	2	3774	252	-	-	75	1	2	331	-	-	187	209	154	132	-	84	-	121	1	2343	8545
U	45	53	114	48	71	10	148	-	65	-	-	247	87	278	3	49	1	402	299	492	-	-	-	1	7	3	255	2678
V	27	-	-	-	683	-	-	-	109	-	-	-	-	-	33	-	-	-	-	-	1	-	-	-	11	-	-	864
W	595	3	-	6	285	-	-	472	374	-	1	12	-	103	264	-	-	35	21	4	2	-	-	-	-	-	326	2503
X	17	-	9	-	9	-	-	-	10	-	-	-	-	-	1	22	-	-	-	23	8	-	-	-	-	-	21	120
Y	11	10	-	-	152	-	1	1	32	-	-	7	1	-	339	16	-	-	81	2	1	-	2	-	-	-	1171	1827
Z	3	-	-	-	26	-	-	-	2	-	-	4	-	-	2	-	-	-	3	-	-	-	-	-	3	9	2	54
Space	1882	1033	864	515	423	1059	453	1388	237	93	152	717	826	478	721	588	42	494	1596	3912	134	116	1787	-	436	2	-	19998
Total	7069	1418	2059	3770	11645	1549	1906	6739	4483	31	932	3885	2163	5241	5781	1339	66	5129	5278	8536	2701	870	2507	121	1855	52	19974	1E+05

  - On remarque que certains digrammes sont plus fréquents que d'autres (ex. : HE avec 3155)
  - On peut raisonner sur les digrammes de la même façon que sur les lettres

Cryptanalyse fréquentielle

- Le fait que la source soit non-markovienne facilite le raisonnement sur les digrammes
  - Fréquence du digramme « Q U » en français : 1.6% (top 10 !)
  - Fréquence du digramme « Q U » source markovienne : 0.066%
  - L'entropie d'une source non-markovienne est moins élevée
  - (Les combinaisons « impossibles » sont plus instructives que les valeurs de fréquences des combinaisons possibles)
- On peut reprendre l'exercice avec des blocs de 3 lettres (trigrammes)
- Si on dispose des tables des fréquence, on peut considérer les *n*-grammes avec *n* qui tend vers l'infini
  - Encore plus de structures ! (mots, phrases, sens du texte, etc.)
  - Encore moins d'entropie

cryptanalyse fréquentielle

- Il y a une limite théorique à la capacité de faire l'analyse fréquentielle en utilisant des blocs
  - Lorsque la taille de *b* (taille du bloc) tends vers l'infini, on obtient l'entropie du langage
  - Pour une source non-markovienne hautement structurée comme le langage humain, l'entropie par bloc diminue de façon asymptomatique jusqu'à la limite
    - Plus de structure implique plus de différences entre les fréquences d'occurrence
    - Indique plus de compressibilité
    - Doit traiter une table de « symboles » de plus en plus grande !
  - Le tableau des digrammes est déjà difficile à interpréter (la fréquence maximale en anglais est de 3155/100 000)
    - Difficile de faire la distinction entre divers digrammes pratiquement équiprobables
    - Plus sensibles aux aléas de la statistique

Cryptanalyse fréquentielle

- En pratique, pour compléter l'analyse, il est beaucoup plus aisé de vous fier à votre propre connaissance du langage, de la sémantique, du sujet du texte, etc.

Contremesures et principes de base

- Maximisation de l'entropie sur symboles de codage
  - On ne peut pas
    - Changer la source (p.ex. pour réduire son entropie)
    - Choisir l'alphabet de codage (p.ex. déterminer par le chiffrement)
  - On peut
    - Ajuster le codage pour maximiser l'entropie
      - Faire du codage par bloc
      - Faire du bourrage (« padding ») avec des caractères aléatoires
      - Implémenter de la compression
    - Utiliser des algorithmes de chiffrement probabilistes
- Maximisation de l'entropie des clés choisies
  - Génération aléatoire
  - Contrôle sur la génération des clés ("souveraineté de clé")

Questions ?

POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE

INF 4420: Sécurité Informatique
Cryptographie II

Cuppens

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Aperçu – Crypto II

- Types de chiffrement
  - Par bloc vs. par flux
  - Symétrique vs. asymétrique
- Algorithmes symétriques modernes
  - DES
  - AES
- Fonctions de hachage
  - Propriétés
  - Obsolètes
  - Présentes et futures:
- Algorithmes à clé publique
  - Arithmétique modulaire
  - Notion de groupe

Type de chiffrement – Bloc vs. Flux

- Chiffrement par bloc
  - Algorithme où chaque mot de code (un « bloc ») est codée avec la même clé *k*
  - Pour la plupart des sources, $|\Sigma|$ = M est petit, en conséquent $|\mathrm{T}|$ = N >> M, de façon à éviter force brute
  - Codage
    - Doit « regrouper » symboles de Σ en blocs dans T
    - Problème de « latence »

- Chiffrement par flux
  - Chaque mot de code est chiffré avec une clé différente
  - Les clés sont générés au « fur et à mesure »

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Type de chiffrement – Symétrique vs. Asymétrique

- Symétrique
  - Clé de déchiffrement = clé de chiffrement
  - La clé doit toujours être gardé secrète !!

- Asymétrique
  - Deux clés différentes
  - En général, il n'est pas possible de déduire clé de déchiffrement en connaissant clé de chiffrement, donc
    - Clé de chiffrement = Clé « publique »
    - Clé de déchiffrement = Clé « privée »
  - Facilite la gestion de clés
  - Permet plusieurs autres applications au-delà du chiffrement

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

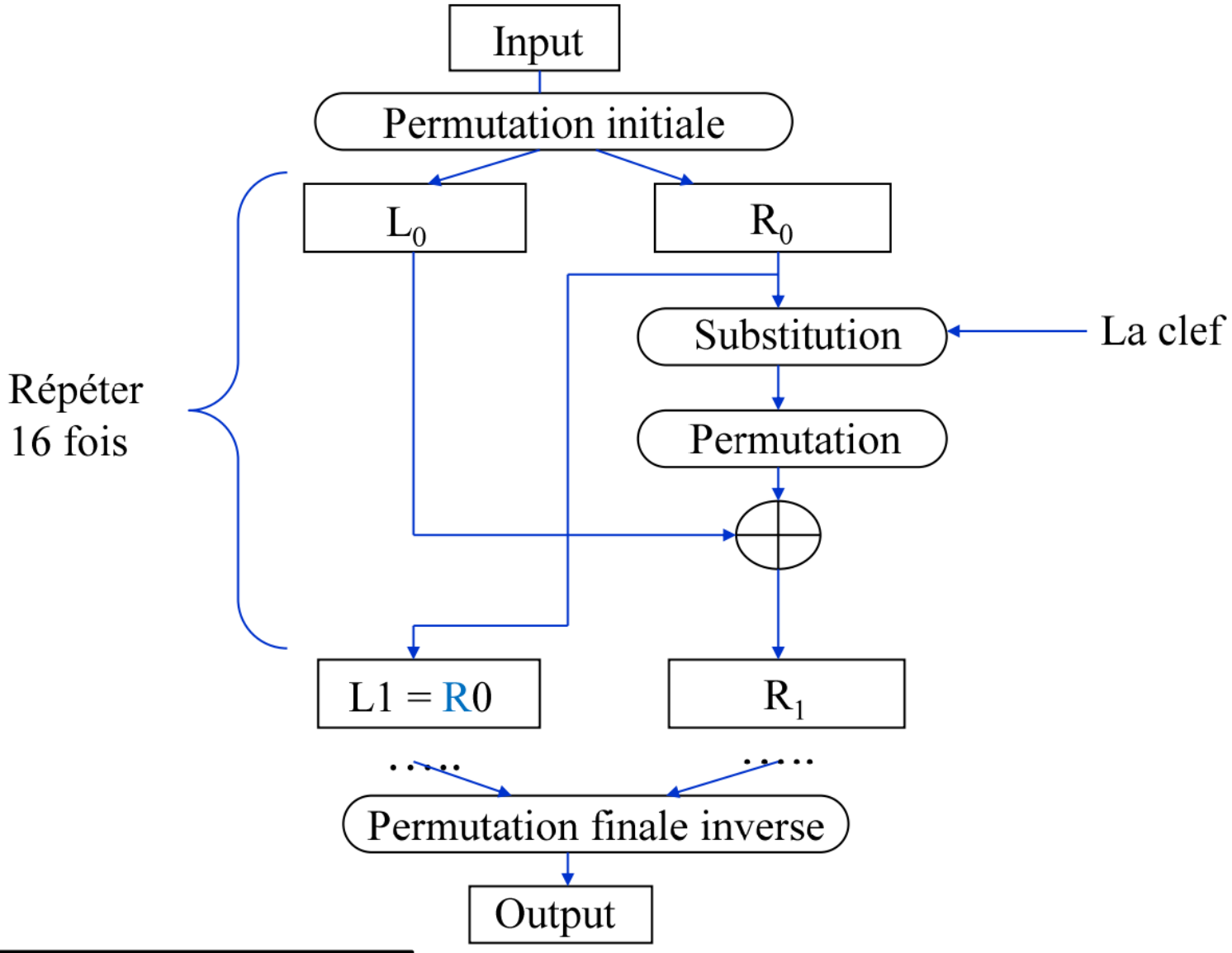
DES (Data Encryption Standard) – Historique

- Développée par le gouvernement américain
  - pour usage général par le public (intérêts privés commerciaux américains)
- Devis et spécifications en 1970
  - établie par le National Institute of Standards and Technology (NIST)
  - complètement spécifié et facile à comprendre
  - sécurité indépendante de l'algorithme lui-même (Principe de Kerkchoff)
  - disponible à tous et adaptable à diverses applications (usage « commercial »)
  - possibilité d'implantation économique en matériel et en logiciel
  - efficace d'utilisation, validable, et exportable
- Deuxième appel de propositions en 1974
  - Choix de l'algorithme "Lucifer" développé par IBM
  - Adopté comme « DES » le 23 novembre 1976

DES
Survol

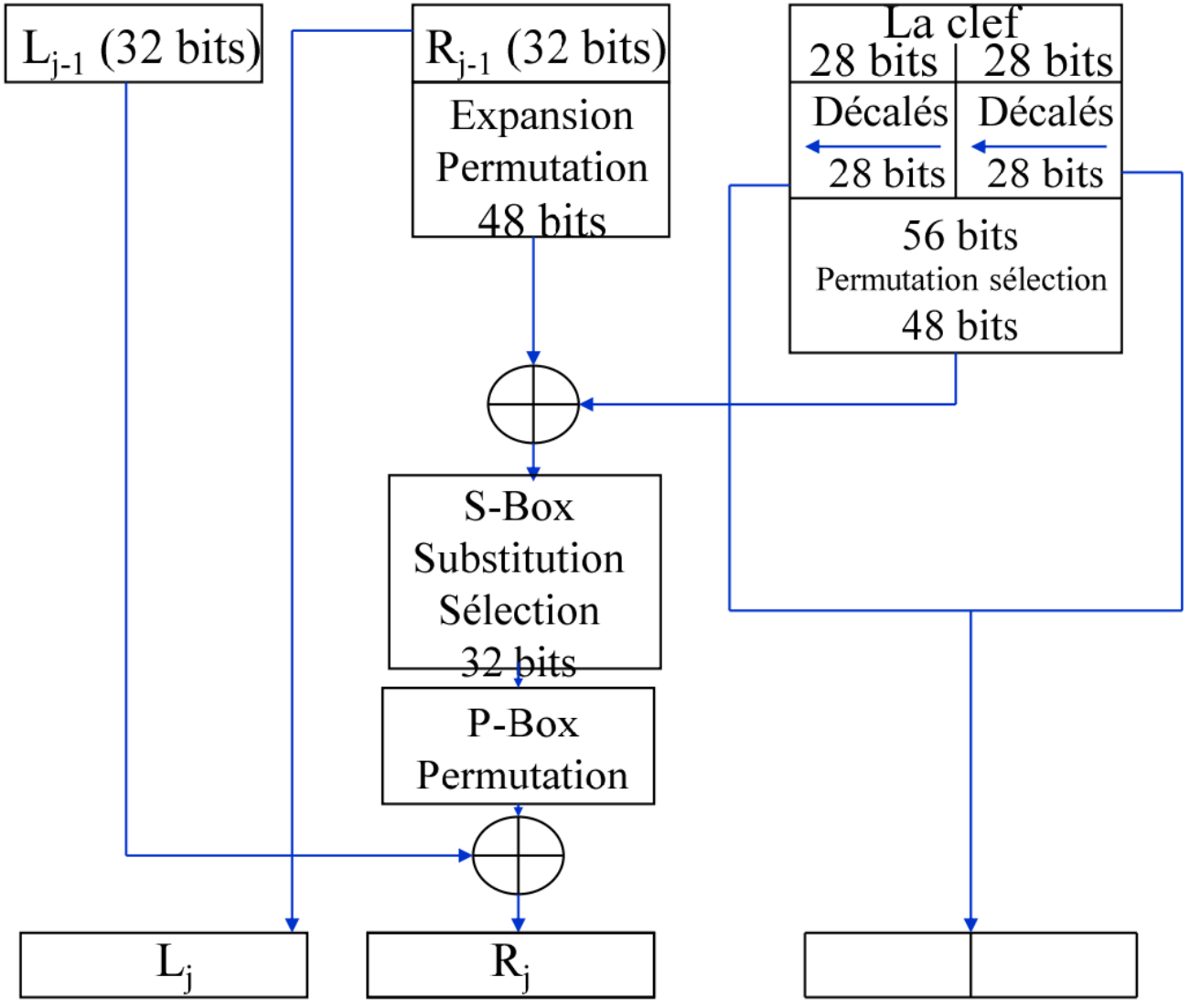
- Application répétée (16 cycles) de
  - Substitution
    - changer systématiquement certains patrons de bits pour d'autres
  - Permutation
    - réarranger l'ordre des bits
- Arithmétique à 64 bits seulement; clef de 64 bits
- Chiffrement par blocs de 64 bits
- Objectifs de sécurité
  - Confusion
    - les bits d'output n'ont aucune relation évidente avec l'input
  - Diffusion
    - répartir les changements sur l'ensemble des bits du message
    - changement au bit *i* du message implique changement dans plusieurs bits du cryptogramme.

Norme DES – Détails



POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Norme DES – Détails d'un cycle



Norme DES – Déchiffrement

- Chaque cycle de déchiffrement dérive du cycle précédent
  - $L_j = R_{j-1}$; $R_j = L_{j-1} \oplus f(R_{j-1}, k_j)$
- Dans l'autre direction
  - $R_{j-1} = L_j$; $L_{j-1} = R_j \oplus f(R_{j-1}, k_j)$
  - si on substitue: $L_{j-1} = R_j \oplus f(L_j, k_j)$
- La procédure est donc réversible
  - la même fonction *f* est utilisée pour le déchiffrement
  - il suffit de prendre les 16 sous clefs dans l'ordre inverse

S-box

- Quelle S-box est utilisée dépend de la clé de chiffrement
- Voici un exemple d'une S-box de DES (S5)

S5	Middle 4 bits of input						
	.0000.	.0001.	…	.1100.	**.1101.**	.1110.	.1111.
0....0	0010	1100	…	1101	0000	1110	1001
0....1	1110	1011	…	0011	**1001**	1000	0110
1....0	0100	0010	…	0110	0011	0000	1110
1....1	1011	1000	…	1010	0100	0101	0011

– Exemple 6-bit string 0 1101 1 est converti en 4-bit string 1001

Sécurité de la norme DES

- Questions relatives à la conception de l'algorithme
  - caractère confidentiel de la conception
  - présence de « trappes » (choix des s-box) ?
  - possibilité d'une faiblesse fondamentale ?
- Le nombre d'itérations (16) est il suffisant
- La taille de la clef (56 bits) est elle suffisante ?
  - originalement, Lucifer prévoyait 128 bits
  - possibilité d'une attaque force brute réussie
  - possibilité d'une attaque de type parallèle
  - possibilité de réussite d'une attaque de texte clair choisi

- Toutes ces questions avaient des réponses satisfaisantes

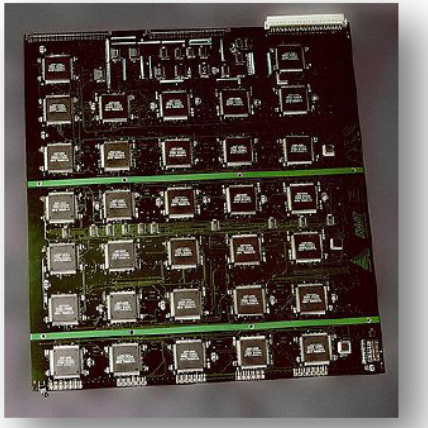
Variantes de DES

- Double DES
  - Choisir deux clefs k_1 et k_2
  - Chiffrer deux fois: $E(k_2, E(k_1, m))$
  - Est équivalent à un DES avec clé de 57 bits
    - 1 bit de clé supplémentaire…
    - … seulement deux fois plus de travail pour briser
- Triple DES (ou 3DES)
  - Deux clefs
  - Trois opérations: $E(k_1, D(k_2, E(k_1, m)))$
  - Équivalent à doubler la taille effective de la clé – 112 bits
  - Très robuste et effectif contre toutes les attaques faisables connues

Attaque par force brute – Limites ultimes

- Entropie
  - de la source de clé ➔ nombres d'essais de clé
  - du texte clair (« plaintext ») ➔ facilité de reconnaissance de la bonne clé
- Puissance de l'attaquant
  - Vitesse d'essai du matériel disponible
  - Nombre de machines disponibles (= budget total / cout par machine)
- Patience de l'attaquant
  - Combien de temps a-t-il ?
    (après combien de temps l'information n'est plus utile ?)
- Durée de vie de l'information
  - Après combien de temps l'information n'est plus confidentielle ?
- Loi de Moore
  - « À chaque 18 mois la puissance de calcul disponible pour un même budget double »
    ➔ Pour une sécurité équivalente on doit ajouter un bit de clé à chaque 18 mois…

« DES is dead… »



- Attaque par force brute
  - 56 bits de clé ne sont pas suffisants aujourd'hui
  - COPACABANA
    - Cost-Optimized Parallel COde Breaker
    - Matériel spécialisé
    - Peut retrouver la clé en moins d'une journée !
    - Puissance
      - Nombre de secondes dans une journée = 60*60*24 = 86 400 ~ 2^{18} s
      - Nombre d'essais = 2^{56} clés
      - Vitesse d'essai = 2^{56} / 2^{18} = 2^{56-18} = **$\underline{2^{38}\ \text{clé/s}}$** (~ 256 Giga-clé/s) (gardez ce chiffre en tête!!)
    - COPACABANA est facilement reconfigurable pour d'autres algorithmes…
- DES est obsolète, mais 3DES survi
  - demeure une norme acceptée quand même (pour le moment…)