

CRYPTOGRAPHIE I – MODÈLE DE SHANNON RÉVISÉ

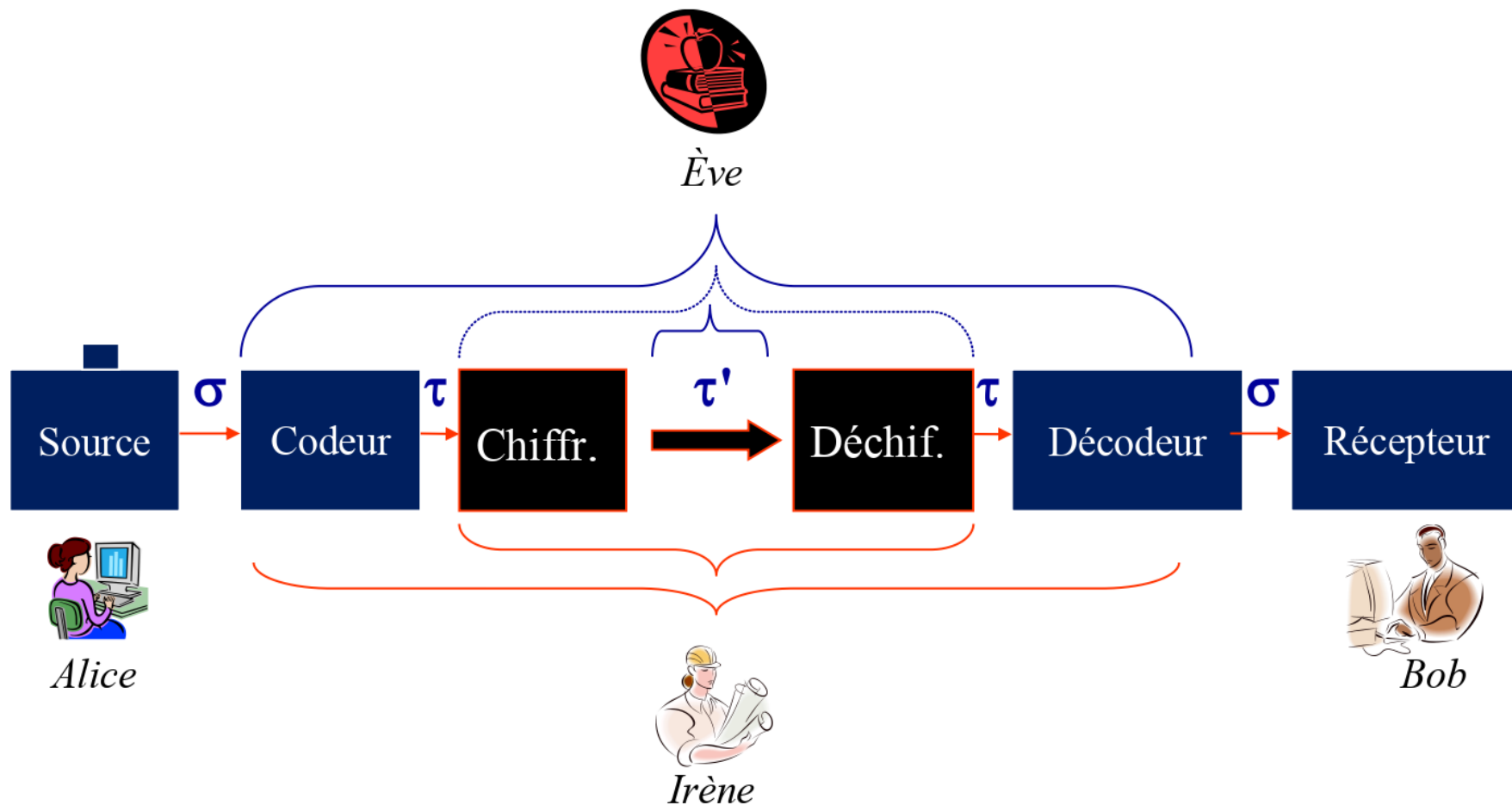


**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE



Modèle de Shannon révisé





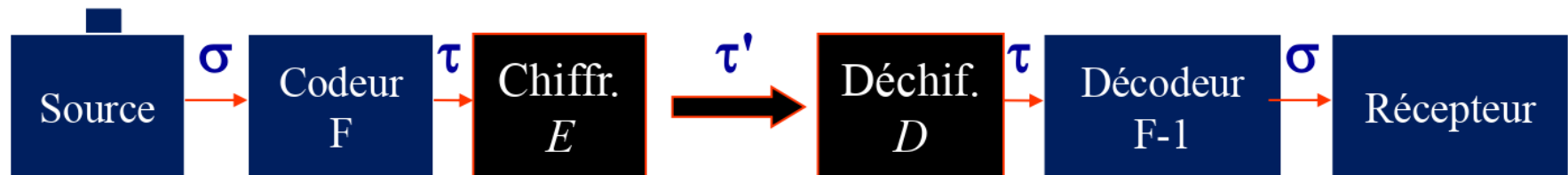
Modèle de Shannon révisé

- Ève
 - Peut intercepter impunément tous les mots de codes τ transmis sur le canal
- Irène
 - Choisit l'algorithme de chiffrement
 - Détermine la politique de choix et gestion de clés
 - Doit tenir en compte le codage
 - en considérant les caractéristiques de la source (DP, entropie, etc.)
 - en influençant le choix de codage (si possible)
 - en choisissant et adaptant l'algorithme de chiffrement en conséquence (choix de taille de clés, compression/décompression, etc.)
 - Pourquoi : voir TP 1...



Algorithme de chiffrement – Concepts généraux

- Alphabet
 - Entrée : T
 - Sortie : en général T , mais peut-être un autre alphabet T'
- Fonction de chiffrement
 - Clé de chiffrement = k_e
 - $\tau' = E(k_e, \tau) = E_{k_e}(\tau)$
- Fonction de déchiffrement
 - Clé de déchiffrement = k_d
 - $\tau = D(k_d, \tau') = D_{k_d}(\tau')$



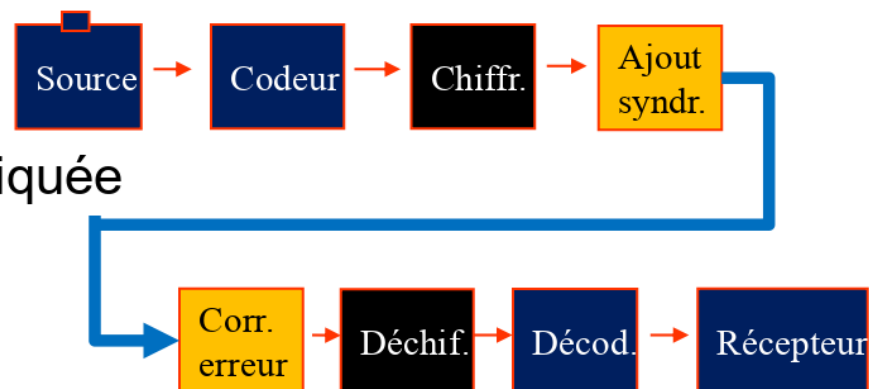


Cryptographie et correction d'erreurs

- Ne corrige pas les erreurs

- Il faut donc que

- $\tau = \tau'$ (pas de bruit), ou que
 - la correction d'erreur soit appliquée
 - après le chiffrement et
 - avant le déchiffrement



- La correction d'erreur

- Constitue une forme de protection de l'intégrité des messages

- Protège contre des erreurs aléatoires (menace accidentelle)
 - P.ex. erreur de transmission due au bruit, interférence accidentelle, etc.
 - Ne protège pas contre la menace délibérée
 - P.ex. des erreurs introduites de façon « intelligente » par un acteur malveillant ayant accès au canal (Ève !)

CRYPTOGRAPHIE I – CRYPTOGRAPHIE CLASSIQUE



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE



Algorithmes "classiques" mono-alphabétiques

- Algorithme de César
 - Source
 - texte en caractères latin
 - Codage
 - lettres \rightarrow chiffres de 1 à 26
(20 pour être historiquement exact)
 - Chiffrement
 - $x \rightarrow x+3 \bmod 26$
 - Clés
 - nil
- Algorithme de décalage
 - Source et codage
 - idem
 - Chiffrement
 - $x \rightarrow x + k \bmod 26$
 - Clés
 - $k \in \{1, \dots, 26\}$
- Algorithme de substitution
 - Source
 - Idem
 - Codage
 - aucun
 - Chiffrement
 - $x \rightarrow \pi(x)$
 - Clé
 - π (une table de substitution)
- Algorithme afin
 - Source et codage
 - lettres en chiffres
 - Chiffrement
 - $x \rightarrow a x + b \bmod 26$
 - Clé
 - (a, b) où $a, b \in \{1, \dots, 26\}$



Algorithmes classiques

- Algorithme de substitution

- On prend un texte en clair et, pour chacune des lettres du texte, on utilise la lettre comme index dans une table de substitution (π) pour trouver l'équivalent chiffré
- La table de substitution représente la clé
- « **H**ELLOWORLD » devient :

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
h	e	v	a	m	t	s	i	c	f	n	u	o	r	B	g	q	w	j	y	d	l	x	k	z	p

π

- « **i**muubxbwua »



Algorithmes classiques

- Algorithme de substitution
 - Avec la même clé et plus de texte

I	L	E	T	A	I	T	U	N	E	F	O	I	S	L	H	I	S	T	O	I	R	E	D	U	N	P	E	T	I	T	C	H	A	P	E	R	O	N	R	O	U	G	E
c	u	m	y	h	c	y	d	r	m	t	b	c	j	u	i	c	j	y	b	c	w	m	a	d	r	g	m	y	c	y	v	i	h	g	m	w	b	r	w	b	d	s	m

- On remarque qu'il est difficile de faire la correspondance entre le texte original et le texte chiffré sans connaître la table de substitution π (la clé)
- Sans le texte en clair, il serait aussi difficile d'inférer π à partir du texte chiffré et il est nécessaire d'obtenir un « grand » (au moins une occurrence de 25 lettres sur 26) nombre de texte en clair pour reconstruire la clé
- L'algorithme classique de substitution possède donc des propriétés raisonnables de confusion



Algorithmes classiques

- Algorithme de transposition (« bit shifting »)
 - On prend un texte en clair et on permute la position des lettres (ou des bits dans le cas moderne) entre elles en fonction d'une « clé »
 - Équivalent du jeu Charivari (allez voir sur Internet...)
 - Dans l'antiquité on utilisait un bâton autour duquel on enroulait une lanière de cuir (déguisée en ceinture) où était écrit le texte
- Exemple
 - Avec un « bâton » qui a une épaisseur de deux lettres, « HELLOWORLD » devient

h	l	o	o	l
e	l	w	r	d

- « hloolelwrld »




Algorithmes classiques

- Algorithme de transposition (« bit shifting »)

- Chiffrons ce texte avec un « bâton » de taille 6 et commençons au 3^e « trou de ceinture » (3^e caractère)

e	n	l	i	p	h	n	i
t	e	h	r	e	a	r	l
a	f	i	e	t	p	o	
i	o	s	d	i	e	u	
t	i	t	u	t	r	g	
u	s	o	n	c	o	e	



I L E T A I T U N E F O I S L H I S T O I R E D U N P E T I T C H A P E R O N R O U G E
e n l i p h n i t e h r e a r l a f i e t p o i o s d i e u t i t u t r g u s o n c o e

- Ici, il est « facile » d'inférer le texte original à partir du texte chiffré
- On peut « facilement » retrouver la clé à partir du texte chiffré
- La confusion est donc mauvaise
- Par contre, la disparition d'une lettre entraîne la modification de tout le texte chiffré qui suit
- La transposition amène donc une diffusion raisonnable



Algorithme de Vigenère

- Algorithme de Vigenère

- Source

- Texte en caractères latin

- Codage

- lettres \rightarrow chiffres de 1 à 26

- Clé

- $K = k_1 k_2 \dots k_m$, mot/phrase de longueur m

- Chiffrement

- $x_i \rightarrow (x_i + k_{i \bmod m}) \bmod 26$



La trahison des images, René Magritte 1929

C	E	C	I	N	E	S	T	P	A	U	N	E	P	I	P	E	...
S	E	X	Y	S	E	X	Y	S	E	X	Y	S	E	X	Y	S	...
3	5	3	9	14	5	19	20	16	1	21	14	5	16	9	16	5	...
+	19	5	24	25	5	24	25	19	5	24	25	19	5	24	25	19	...
22	10	1	8	7	10	17	19	9	6	19	13	24	21	7	15	24	...
V	J	A	H	G	J	Q	S	I	F	S	M	N	U	G	O	X	...



Masque jetable

- Connu sous le nom de « One-time Pad »
- Historique
 - Inventé par le capitaine Vernam (US Army Signal Corps) en 1919
 - Utilisée pour le Téléphone Rouge entre Moscou et Washington (guerre froide)
 - Utilisée par Che Guevara en Bolivie
- Fonctionnement
 - $\Sigma = T = \{0,1\}$
 - Algorithme : XOR bit-à-bit du message et de la clé
 - Clé
 - En « théorie »
 - chaîne de bits aléatoires, de longueur “infinie”
 - distribuée à l’avance (physiquement, etc.)
 - mauvaise diffusion et confusion, mais pourtant...
Seul algorithme avec « sécurité parfaite » (Shannon)
 - En « pratique »
 - chaîne de bits générée par un algorithme déterministe
Dépendant des messages/clés antérieurs
Générateur de nombres pseudo aléatoires (avec une « semence »)
 - au moins aussi longue que le message (pas de recyclage de clé)



Confusion et diffusion

- Même dans les algorithmes modernes, la confusion et la diffusion sont deux propriétés recherchées
- Au sens strict
 - Confusion : propriété de rendre la relation entre la clé de chiffrement et le texte chiffré la plus complexe possible
 - Diffusion : propriété où la redondance statistique dans un texte en clair est dissipée dans les statistiques du texte chiffré
 - On remarque que les algorithmes classiques ne respectent pas tout à fait cette propriété
- Objectif
 - Empêcher de retrouver la clé à partir de paires texte chiffré et texte déchiffré (exemple : attaque à texte choisi)
 - Rendre plus difficile l'analyse fréquentielle (on verra plus tard)

CRYPTOGRAPHIE I – CRYPTANALYSE DE BASE



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE



- Force brute
 - Essaie de toute les clés
 1. Déchiffrer le texte chiffré avec la clé à essayer
 2. Voir si le résultat est cohérent
 - Paramètre de difficulté
 - Taille de l'espace de clés
 N bits de clés = 2^N clés possibles
 - Génération de clés non-aléatoire/non-uniforme
Entropie de la source K générant les clés: $H(K) \leq N$
 - Critère de reconnaissance ou de succès
 - Comment savoir si on a la bonne clé?
 - Patron ou format reconnaissable
 - Le texte « fait du sens »
 - Le texte « marche », e.g. mot de passe, etc.
 - Paramètre de difficulté
 - Entropie de la source du message
Entropie basse → moins de messages “valides” → plus facile
Entropie élevée → plusieurs messages “valides” → difficile



- Analyse fréquentielle

- Méthode

1. Établir/retrouver fréquences des symboles de la source
2. Calculer les fréquences des symboles chiffrés obtenus
3. Comparer histogrammes de fréquences
4. Établir relations entre symboles chiffrés et symboles de sources
5. Essayer de déchiffrer le texte

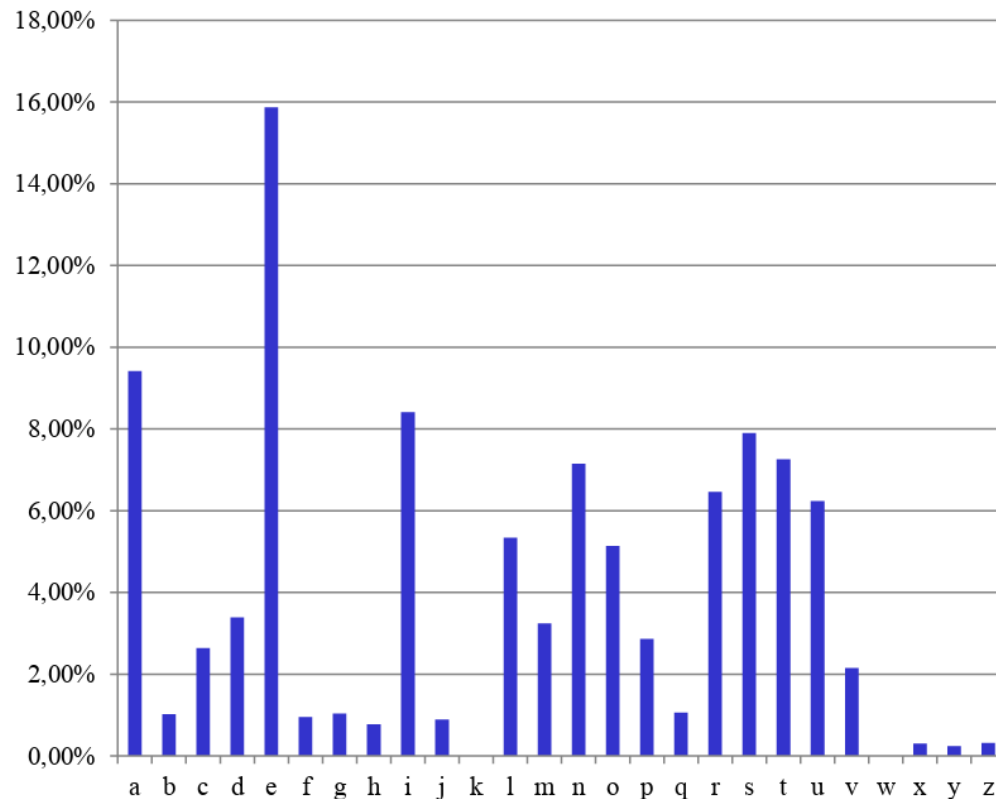
- Difficultés/précisions

- Codage connu → possible d'inverser le codage
- Paramètre de difficulté
 - Entropie de la source du message
 - » Entropie haute → histogramme « plat » → difficile
 - » Entropie basse → histogramme « escarpé » → plus facile
- Variante - Analyse par bloc
 - Si entropie trop haute pour S , alors on essaie avec S^2 , S^3 , ...
 - Compromis: taille de tableau de correspondance vs. entropie
 - Limite ultime = Entropie du langage

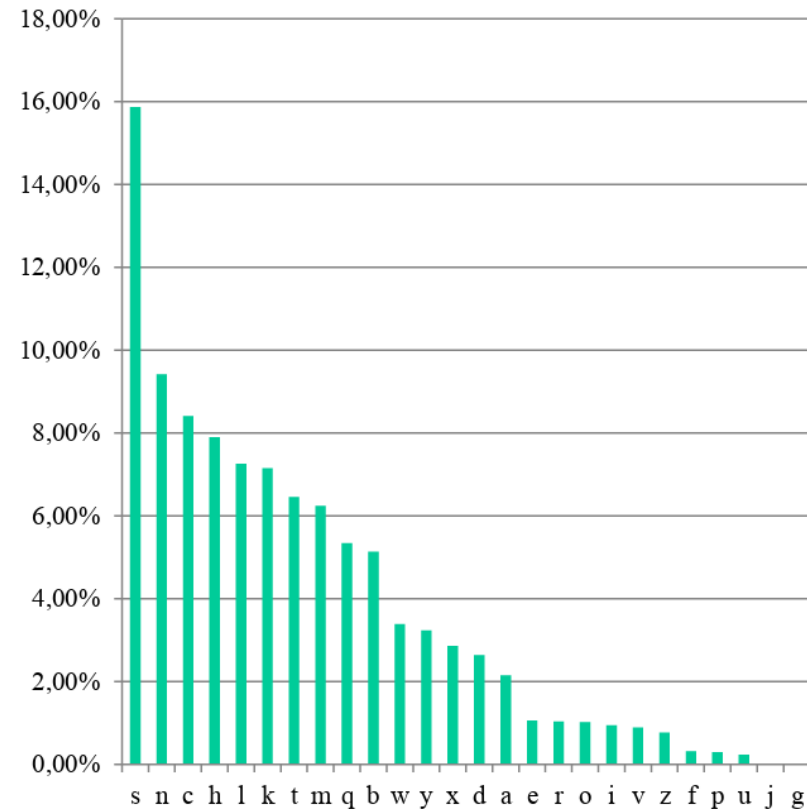


Cryptanalyse fréquentielle

- Histogramme de fréquence par lettre en français



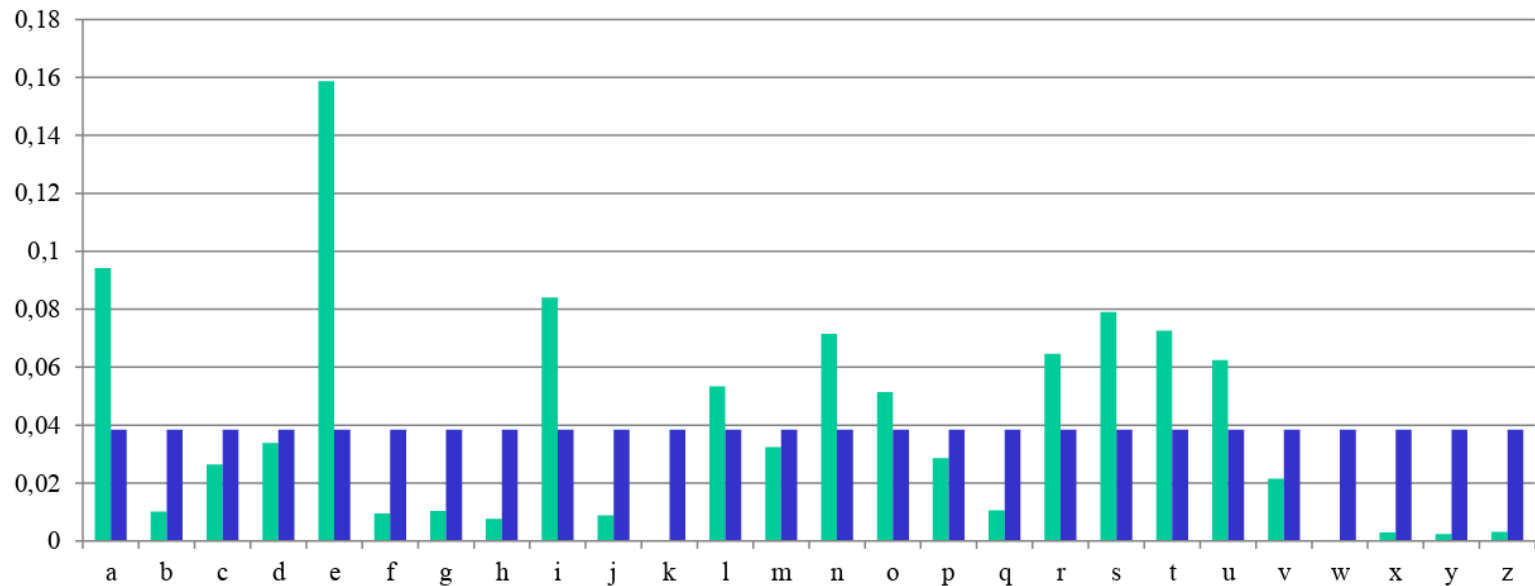
- Histogramme (ordonné) de d'un texte chiffré





Cryptanalyse fréquentielle


- Si l'entropie était maximale (tous les caractères équiprobables), nous allons obtenir l'histogramme suivant



- Il est difficile de tirer des conclusions sur le texte original à partir du texte chiffré



Cryptanalyse fréquentielle

- Une fois les caractères les plus probables démasqués, il devient difficile de déchiffrer les autres caractères
 - Il ne faut pas oublier que, pour un texte chiffré nous utilisons la PSEUDO-entropie, i.e. un estimateur statistique de l'entropie, on doit s'attendre à des déviations entre la valeur « observée » (proportion d'une lettre donnée) et la valeur « attendue » (fréquence d'usage dans le langage) 
 - Les variations seront encore plus grande si l'échantillon est peu statistiquement représentatif (taille, type de langage, etc.)

- Rappel : l'entropie par bloc est donnée par la formule suivante

$$\frac{H(S^b) / b}{\log_2 N}$$

- En prenant des blocs de caractères, on peut obtenir plus d'information