INF8085 : Sécurité Informatique
Cryptographie I

Frédéric Cuppens

Nora Cuppens & José Fernandez

Aperçu du module – Cryptographie

- Définitions et histoire
- Notions de base (théorie de l'information)
- Chiffrement
  - Méthodes « classiques »
  - Chiffrement symétrique
  - Chiffrement à clé publique
- Cryptanalyse de base
- Autres primitives cryptographiques
  - Hachage cryptographique
  - Signature numérique
  - Infrastructure à clé publique (ICP)
- Principes d'applications de la cryptographie
- Risques résiduels d'applications de la cryptographie

Cryptographie I (aujourd'hui)

- Définition et nomenclature
- Historique
- Théorie de l'information
  - Modèle de Shannon
    - Source d'information
    - Codage et compression
  - Entropie
- Chiffrement
  - Chiffrement et codage
  - Algorithmes « classiques »
- Cryptanalyse de base
  - Force brute
  - Reconnaissance de texte
  - Analyse de fréquences

CRYPTOGRAPHIE I – INTRODUCTION ET HISTOIRE

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Définitions et terminologie

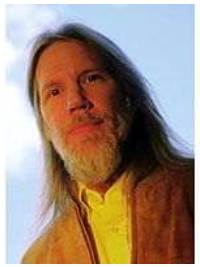
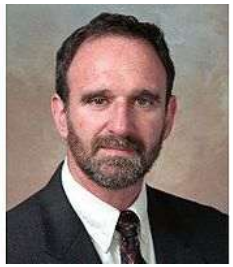
- Un peu de grec classique…
  - Kryptos = « caché », « secret »
  - Graphos = écriture
  - ⇒ Cryptographie
  - ⇒ Cryptanalyse
  - Logos = « savoir »
  - ⇒ Cryptologie
  - Stéganos = « couvert », « étanche »
  - ⇒ Stéganographie
- Un peu d'américain…
  - Alice
  - Bob
  - Ève
  - (Charlie)
  - Encrypt and Decrypt
- Un peu de français
  - Chiffrer et déchiffrer
  - Coder et décoder
  - Crypter et décrypter (!)
  - Irène !!! (l'ingénieure)
- Un peu de math…

Historique

- Les trois ères de la cryptographie
  - « Classique »
    - Jusqu'au masque jetable (chiffre de Vernam)
    - Chiffrement manuel ➔ chiffrement faible
  - « Moderne »
    - Crypto électro-mécanique et WWII (voir applet Enigma)
    - Guerre froide …
    - Crypto électronique et informatique – DES
    - Chiffrement par machines spécialisées ➔ chiffrement plus complexes
    - Réservés aux organisations pouvant acquérir l'équipement

Historique

- Les trois ères de la cryptographie (suite)
  - « Âge d'or »
    - Cryptographie à clé publique
    - 1976 - Whitfield Diffie & Martin Hellman
      - Introduise la notion de **cryptographie à clé publique**
      - Algorithme d'échange de clé (DH)
      - Introduise la notion de **signature numérique**

    - 1978 - Ronald Rivest, Adi Shamir, Leonard Adleman
      - Premier algorithme à clé publique (RSA)



    - 1973 – Clifford Cocks
      - Invente en parallèle un algorithme équivalent à RSA au sein du GCHQ
      - L'algorithme est classifié « TOP SECRET »
      - Existence dévoilée seulement en 1997

Historique

- Les trois ères de la cryptographie (suite)
  - « Âge d'or » (suite)
    - « Démocratisation » de la cryptographie
      - Années 80

        Cryptographie sur PC (PGP = Pretty Good Privacy))
      - Années 90 et 00

        Levée des restrictions d'exportations de cryptographie

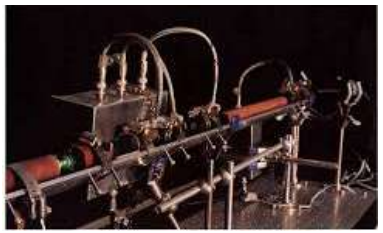
        Internet et Web
        - Protocoles réseaux sécurisés : SSH, SSL/TLS, IPSEC, etc.

        Infrastructures à clé publique et signature numérique
        - Transactions commerciales (bancaire et parabancaires)
        - Identité numérique
        - Cryptomonnaie
        - …

Historique

- Les trois ères de la cryptographie (suite)
  - Apocalypse « imminent » et ère post-quantique
    - 1984 – Charles Bennett et Gilles Brassard
      - Invention de la cryptographie quantique –
        - base sa sécurité sur les propriétés de la mécanique quantique

    - 1994 – Peter Shor (suivant les travaux de Dan Simon)
      - Découverte de la cryptanalyse quantique
        - Casse tous les algorithmes à clé publique connus
        - Nécessite d'un ordinateur quantique…



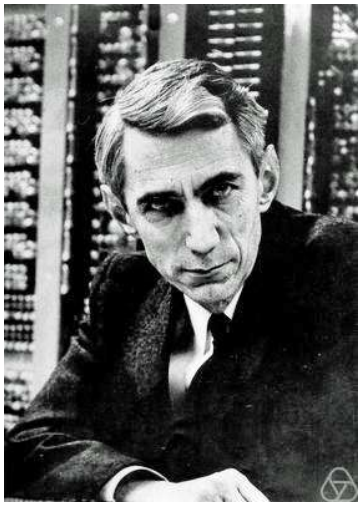
    - Années 10
      - Proposition d'algorithmes à clé publique « post quantiques »
        - Semblent résister à la cryptanalyse quantique
        - Peu pratiques à utiliser
        - Pas (encore) de standard établi
        - Adoption très lente…

CRYPTOGRAPHIE I – THÉORIE DE L'INFORMATION – MODÈLE DE SHANNON

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Claude Shannon



- IIe guerre mondiale
  - Contribue aux efforts de cryptanalyse de guerre
- Père de la Théorie de l'information
  - 1948 – « A Mathematical Theory of Information »
- Fondement théorique de la cryptographie
  - 1945 – « A Mathematical Theory of Cryptography »
    - Classifié – basée sur ses travaux de cryptanalyse
  - 1949 – « Communication Theory of Secrecy Systems »
    - Version non-classifiée, publié dans Bell Technical Journal

Claude Shannon

- Contributions
  - Introduit une définition mathématique de l'information
    - Source d'information
    - Modèle de Shannon – transmission d'information
  - Introduit la notion d'entropie (dans le contexte de l'information)
    - Définit le **bit** comme unité de mesure de l'information
    - Établit les limites fondamentales de la compression
    - Capacité maximale d'un canal de transmission (sans bruit)
    - Introduit une notion mathématique du bruit
      - Établit les limites fondamentales des codes correcteur d'erreurs
  - Introduit une théorie du « secret » en information
    - Modèle de Shannon révisé – transmission d'informations **secrètes**
      - Décrit le lien entre codage et chiffrement

Théorie de l'information

- « Information »
  - Valeur instantanée d'une variable aléatoire qui est transmise vers un récepteur à travers un canal de communication
- Concepts importants
  - Variable aléatoire
  - Canal de communication – Transmission
    ou
  - Moyen de stockage
- Exemples
  - La couleur du ciel (variable aléatoire) transmise via les ondes lumineuses (canal de transmission) vers votre œil (récepteur)
  - Contenu d'un fichier (variable aléatoire) transmise via le réseau téléphonique (canal de transmission) vers votre collègue (récepteur)

Théorie de l'information

- L'information est un concept *abstrait*
  - la valeur de l'information dépend des attentes du récepteur
    - Couleur du ciel : est-ce vraiment une information ?
    - (théorie de la décision) Est-ce que la température du soleil est critique à ma décision d'investir dans une entreprise web ?

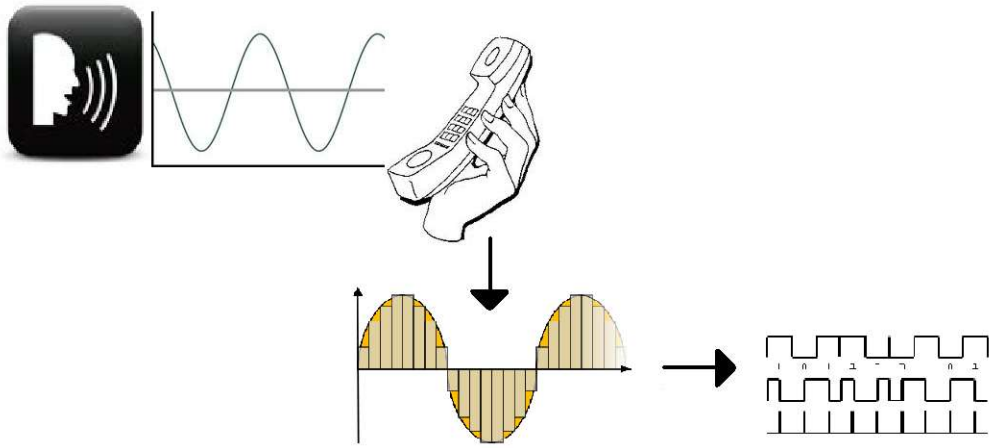
- Théorie de l'information (« Communication Theory »)
  - Ne s'intéresse pas à la sémantique de l'information (son « sens »)
  - S'intéresse à la quantité d'information qui manque au récepteur
    - Wheeler
      - *« information » in communication theory is not related to what you do say, but to what you could say*
  - Information = manque de connaissance du récepteur

Théorie de l'information

- Pour mesurer cette méconnaissance
  - Quantité d'information obtenue par observation directe de l'information obtenue/transmise
    - Représentée par la valeur de la variable aléatoire
    - Plus cette valeur est « aléatoire »
      - Plus grande est la méconnaissance du récepteur **avant** sa transmission
      - Plus sa transmission « ajoute » de l'information
    - Si cette valeur est peu aléatoire ou déterministe
      - Le récepteur a peu d'incertitude sur la valeur (méconnaissance faible)
      - La transmission n'apprend pas grand-chose au récepteur (valeur information de l'information faible)
  - Mesure mathématique d'information
    - Entropie de la variable aléatoire
  - Unité de mesure
    - Généralement le *bit*
      - Défini ainsi pour faciliter la représentation et calculs mathématiques

Théorie de l'information

- Source d'information
  - « Boîte noire »
  - Produit des symboles
    - selon un processus stochastique
    - seront codés (transformés)
    - seront stockés ou transmis
  - Variable aléatoire
    - associée au symbole produit
  - Série de symboles
    - différente à chaque fois (« réinitialisation »)
    - produite selon le même processus stochastique

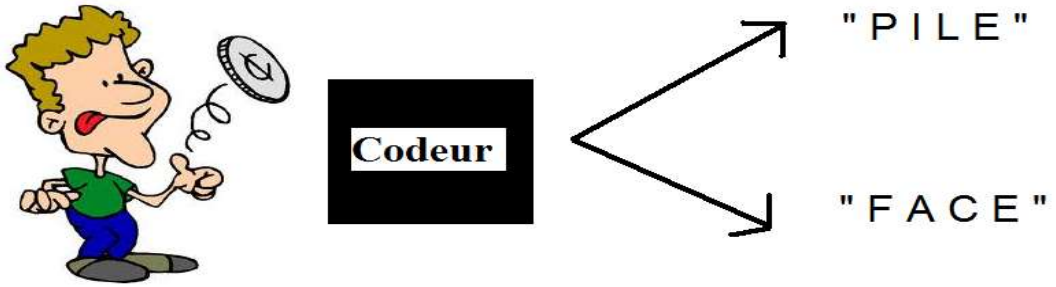


- Source discrète
  - un symbole à la fois
  - sur demande (« bouton »)

« a », « z », « θ », …

Théorie de l'information

- Transmission/stockage de l'information
  - La source produit de l'information « pure » sous forme abstraite
  - Ne peut pas être transmise ou stockée dans cet état « pur »
  - Doit avoir une <u>représentation physique</u> pour être transportée/stockée
- Codage
  - Processus de transformation de l'information
  - S'adapte au canal de transmission ou moyen de stockage
  - Permet au récepteur de reconvertir (décoder) l'information dans une forme intelligible (même forme qu'à la source)



  - <u>Codage ≠ chiffrement</u>
    - Le codage ne protège pas la confidentialité de l'information

Théorie de l'information

- Compression et codage
  - Le codage peut permettre de faire de la compression
    - Moins de symboles utilisés dans la transmission/stockage que par la source
  - 1er théorème de Shannon (voir plus loin)
    - Établit limite de la compression sans perte d'information *(lossless compression)*
      - Codes de Huffman
      - Lempel-Ziv-Welch (LZW)
    - Ne s'applique pas à la compression avec perte (lossy compression)
      - MP3
      - JPEG
      - MPEG

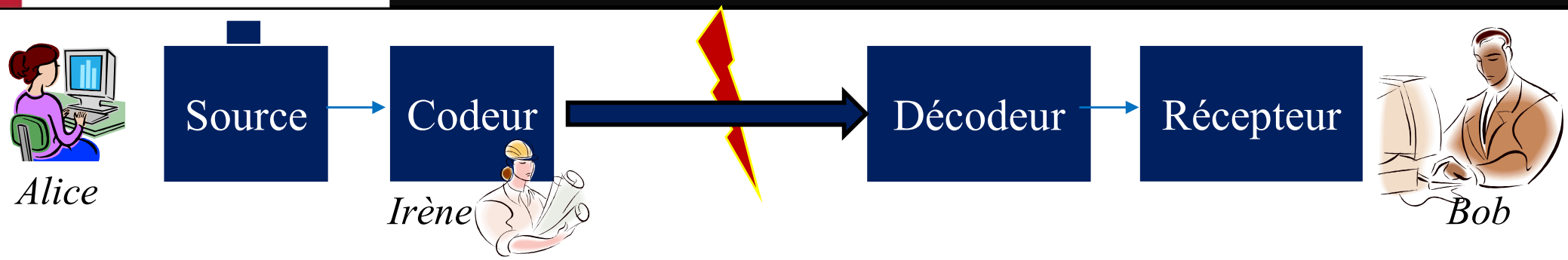
Théorie de l'information

- Les composants que nous avons évoqués sont du côté de la source
  - On fait l'image miroir pour avoir les composants du côté du récepteur
  - Source – codeur $\Rightarrow$ décodeur – récepteur

- On peut alors créer un modèle mathématique plus formel

  ➔ le modèle de Shannon

Modèle de Shannon



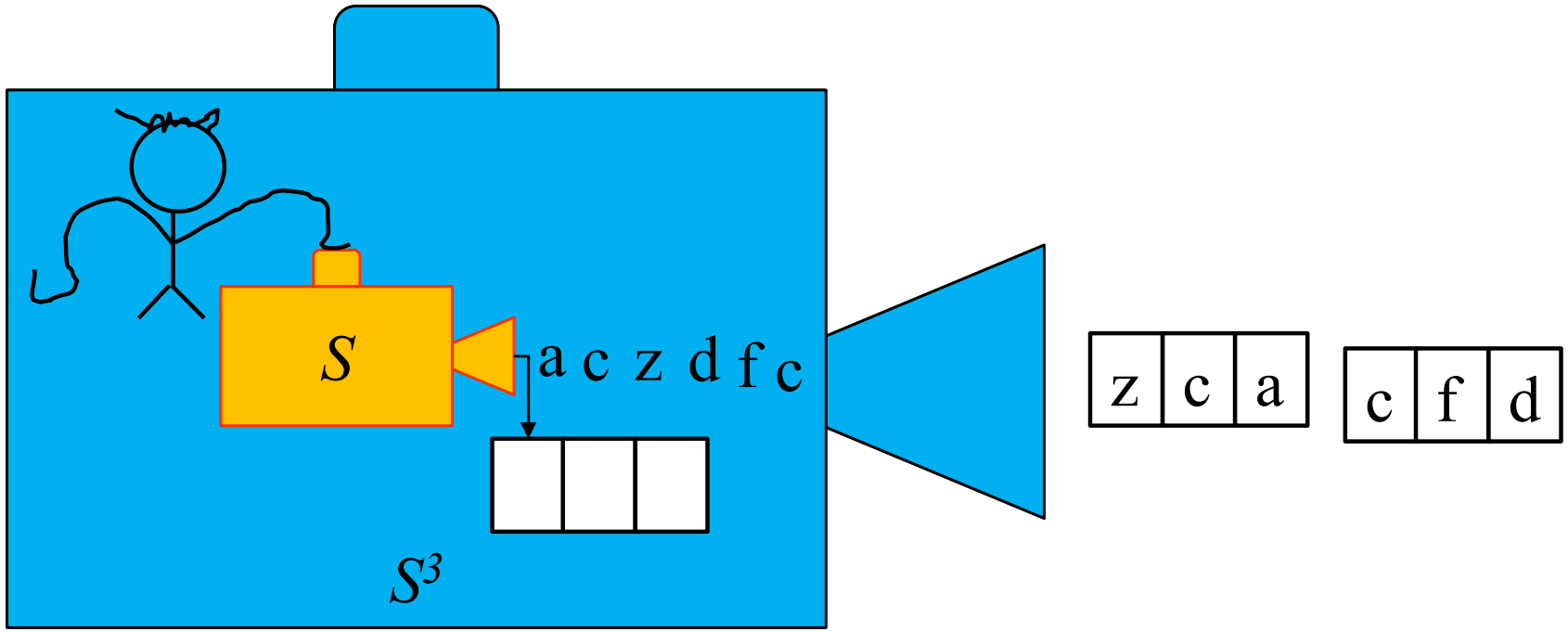
- Source
  - Produit des symboles d'un "alphabet" (∑)
  - Fonctionne "sur demande" (d'où le "bouton")
- Codage
  - Regroupe et transforme les symboles de la source dans un format pouvant être transmis ou sauvegardé
- Canal
  - Peut introduire du bruit
    - symbole reçu ≠ symbole transmis
- Décodage
  - Permet de reconstruire le message original
    - séquence des symboles de source

Source d'informations

- Alphabet
  - Ensemble discret fini $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$
  - Par convention taille de Σ, $|\Sigma| = M$
- Contrôle
  - Un "bouton" qui permet d'obtenir un symbole à la fois
- Principe de la boîte noire
  - Autre que le bouton et un nombre petit d'observations (symboles), on ne peut rien savoir sur le contenu ou fonctionnement de la source (sauf peut-être Alice, mais pas Ève, Irène ou Bob)
- Pourquoi cette abstraction ??
  - Permet de discuter de l'efficacité du codage (théorie de l'information)
  - Permet d'analyser correctement la résistance à certaines menaces
    - Algorithmes de chiffrement
    - Choix de mots de passe et phrases de passe
    - …

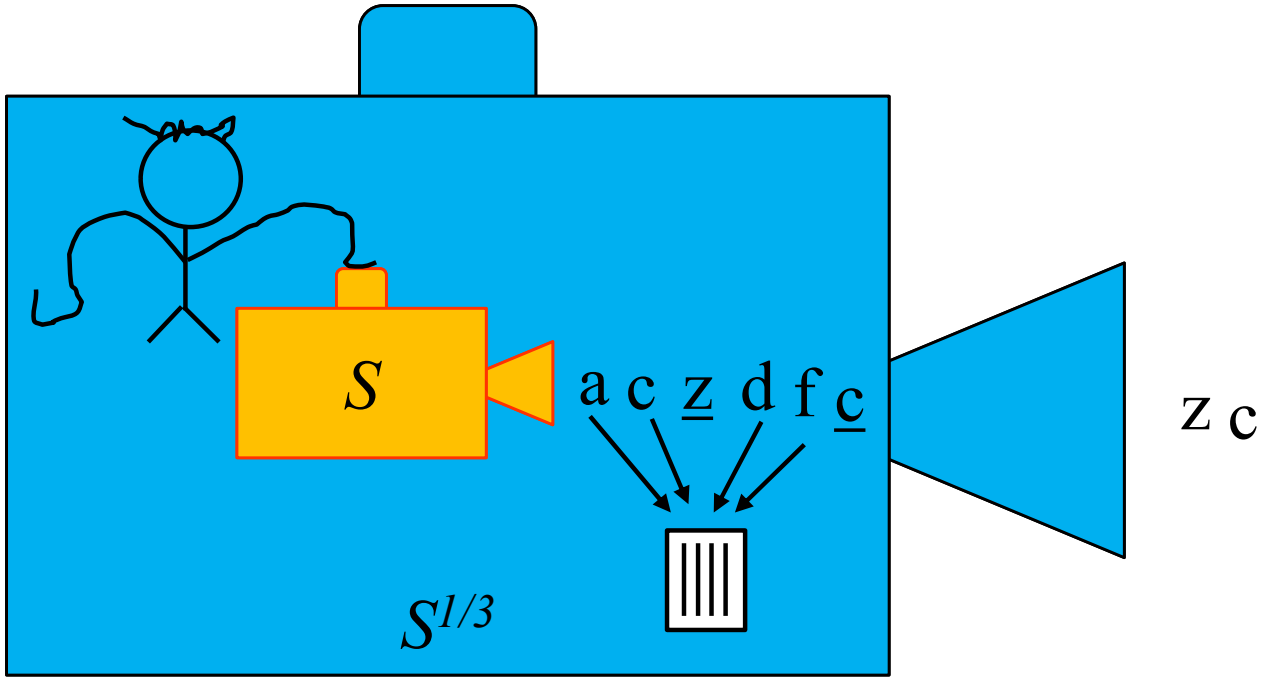
Sources dérivées

- Source par bloc
  - Étant donné une source S, et un entier positif b
  - S^b représente la source obtenue en encapsulant S par une boîte
    - qui mets b symboles de S dans un tampon (« buffer ») avant de les sortir
  - Noter que l'alphabet de S^b est maintenant Σ^b



Sources dérivées

- Source par échantillonnage
  - Étant donné une source *S*, et un entier positif *b*,
  - $S^{1/b}$ représente la source obtenue en encapsulant *S* par une boîte
    - qui émet seulement le 1er symbole de chaque *b* symboles sortie de *S*
  - L'alphabet de $S^{1/b}$ est le même que *S*, soit Σ



Types de source d'information

- Déterministe
  - La boîte « connaît » à l'avance toute la séquence de symboles (potentiellement infinie…)
- Probabiliste
  - La boîte choisit les symboles au fur et à mesure selon une distribution de probabilité
    - **Processus markovien ou "sans mémoire"**
      - p_i = Prob (S => " σ_i"), $\forall$ 1< i < M
      - e.g. Prob (S^b => " σ_i , σ_j") = $p_i\, p_j$
    - **Processus non-markovien**
      - Les probabilités de symboles peuvent dépendre des symboles antérieurs sortis de la source…

Codage

- Translittération
  - Un codage traduit les symboles de source vers un autre « alphabet » $\mathrm{T} = \{ \tau_1, \ldots, \tau_N \}$, *(Tau majuscule)*
- Fonction de codage
  - $F : \Sigma \rightarrow \mathrm{T}$,
    - $\tau = F(\sigma)$, représente comment le symbole σ devra être transmis
- Fonction de décodage
  - $F^{-1}: \mathrm{T} \rightarrow \Sigma$
    - $\sigma' = F^{-1}(\tau')$,
      - Si $\tau' \neq \tau$ alors $\sigma' \neq \sigma$

        il y a eu erreur de transmission (bruit dans le canal)
      - Si $\tau' = \tau$ alors $\sigma' = \sigma$

        transmission sans erreur

        Bob reçoit ce que Alice (source) a émis

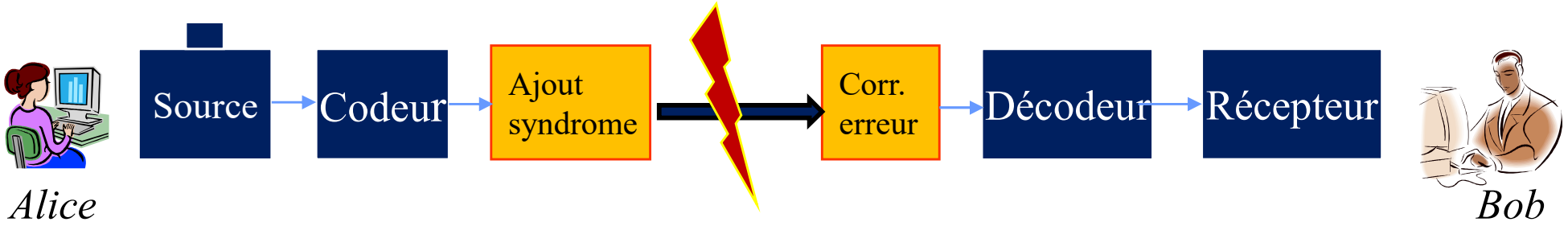
      ➔ F est nécessairement une injection

Correction d'erreur

- Code correcteur d'erreur
  - L'introduction de bruit dans le canal est compensé en utilisant un code correcteur d'erreur dans le codage t.q.
    Prob ($F^{-1}(\tau') = \tau$) → 1, où τ' est le symbole reçu via le canal
- L'efficacité du code correcteur d'erreur
  - Dépend du niveau de bruit introduit par le canal
    → celui-ci peut être mesuré avec l'entropie de Shannon
  - Se mesure également en nombre de bits nécessaires par symbole de source, pour un code qui corrige « presque toutes les erreurs »
- 2e Théorème de Shannon
  - Établit le lien entre l'efficacité du code correcteur d'erreur et le niveau de bruit du canal

Correction d'erreur

- En pratique, la correction d'erreur est souvent Une étape distincte et séparée du codage
  - Chez Alice
    - Codage supplémentaire après codage initial
    - Ajout d'information supplémentaire (*syndrome*)
  - Chez Bob
    - Décodage initial avant décodage final
    - Analyse du syndrome et du message
      - permet de corriger les erreurs (avec haute probabilité)



CRYPTOGRAPHIE I – THÉORIE DE L'INFORMATION – ENTROPIE

POLYTECHNIQUE MONTRÉAL

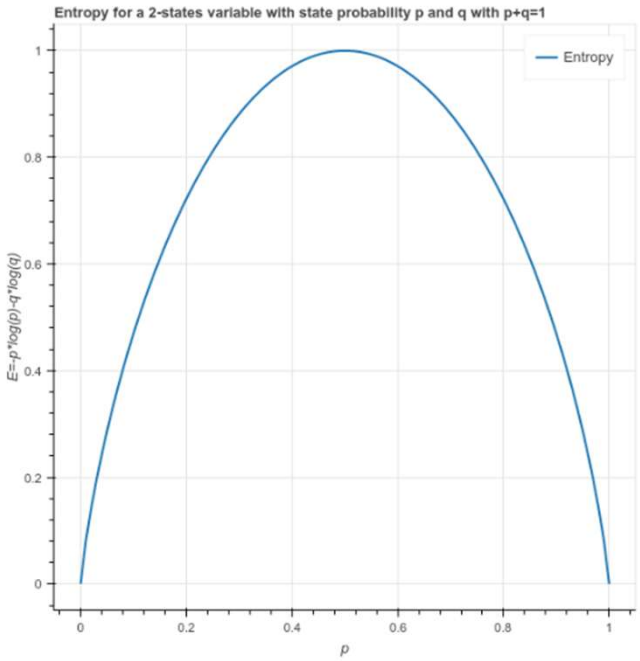
UNIVERSITÉ D'INGÉNIERIE

Efficacité du codage - Compression

- Compression
  - Dans certaines circonstances, on voudrait pouvoir coder en utilisant moins de bande passante, p.ex. tel que $N < M$
  - Efficacité du code
    - est mesurée en bits transmis par chaque symbole de source émis
  - 1er Théorème de Shannon
    - Efficacité maximum d'un code compresseur est approximativement égale à $H(S)$
    - Il existe un code compresseur (sans erreur) avec efficacité $H(S) + 1$
  - Qu'est-ce « $H(S)$ » → L'entropie de la source S

Entropie de Shannon

- Définitions
  - $H(S) = \Sigma_i \; p_i \log_2 1/p_i$



- Propriétés
  - Fonction convexe
  - $\Sigma = \{0,1\}$
    - Prob (S="0") = p, Prob (S="1") = q = 1-p
    - Valeur minimale
      - Prob (S="0") = 1; Prob (S="1") = 0
        - ➔ $H(S)$ = 0 **bit**
    - Valeur maximale
      - Prob (S="0") = Prob (S="1") = ½
        - ➔ $H(S)$ = 1 **bit**
  - Σ arbitraire, $|\Sigma| = N$
    - Valeur minimale
      - Prob ($S = \sigma$)=1 pour un σ donné, Prob ($S = \sigma$) = 0 pour tous les autres
        - ➔ $H(S)$ = 0 **bit**
    - Valeur maximale
      - Prob ($S = \sigma_i$) = Prob ($S = \sigma_j$), $\forall\ \sigma_i , \sigma_j \in \Sigma$
        - ➔ $H(S) = \log_2 N$ **bit**

Entropie

- Exemple de calcul d'entropie
  - Pile ou face
    - Alphabet {`pile`, `face`}
    - Probabilité d'occurrence des symboles (p_i): chaque symbole équiprobable avec une probabilité de ½
    - $H(S) = \Sigma_i \; p_i \log_2 1/p_i$
      - `pile` : ½ $\log_2$ (1 / ½) = ½ $\log_2$ 2 = ½ * 1 = ½
      - `face` : ½ $\log_2$ (1 / ½) = ½ $\log_2$ 2 = ½ * 1 = ½
      - $H(S)$ = ½ + ½ = 1 *bit*
  - Alphabet équiprobable
    - Alphabet = {`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`}
    - Probabilité d'occurrence des symboles (p_i):
      - chaque symbole équiprobable avec une probabilité de 1/26
    - $H(S) = \Sigma$ 1/26 $\log_2$ 1/(1/26)= 26*1/26 * $\log_{10}(26)/\log_{10}$ 2=
      = $\log_{10}(26)/\log_{10}$ 2 = 4.7 *bits (on peut vérifier :* $2^{4.7} = 25.99$*)*

Analyse fréquentielle vs. entropie

- Problème
  - L'entropie de Shannon
    - est définie à partir de *probabilités*
    - s'applique seulement aux sources markoviennes
  - Comment calculer/utiliser l'entropie sur
    - des sources non-markoviennes ?
    - des textes/séquences finies de symboles ?
- « Solution »
  - Fréquence de symbole
    - Soit $S_N = s_1, s_2, \ldots s_N$, $s_i \in \Sigma$, une séquence d'une source S, on définit:

$$f_i(S_N) = \frac{|\{j \mid s_j = \sigma_i\}|}{N}$$

  - Pseudo-entropie
    - Définie/calculée à partir des fréquences (au lieu de probabilités)

Pseudo-entropie

- Pour une séquence finie *SN* $\Psi(S_N) = \sum_i f_i(S_N) \log \frac{1}{f_i(S_N)}$
- Pour une séquence *S* $\Psi(S) = \lim_{N \to \infty} \Psi(S_N)$
- Pour une source d'information quelconque *S*
  - A chaque fois qu'on utilise la source « N fois »
  - On obtient une séquence SN différente de longueur N
  - On calcule la pseudo entropie $\Psi(S_N)$ de cette séquence, qui est elle-même une variable aléatoire
  - Sa valeur espérée $\overline{\Psi(S_N)}$ représente une pseudo-entropie de la source sur des séquences de longueur N
- On considère alors la pseudo-entropie de la source comme étant la limite de cette valeur espérée

$$\Psi(S) = \lim_{N \to \infty} \overline{\Psi(S_N)}$$

Entropie vs. pseudo-entropie

- Pour les sources markoviennes
  - La pseudo-entropie d'une séquence générée par la source va s'approcher de l'entropie
  - Cette convergence est bonne lorsque la taille de la sous-séquence est grande, parce que les fréquences f_i s'approchent des probabilités p_i (loi des grands nombres)
    - Quand $N \rightarrow \infty$, alors $\Psi(S_N) \rightarrow H(S)$
    - Si N est trop petit, alors
      - déductions faites à partir des f_i non valable statistiquement
        ➔ cryptanalyse difficile (voir TP 1)
- Pour les sources non-markoviennes
  - L'entropie *H(S)* n'est pas vraiment définie,
    - On utilise $\Psi(S)$ à la place (outil de calcul d'entropie TP1)
    - On écrira dans le reste du cours « H(S) »,
      mais on veut vraiment dire $\Psi(S)$ …

Interprétation de l'entropie d'une source

- Interprétation de $H(S)$
  - 1er théorème : Chaque symbole émit par S peut être codé individuellement avec en moyenne $H(S)$ bits
  - Et si on permet que le codage regroupe 2 lettres à la fois ?
    - ➔ Par 1er théorème on peut coder chaque digramme (2 symboles) avec $H(S^2)$ bits, soit $H(S^2)/2$ bits par symbole
    - ➔ Mais si $H(S^2)/2 \leq H(S)$, donc on peut avoir un gain en compression
- Taux de compression
  - Sans compression
    - ➔ log N bits par symbole, dans le pire cas (entropie maximale)
  - Avec compression par bloc de b symbole
    - ➔ $H(S^b)/b$ bits par symbole
  - Taux de compression = $\frac{H(S^b)/b}{\log N}$

Source markovienne vs. non markoviennes

- Source markovienne
  - Si S est markovienne, alors $H(S^b) = b*H(S)$
    - Conséquence: Aucun gain de compression en codant par bloc
    - Intuition: Il n'existe pas de corrélation entre les symboles (distribution de probabilité indépendante), et chaque symbole doit être codé individuellement
- Source non markovienne
  - En général $H(S^b) \leq b*H(S)$
    - Conséquence1: Il y a en général un gain de compression en codant par bloc
    - Intuition:
      - Les probabilités des symboles dépendent des symboles antérieurs
      - Cette « dépendance » statistique peut être exploité par le codage pour réduire le nombre de symboles ou le nombre de bits dans leur codage

        P.ex. en français
        - la lettre « u » suit (presque toujours) la lettre « q »
        - Un sujet est suivi d'un verbe, p.ex. « Je_ » doit être suivi d'un verbe conjugué à la 1e personne du singulier
      - Le gain de compression devrait augmenter en considérant des tailles de blocs plus grandes

Entropie du langage de la source

- En théorie,
  - plus la taille de bloc *b* est grande,
    plus le taux de compression est élevé (jusqu'à une certaine limite)
- Langage associé à une source *S*
  - ensembles de chaines finies générées par *S*
- L'entropie H_L du *langage* associé à la source *S,*

$$H_L(S) = \lim_{b \to \infty} \frac{H(S^b)}{b}$$

  - est le minimum de bits nécessaires (en moyenne) pour coder chaque symbole de chaînes émises par S, même si on permet de coder avec des tailles de blocs arbitraires
  - représente la limite ultime de compression

CRYPTOGRAPHIE I – MODÈLE DE SHANNON RÉVISÉ

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Modèle de Shannon révisé

Ève

Source σ Codeur τ Chiffr. τ' Déchif. τ Décodeur σ Récepteur

Alice

Irène

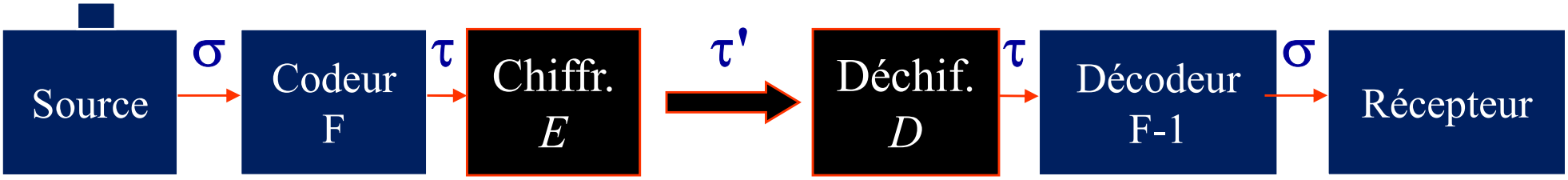
Bob

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…

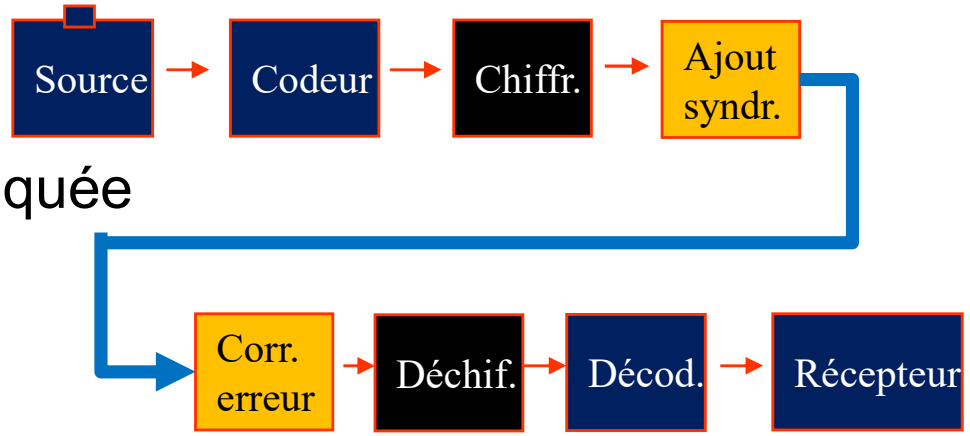
Algorithme de chiffrement – Concepts généraux

- 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
      - <u>après</u> le chiffrement et
      - <u>avant</u> le déchiffrement



- La correction d'erreur
  - Constitue une forme de protection de l'<u>intégrité</u> des messages
    - Protège contre des erreurs aléatoires (menace accidentelle)
      - P.ex. erreur de transmission due au bruit, interférence accidentelle, etc.
    - <u>Ne protège pas</u> 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 → 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
    - $k \in \{1,\ldots,26\}$
- 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 $a, b \in \{1,\ldots,26\}$

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
  - « HELLOWORLD » devient :

π

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

  - « imuubxbwua »

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 OI 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 bc 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

  - « hloolelwrd »

Algorithmes classiques

- Algorithme de transposition (« bit shifting »)
  - Chiffrons ce texte avec un « bâton » de taille 6 et commençons au 3e « trou de ceinture » (3e 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 → chiffres de 1 à 26
  - Clé
    - $K = k_1\, k_2 \ldots k_m$, 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	19		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 MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Méthodes de cryptanalyse de base

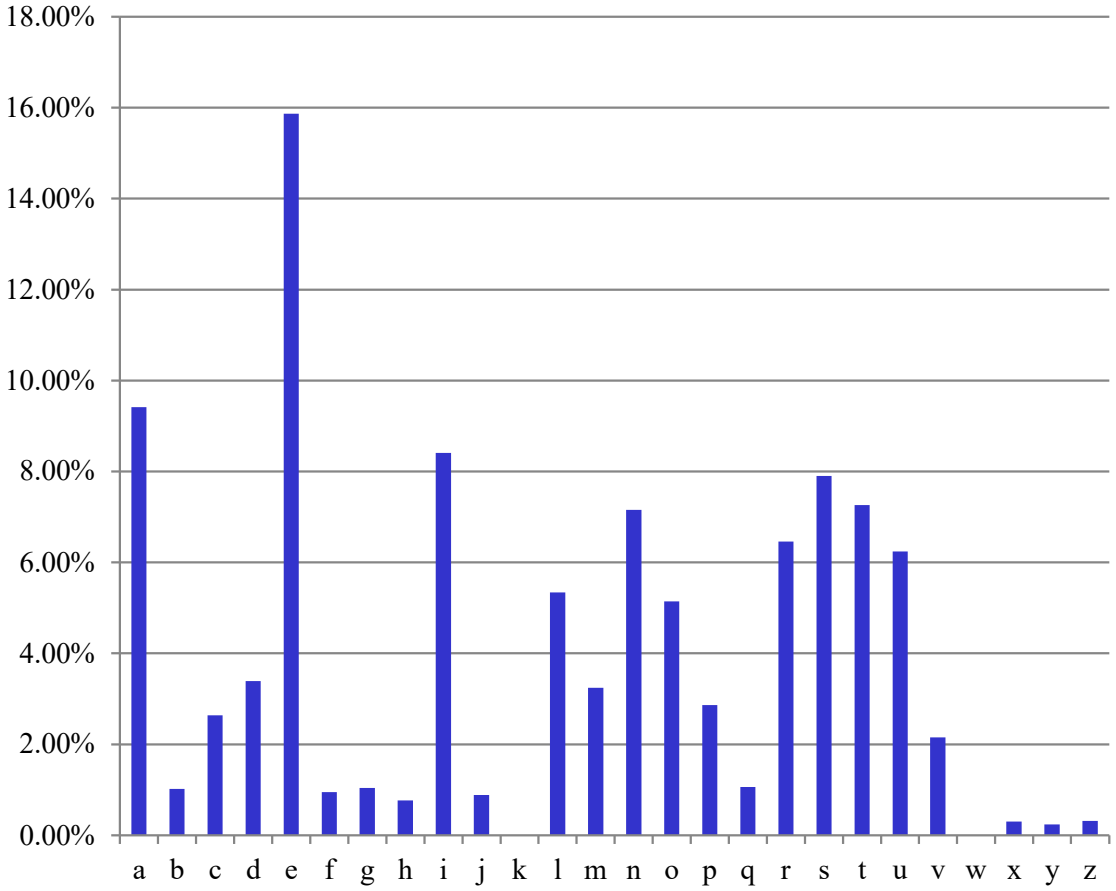
- Force brute
  - Essaie de toute les clés
    1. Déchiffrer le texte chiffré avec la clé à essayer
    2. Voir si le résultat est cohérent
    - Paramètre de difficulté
      - Taille de l'espace de clés
        N bits de clés = 2^N clés possibles
      - Génération de clés non-aléatoire/non-uniforme
        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

Méthodes de cryptanalyse de base

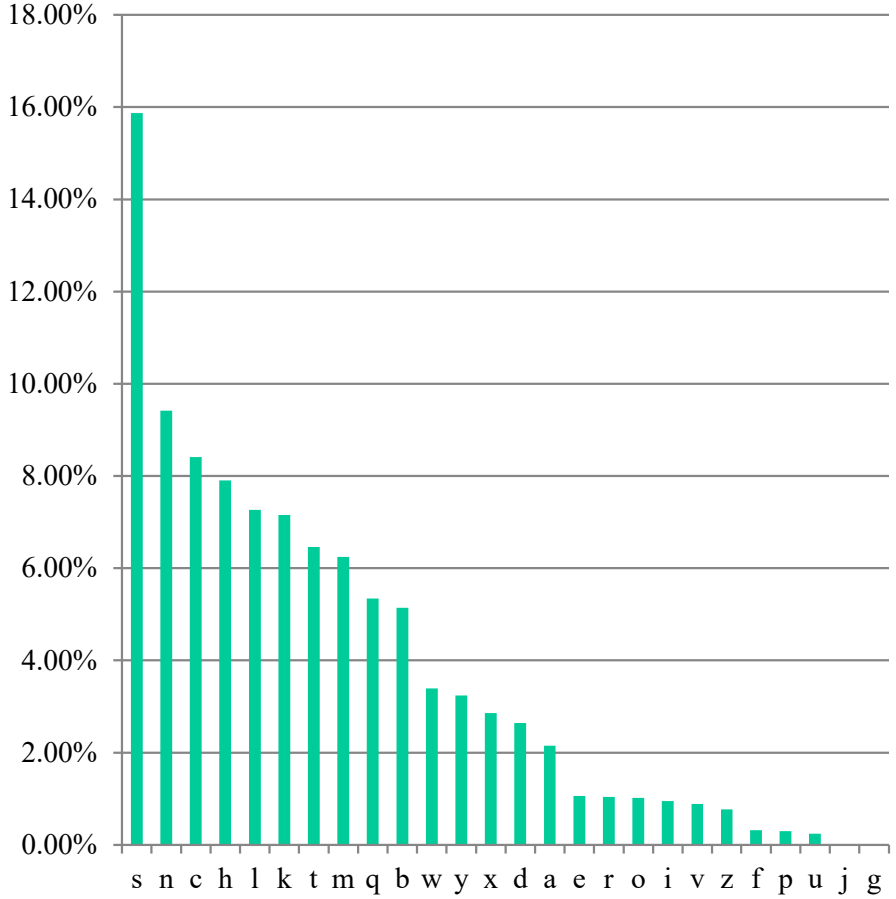
- Analyse fréquentielle
  - Méthode
    1. Établir/retrouver fréquences des symboles de la source
    2. Calculer les fréquences des symboles chiffrés obtenus
    3. Comparer histogrammes de fréquences
    4. Établir relations entre symboles chiffrés et symboles de sources
    5. Essayer de déchiffrer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

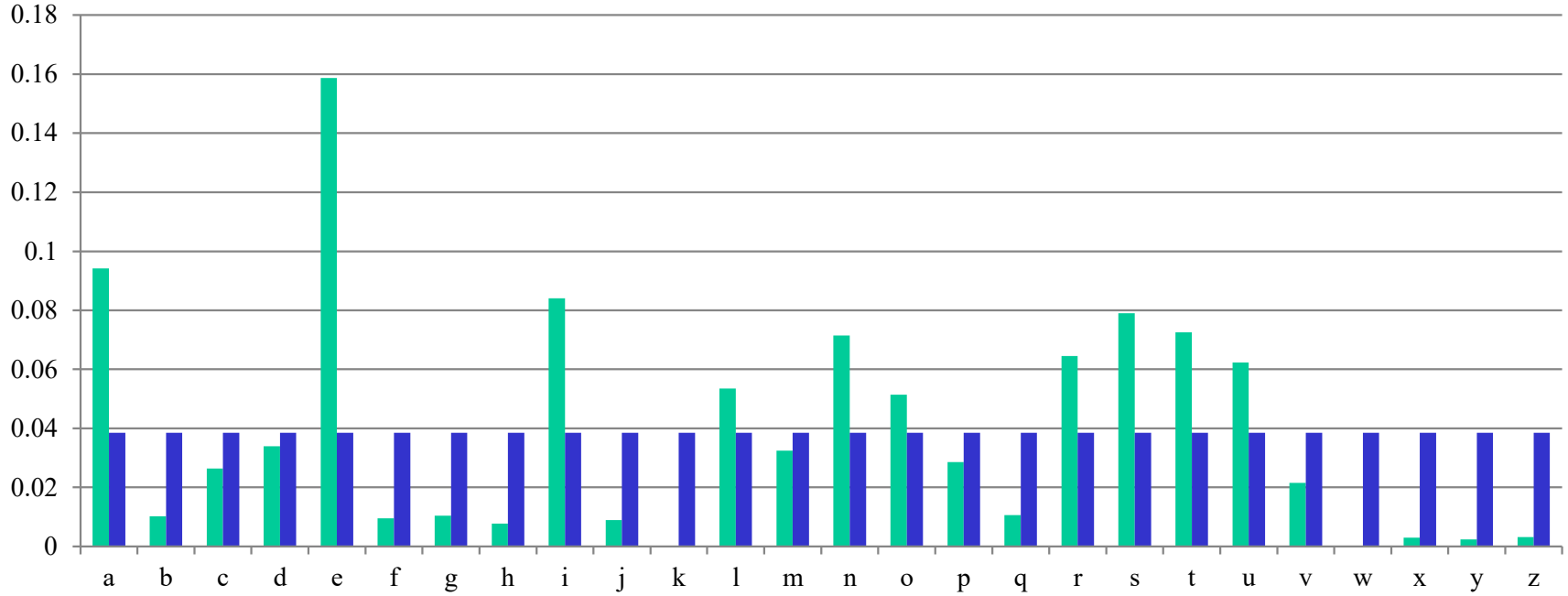


- Histogramme (ordonné) de d'un texte chiffré



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 on 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 donné par la formule suivante

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

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

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	-	2627	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	218	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	876	478	721	588	42	494	1596	3912	134	116	1787	-	436	2	-	19998
Total:	7069	1418	2059	3770	11645	1549	1906	6739	4483	131	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")

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Questions ?