


1. Introduction à la cryptographie


1.1. Communications sécurisées

- Confidentialité des données transmises : importance
 - Contexte personnel 
 - **Protection de la vie privée** : les données personnelles (médicales, bancaires, communications privées) doivent être gardées confidentielles pour protéger le droit des individus à la vie privée
 - **Protection contre le vol d'identité** : la confidentialité aide à empêcher l'accès non-autorisé aux données personnelles, pour réduire les risques de vol d'identité et de fraude
 - **Sécurité personnelle** : elle peut être compromise si des informations sensibles (adresse personnelle, emploi du temps) tombent entre de mauvaises mains

1.1. Communications sécurisées

- Confidentialité des données transmises : importance
 - Contexte professionnel 
 - **Avantage compétitif** : stratégies d'entreprises et propriétés intellectuelles sont critiques pour les compagnies. Les dévoiler pourrait causer des pertes ou permettre leur exploitation par les concurrents
 - **Confiance des clients** : les entreprises qui traitent des données clients doivent maintenir leur confidentialité. Toute fuite serait dommageable pour la réputation et pourrait avoir des répercussions légales
 - **Obligation légale** : pour certaines industries
 - https://fr.wikipedia.org/wiki/Règlement_général_sur_la_protection_des_données



1.1. Communications sécurisées

- Confidentialité des données transmises : importance
 - Contexte gouvernemental 
 - **Sécurité nationale** : la confidentialité des opérations gouvernementales, stratégies de défense et informations classées est vitale pour la sécurité nationale
 - **Relations diplomatiques** : une communication confidentielle entre les gouvernements garantit des négociations et une diplomatie sans interférence
 - **Enquêtes policières et judiciaires** : garder certaines informations confidentielles facilite les enquêtes menées et protège l'identité des informateurs et témoins



1.1. Communications sécurisées

- Risques associés aux communications
 - Interception 🧠
 - **Écoute clandestine** : des individus ou entités non-autorisés peuvent intercepter les communications (e-mails, appels téléphoniques, messages) pour récolter des informations sensibles
 - **Fuite de données** : des pirates peuvent infiltrer des systèmes ou réseaux pour voler des données, en accédant à des informations confidentielles enregistrées
 - Falsification 🤡
 - **Altération de données** : des agents malveillants peuvent modifier l'information durant sa transmission, générant des information fausses ou fallacieuses
 - **Intégrité des données** : elle peut être attaquée, afin d'affecter le processus de prise de décision

1.1. Communications sécurisées

- Risques associés aux communications
 - Imitation 
 - **Spoofing** : falsifier l'identité d'un expéditeur ou d'un destinataire, pour accéder de manière non-autorisée à des informations ou un système
 - **Hameçonnage (phishing)** : faux e-mails ou faux sites web piégeant les individus en les poussant à divulguer des informations sensibles, comme leur identifiants et mots de passes
 - Déni de service 
 - **Attaques DDoS** (attaque par déni de service distribuée) : inonder un système ou un réseau avec un trafic excessif pour rendre des services disponibles aux utilisateurs légitimes
 - **Coupure de service** : toute action interrompant la disponibilité normale d'un service, empêchant l'accès aux informations et systèmes critiques

1.1. Communications sécurisées

- Risques associés aux communications
 - Menaces internes 
 - Menaces de cybersécurité qui proviennent d'**utilisateurs autorisés**, c'est-à-dire d'employés, de fournisseurs ou de partenaires commerciaux, qui abusent, intentionnellement ou non, de leur accès légitime, ou dont les comptes sont détournés par des cybercriminels
 - Menaces physiques 
 - **Le vol ou la perte de matériel** (portables, clés USB) contenant des informations sensibles peut conduire à des accès non-autorisés
 - **Surveillance physique** : l'observation direct des canaux de communication physique, contournant ainsi les mesures de sécurité numérique

- **Définition**

- Discipline de la cryptologie s'attachant à protéger les communications et les données en les convertissant dans un format illisible, sauf pour ceux possédant la clef de déchiffrement
- Cela implique donc des techniques pour écrire l'information de telle manière que seules les parties autorisées peuvent y accéder et la comprendre
- Notes :
 - La **cryptologie** englobe
 - Cryptographie, l'écriture secrète
 - Elle-même se subdivise en chiffrement et encodage (voir ci-après)
 - Cryptanalyse, l'analyse de cette dernière
 - La cryptographie se distingue de la **stéganographie** qui fait passer inaperçu un message dans un autre message
(p. ex. message caché dans une image, avec OutGuess)

1.2. Bases de la cryptographie

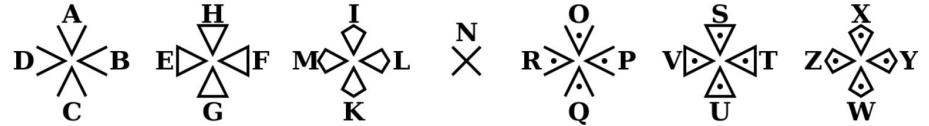
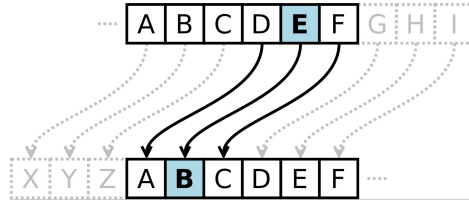
- Rôles de la cryptographie, dont **trois propriétés** des données qu'elle vise à garantir
 - **Confidentialité** : encoder les données sorte que, même si elles sont interceptées, elles seront inintelligibles pour les utilisateurs non-autorisés
 - **Intégrité** : maintenir l'intégrité des données en détectant toute altération non-autorisée (voir fonctions de hachage et *checksums*)
 - **Authentification** : les signatures numériques et protocoles d'authentification utilisent des techniques cryptographiques pour garantir que les messages échangés proviennent de sources légitimes
 - **Non-répudiation** : les signatures numériques permettent de s'assurer qu'un contrat ne peut être remis en cause par l'une des parties
 - **Transactions sécurisées** : e-commerce, banques en ligne
 - **Protection de la vie privée** : protéger les données et communications personnelles des accès non-autorisés

1.3. Évolution des techniques

- Antiquité

- **Chiffrement par substitution monoalphabétique**

- Substituer dans un message chacune des lettres de l'alphabet par une autre
 - Chiffrement par décalage (César, ROT13), chiffre des Templiers



- **Chiffrement par transposition (ou permutation)**

- Changer l'ordre des lettres (anagrammes)
 - Scytale



1.3. Évolution des techniques

- Renaissance
 - **Chiffrement par substitution polyalphabétique** : pour une lettre donnée, son chiffrement ne sera pas toujours le même selon l'endroit du texte
 - **Chiffre d'Alberti** (1404 - 1472)
 - Les substitutions sont définies par le positionnement relatif de deux disques concentriques
 - Ce positionnement peut changer selon l'endroit du texte
 - « dans mon message, j'écrirai un D majuscule et, à partir de ce point, k ne signifiera plus B mais D »



1.3. Évolution des techniques

- Renaissance
 - **Chiffrement par substitution polyalphabétique**
 - **Chiffre de Trithémius** (1462 - 1516)
 - On chiffre la première lettre du message clair avec la première ligne, la deuxième lettre avec la deuxième ligne, etc.
 - Cela revient à une suite de décalages de César
 - La première lettre n'est pas décalée, la deuxième est décalée d'un cran dans l'alphabet, la troisième de deux crans, etc.

	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

1.3. Évolution des techniques

- Renaissance
 - Chiffrement par substitution polyalphabétique
 - Chiffre de Vigenère (1523 - 1596)
 - Introduit la notion de clé

Lettre de la clé	Lettre en clair																									
	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Texte en clair : j'adore écouter la radio toute la journée
Clé répétée : M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
^ ^^^
| |Colonne 0, ligne I : on obtient la lettre W.
| |Colonne D, ligne S : on obtient la lettre V.
| Colonne A, ligne U : on obtient la lettre U.
Colonne J, ligne M : on obtient la lettre V.

Texte chiffré : V'UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY
Clé répétée : M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
^ ^^^
| |Ligne I, on cherche W : on trouve la colonne 0.
| |Ligne S, on cherche V : on trouve la colonne D.
| Ligne U, on cherche U : on trouve la colonne A.
Ligne M, on cherche V : on trouve la colonne J.



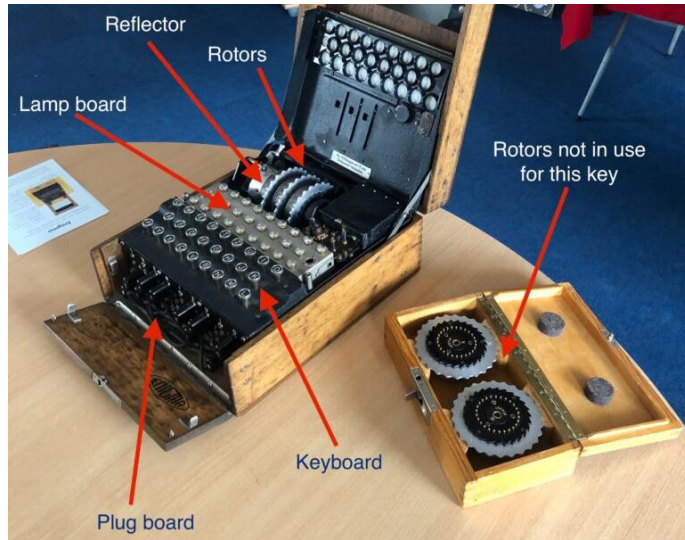
Arithmétique modulaire

$\text{Chiffré}[i] = (\text{Texte}[i] + \text{Clés}[i]) \bmod 26$

$\text{Texte}[i] = (\text{Chiffré}[i] - \text{Clés}[i]) \bmod 26$

1.3. Évolution des techniques

- Seconde guerre mondiale
 - Enigma
 - Famille de machines de chiffrement électromécaniques
 - Version commerciale brevetée par Arthur Scherbius en 1918
 - Plus tard, version militaire utilisée par l'armée allemande



Lorsqu'une touche du clavier est pressée :

- L'ampoule correspondant à la lettre chiffrée s'allume
- Le système de rotors tourne d'un cran
 - Presser la même lettre deux fois consécutives donnera deux lettres chiffrées différentes (chiffrement polyalphabétique)

1.3. Évolution des techniques

- Seconde guerre mondiale
 - Enigma : design et combinaisons
 - 3 rotors crantés, sur 3 emplacements (droite, milieu, gauche) à choisir parmi 5 rotors différents
 - $5 \times 4 \times 3 = 60$
 - Chaque rotors possède 26 crans
 - $26 \times 26 \times 26 = 17\,576$
 - Tableau de connexions avec 10 câbles : permet de définir des permutations pour 10 paires de lettres (version militaire seulement)
 - $$\frac{26!}{6! \times 10! \times 2^{10}} = 150\,738\,274\,937\,250$$
 - **Total = 158 962 555 217 826 360 000 configurations possibles**


1.3. Évolution des techniques

- Seconde guerre mondiale
 - Enigma : **fonction de chiffrement involutive** (*self-reciprocal cipher*)
 - Les processus de chiffrement et déchiffrement sont les mêmes
 - On déchiffre en tapant le message chiffré sur une machine Enigma avec la même configuration que celle ayant servi au chiffrement
 - Propriété garantie par l'utilisation d'un **réflecteur** à la suite du dernier rotor
 - Facilite l'utilisation
 - Mais empêche de substituer une lettre à elle-même dans le texte chiffré
 - Vulnérabilité à une « attaque à texte clair connu »
 - Conception de la « bombe » par Alan Turing pour trouver la configuration d'Enigma utilisée, c.-à-d. trouver la clé
- Note : les Alliés finir par capturer des exemplaires de ces machines...



1.3. Évolution des techniques

- Cryptologie moderne
 - **1940s - Avènement de l'informatique** : (classique → moderne) les ordinateurs révolutionnent la cryptographie, permettant des calculs plus rapides et des algorithmes plus complexes
 - **1970s - Développements mathématiques** : concepts comme la théorie des nombres, les structures algébriques et la théorie de la complexité
 - **1976 - Cryptographie asymétrique** : introduction généralement attribuée à Whitfield Diffie et à Martin Hellman
 - **1977 - Algorithme RSA** : développement par Rivest, Shamir et Adleman, aujourd'hui fondamental pour le chiffrement et les signatures numériques
 - **1980s - Fonctions de hachage cryptographiques** : leur émergence offre des moyens de vérifier l'authenticité et l'intégrité des données
 - **1990s - Cryptanalyse différentielle et linéaire** : développement de techniques capables de casser des algorithmes de chiffrement symétriques
 - **2001 - Advanced Encryption Standard (AES)** : adoption, en remplacement de Data Encryption Standard (DES), du fait de sa robustesse et de son efficacité

1.4. Perspectives actuelles

- Informatique quantique et cryptographie post-quantique 
 - **Défis posés** : le développement des ordinateurs quantiques constitue une menace pour les méthodes de chiffrement actuelles, car ils pourraient résoudre, de manière efficiente, les problèmes mathématiques sur lesquels ces algorithmes reposent
 - RSA : décomposition en facteurs premiers
 - (EC)DH : problème du logarithme discret
 - **Solutions explorées** : développement d'algorithmes à résistance quantique basés sur des problèmes d'optimisation sur les réseaux euclidiens
 - Problème des vecteurs courts (SVP)
 - Problème de l'apprentissage avec erreur (LWE)

1.4. Perspectives actuelles


- Blockchain et crypto-monnaies 
 - **Défis posés** : résoudre les problèmes d'extensibilité (*scalability*), d'anonymat et de sécurité, sans compromettre la décentralisation
 - **Solutions explorées** : nouveaux mécanismes de consensus et techniques comme la ***zero-knowledge proof***
- Chiffrement homomorphe 

Algorithme permettant d'effectuer certaines opérations mathématiques sur des données chiffrées sans avoir à les déchiffrer d'abord

→ Permet de confier des calculs à un agent externe, sans que les données ni les résultats soient accessibles à cet agent

 - **Défis posés** : améliorer l'efficacité
 - **Solutions explorées** : concernent principalement le *cloud computing*

1.5. Terminologie

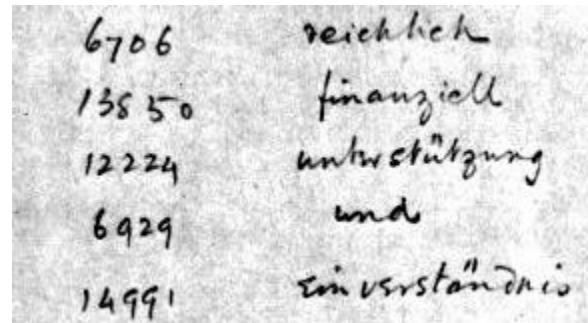
- **Chiffrement** (*encryption*) : processus de conversion d'un message en clair, m (données intelligibles) en un message chiffré c (données inintelligibles), en utilisant un algorithme E et une clé k
 - $E(k_e, m) = c$
 - Déterministe ou probabiliste
 - **Déchiffrement** (*decryption*) : processus réciproque du chiffrement, transformant c en m en utilisant une clé appropriée
 - $D(k_d, c) = m$
 -  Toujours déterministe
- (E, D) : cryptosystème
- **Primitive cryptographique** : opération fondamentale cryptographique, de bas niveau et bien établi (*building blocks*), sur la base de laquelle est bâti tout système de sécurité informatique, algorithme ou protocole
 - Exemples : Chiffrement de flux, par bloc, fonction de hachage, MAC, PRNG, RSA, DH

1.5. Terminologie

- **Cryptanalyse** : ensemble des techniques d'analyse des systèmes cryptographiques, ayant pour but de trouver des vulnérabilités et d'être capable de déchiffrer sans connaître la clé. Le processus par lequel on tente de comprendre un message en particulier est appelé une **attaque**.
- **Adversaire** : entité cherchant à casser la sécurité d'un système cryptographique
 - Cryptanalyste : adversaire théorique (individu ou algorithme)
 - Attaquant : quelqu'un essayant activement d'exploiter des failles du système cryptographique
- **Alice et Bob** : pour les « personne A » et « personne B » qui cherchent à communiquer
 - D'autres noms sont utilisés, comme Eve (*eavesdropper*), Mallory (*malicious attacker*), ou Oscar (*opponent*)

1.5. Terminologie

- **Chiffre (cipher) vs code** : le chiffrement d'un message s'opère au niveau des caractères de celui-ci, caractères individuels ou bien petits groupes de caractères.
Un code, lui, encrypte un message par remplacement des mots ou des phrases. Il opère donc au niveau de la signification du message.
 - Un message peut être encodé avant chiffrement pour compliquer la cryptanalyse
 - **Nécessité d'un livre-code** ou d'une table de correspondance
 - Exemples :
« Les sanglots longs des violons de l'automne... » = « Sabotez les voies ferroviaires ! »



Extrait du décodage du télégramme diplomatique Zimmermann (1917)

- **Cryptographie symétrique** (ou à clé secrète) : la même clé est utilisée pour le chiffrement et le déchiffrement
 - Problématique de la distribution des clés, qui doit être sécurisée
 - Chiffrement par flot (ou de flux) : opère sur un flux continu de données (*i.e.* pas de découpage). Exemples : A5/1, RC4
 - Chiffrement par bloc : opère sur des données découpées en blocs de taille généralement fixe. Exemples : DES, AES
 - Code d'authentification de message (MAC) : code accompagnant des données dans le but d'assurer l'intégrité de ces dernières

1.6. Classification

- **Cryptographie asymétrique** (ou à clé publique) : système cryptographique utilisant des paires de clés, une publique pour le chiffrement, une privée pour le déchiffrement
 - Chiffrement asymétrique (RSA, DH, ...)
 - Schéma de signature numérique : mécanisme permettant d'authentifier l'auteur d'un document électronique et d'en garantir la non-répudiation
- **Cryptographie hybride** : combine
 - un chiffrement asymétrique (plus sécurisé) pour l'échange d'une clé symétrique,
 - et un chiffrement symétrique (plus rapide) pour les données transmises.
 - souvent couplées avec un mécanisme d'authentification (p. ex. TLS, *Transport Layer Security*)



2. Chiffrements symétriques

2.1. Masque jetable

- **Chiffre de Vernam** (1917) : **XOR** entre une clé secrète k et le message clair m pour produire le chiffré c

$$m \oplus k = c$$

Chiffrement

$$\begin{array}{rcccccccc} & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & m \\ \oplus & & & & & & & & & \\ & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & k \\ \hline = & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & c \end{array}$$

Déchiffrement

$$\begin{array}{rcccccccc} & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & c \\ \oplus & & & & & & & & & \\ & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & k \\ \hline = & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & m \end{array}$$

On peut également retrouver k à partir de m et c : $m \oplus c = k$

2.1. Masque jetable

- **Vulnérabilité** : utiliser deux fois la même clé k pour deux messages clairs m_1 et m_2 permet, à quelqu'un qui aurait intercepté m_2 , c_1 et c_2 , de déchiffrer m_1 sans la clé k

$$m_1 \oplus k = c_1$$

$$m_2 \oplus k = c_2$$

$$m_1 \oplus c_1 = m_2 \oplus c_2$$

$$m_1 = (m_2 \oplus c_2) \oplus c_1$$

Note : par commutativité, on a aussi $m_1 \oplus m_2 = c_1 \oplus c_2$,

→ Crée une vulnérabilité en interceptant uniquement c_1 et c_2

- La sécurité du chiffrement repose donc sur une **utilisation unique de la clé** (ou masque) → Masque jetable ou **one-time pad (OTP)**
- La fonction **XOR** nécessite que la clé soit de la **même longueur que le message**
Très peu pratique (→ **chiffrements de flux**), surtout si la clé doit être gardée secrète

2.1. Masque jetable

- Si la clé secrète k (à usage unique et de même longueur que m) **est parfaitement aléatoire**, alors OTP permet un « secret parfait » (*perfect secrecy*), concept introduit par Claude Shannon (1949)

OTP est le seul algorithme offrant une sécurité parfaite

Aucune information* sur le message clair m ou la clé k ne peut être extraite** du chiffré c , même en disposant d'une puissance de calcul illimitée
→ OTP résiste aux attaques par force brute avec un ordinateur quantique

- Clé parfaitement aléatoire : à chaque position de la séquence, les valeurs 0 et 1 sont équiprobables
→ Distribution uniforme (discrète) : $k \sim U(0^l, 1^l)$, où l est la longueur de la clé

*Information telle que définie par Claude Shannon

**La longueur de m est cependant connue et égale à celle de c

2.1. Masque jetable

- Exemple pour $k \in K = \{0, 1\}^3$

$$\begin{array}{llll} \oplus \begin{array}{l} 111 \ c \\ 000 \ k_1 \\ \hline 111 \ m_1 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 001 \ k_2 \\ \hline 110 \ m_2 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 011 \ k_3 \\ \hline 100 \ m_3 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 111 \ k_4 \\ \hline 000 \ m_4 \end{array} \\ \oplus \begin{array}{l} 111 \ c \\ 110 \ k_5 \\ \hline 001 \ m_5 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 100 \ k_6 \\ \hline 011 \ m_6 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 101 \ k_7 \\ \hline 010 \ m_7 \end{array} & \oplus \begin{array}{l} 111 \ c \\ 010 \ k_8 \\ \hline 101 \ m_8 \end{array} \end{array}$$

Avec une clé k **inconnue de l'adversaire** et générée de manière parfaitement aléatoire, le message chiffré c « 111 » pourrait résulter du chiffrement de **tous les messages clairs possibles** (m_1 à m_8) de façon équiprobable (1 chance sur 8).

Propriété : un **XOR** entre n'importe quelle séquence et une séquence aléatoire produit une séquence également aléatoire.

2.1. Masque jetable

$$\Pr[E(k, m_0) = c] = \Pr[k \oplus m_0 = c] \quad (1)$$

$$= \frac{|k \in \{0, 1\}^m : k \oplus m_0 = c|}{|\{0, 1\}^m|} \quad (2)$$

$$= \frac{1}{2^m} \quad (3)$$

$$\Pr[E(k, m_1) = c] = \Pr[k \oplus m_1 = c] \quad (4)$$

$$= \frac{|k \in \{0, 1\}^m : k \oplus m_1 = c|}{|\{0, 1\}^m|} \quad (5)$$

$$= \frac{1}{2^m} \quad (6)$$

donc

$$\begin{aligned} \Pr[E(k, m_0) = c] &= \Pr[E(k, m_1) = c] && \forall m_0, m_1 \in M, \text{ avec } |m_0| = |m_1| \\ &&& \forall c \in C \\ \text{avec } M &= K = \{0, 1\}^m \end{aligned}$$

- **Convention d'écriture**

- M : espace des messages en clair

- $M = \{0, 1\}^l$

- $|M| = 2^l$

- C : espace des messages chiffrés

- K : espace des clés

- La sécurité parfaite d'OTP requiert que $|K| \geq |M|$

- **Contenu en information** (ou auto-information)

- Défini par Claude Shannon, 1948

Soit une variable discrète X qui prend une valeur x

(p. ex. la valeur donnée à une position sur une séquence numérique)

Le contenu en information $I_X(x)$ de cet évènement est :

$$I_X(x) := -\log[p_X(x)] = \log\left(\frac{1}{p_X(x)}\right)$$

avec $p_X(x) \neq 0$ et $I_X(x) \in [0, +\infty]$

En utilisant un logarithme de base 2, l'unité de $I_X(x)$ est le shannon (Sh) ou bit ;
le hartley (Hart) pour la base 10 ; le *natural unit of information* (nat) pour la base e

- **Entropie de Shannon**

- Elle correspond au contenu en information moyenné sur toutes les positions de la séquence. Elle s'écrit $\mathbf{H(X)}$.

$$\begin{aligned}\mathbf{H(X)} &= \sum_x -p_X(x) \log p_X(x) \\ &= \sum_x p_X(x) \mathbf{I_X(x)} \\ &\stackrel{\text{def}}{=} \mathbf{E [I_X(X)]},\end{aligned}$$

avec $\mathbf{H(X)} \in [0, +\infty]$

- Pour une séquence de longueur donnée, **l'entropie de Shannon est maximum** si les différentes valeurs possibles à chaque position sont équiprobables, c.-à-d. si la séquence est parfaitement aléatoire (loi uniforme ; **maximum d'incertitude**)

Note : maximum mais pas ∞ , car séquence de longueur finie

- **Entropie de Shannon**

- Exemple : entropie de la séquence de valeurs décimales « 31100427 »

$$p(0) = 2/8 = 0,25$$

$$p(1) = 2/8 = 0,25$$

$$p(2) = 1/8 = 0,125$$

$$p(3) = 1/8 = 0,125$$

$$p(4) = 1/8 = 0,125$$

$$p(5) = 0/8 = 0$$

$$p(6) = 0/8 = 0$$

$$p(7) = 1/8 = 0,125$$

$$p(8) = 0/8 = 0$$

$$p(9) = 0/8 = 0$$

$$\begin{aligned} H(X) &= \\ &-0,25 \log_2(0,25) + \\ &-0,25 \log_2(0,25) + \\ &-0,125 \log_2(0,125) + \\ &-0,125 \log_2(0,125) + \\ &-0,125 \log_2(0,125) + \\ &-0,125 \log_2(0,125) \\ &= 2,125 \text{ bits} \end{aligned}$$

- **Entropie conditionnelle**

- Pour deux séquences X et Y :

$$H(Y|X) = - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)}$$

- $H(Y|X) \leq H(Y)$
 $H(Y|X) = H(Y)$ ssi Y et X indépendantes

- Pour un message M et son chiffré C , on définit ainsi :

- **Sécurité parfaite** : $H(M|C) = H(M)$

- Connaître C n'apporte aucune information sur M (sauf sa longueur)
- La probabilité qu'un adversaire* casse le chiffrement est nulle

- **Sécurité inconditionnelle** : $H(M|C) \lesssim H(M)$

- La probabilité qu'un adversaire* casse le chiffrement est **négligeable**

* disposant d'une puissance de calcul illimitée

2.4. Générateur d'aléa

- OTP requiert une clé parfaitement aléatoire
 - Peut être obtenue grâce à un générateur de nombres aléatoires physique
True random number generator (TRNG)
Basé sur la variation aléatoire de propriétés physiques : p. ex. *timestamp*



Le mur de lampes à lave de Cloudflare

- **Limitation** : produisent un nombre limité de bits aléatoires par seconde

- **Générateur de nombres pseudo-aléatoires**

Pseudorandom number generator, PRNG

Algorithme qui génère une séquence de nombres qui apparaît aléatoire, mais qui est déterminée par une valeur initiale appelée « graine » (*seed*)

⚠ Sortie **déterministe** → **même graine = même sortie**

- Exemple : **générateur congruentiel linéaire**

$$X_{n+1} = (a \cdot X_n + c) \bmod m,$$

où X_n est le nombre pseudo-aléatoire courant,
et a , c et m sont des constantes, $m > 0$

Sans nécessairement être aléatoire, la graine doit être choisie de sorte à ne pas être prédite

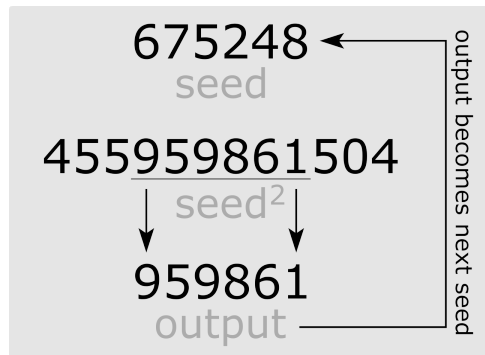
La valeur de X_0 est celle de la graine, à partir de laquelle la formule se répète de façon récurrente, pour générer une séquence de nombres.

Limitation : un mauvais choix des constantes peut générer une suite périodique

P. ex., pour le générateur d'UNIX, $a = 1103515245$, $c = 12345$ et $m = 2^{31}$

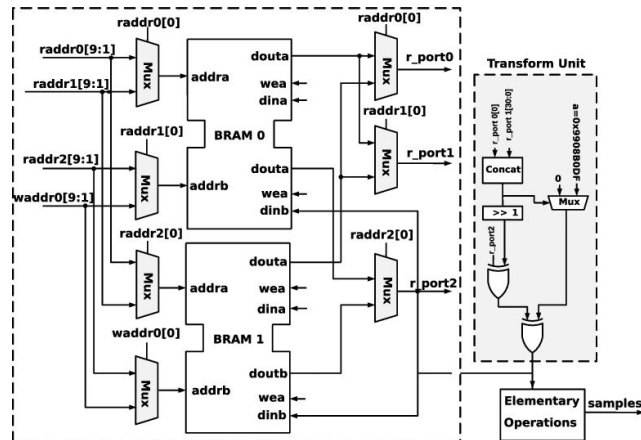
- **Générateur de nombres pseudo-aléatoires**

- Autres exemples :



Méthode du carré médian

Pour une graine de $2n$ chiffres, la période maximum est de 10^n (un peu plus, si nombre impair de chiffres).

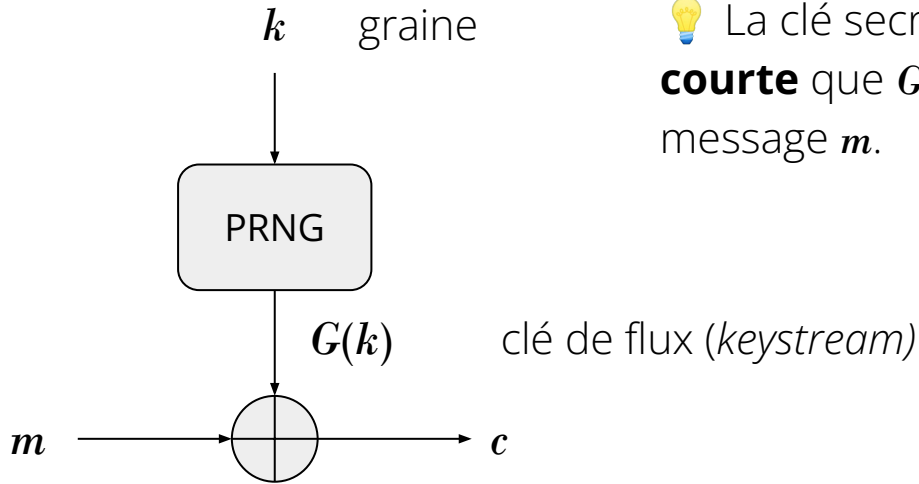


Mersenne Twister (MT19937)

Sa période maximum est de $2^{19937} - 1$. Mais **insuffisant pour la cryptographie**, car il existe des algorithmes qui permettent d'en prédire le comportement.

2.5. Chiffrement de flux

- Architecture générale



💡 La clé secrète échangée est k . Elle est **beaucoup plus courte** que $G(k)$ qui est de la même longueur que le message m .

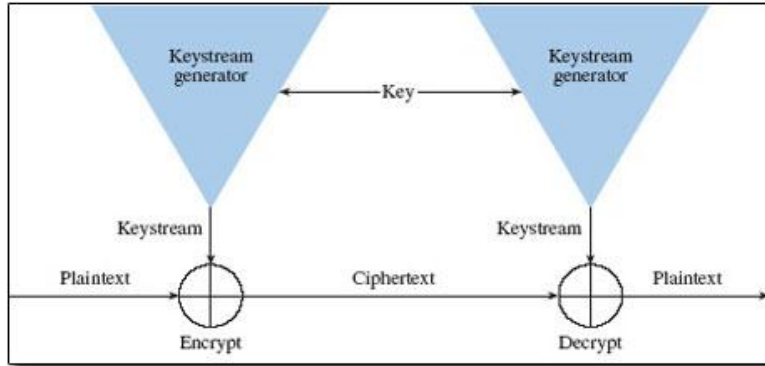
$$E(k, m) = m \oplus G(k)$$

$$D(k, c) = c \oplus G(k)$$

On peut donc considérer un chiffrement continu comme un **XOR** combiné à un PRNG. Ce dernier doit impérativement être **sécurisé pour la cryptographie**.

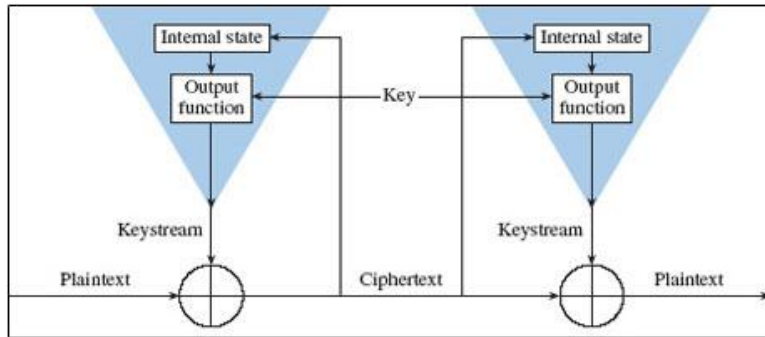
2.5. Chiffrement de flux

- Deux catégories



Synchrone

Calcule le flux de clé en fonction de la clé secrète k , mais indépendamment du texte clair m et du chiffré c .



Asynchrone (ou auto-synchronisant)

Calcule chaque bit du flux de clé comme une fonction de k et des n précédents bits de c . Avantage : le récepteur se synchronise avec le générateur après réception de n bits de c , ce qui facilite la correction d'erreurs. Désavantage : 1 bit d'erreur à la transmission entraînera n bits d'erreurs pour le récepteur (propagation d'erreur).

2.5. Chiffrement de flux

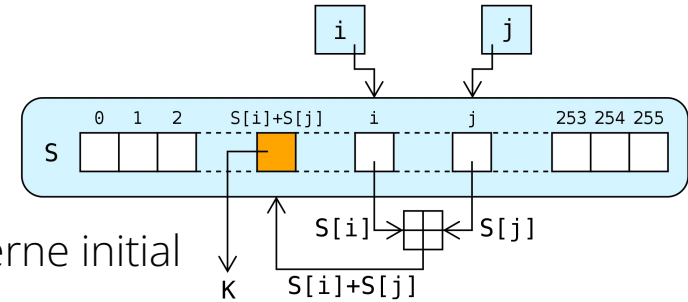
- **Générateur de nombres pseudo-aléatoires sécurisé**

Cryptographically secure PRNG (CSPRNG)

- **RC4 (Rivest Cipher 4), 1987**

- Utilisé depuis les premières versions des protocoles SSL/TLS, WEP, WPA

- Expansion de la clé (*key schedule*) :
transformation de la clé en un état interne initial
(vecteur de 256 octets)
- Génération d'un **flux d'octets** pseudo-aléatoires par modifications
itératives (permutations) de l'état interne, dépendantes de la clé secrète.
- ⚠ Corrélations entre graine et flux



- **Salsa20** et **ChaCha**, 2008

- Protocoles TLS, IPsec, SSH et OpenVPN

- **Preuve de sécurité**

Démonstration qu'un schéma cryptographique respecte des propriétés de sécurités spécifiques, comme la confidentialité, l'intégrité et l'authenticité.

Il existe différents types de preuves cryptographiques, dont :

- Théorie de l'information
- Argument hybride 🖱️
- Preuve par jeu
- Réduction polynomiale
- *Zero knowledge proof*

- **Indistinguabilité calculatoire**

- Deux distributions de probabilités sont calculatoirement indistinguables, s'il n'existe pas d'algorithme **efficace** qui puisse les discerner de manière **significative**.

- \mathcal{A} : algorithme probabiliste terminant en temps polynomial (proportionnel à la longueur de la séquence en entrée x)
 - Aussi appelé distingueur (*distinguisher*) ou test statistique
 - $\mathcal{A}(x) = 1$: la sortie (booléen) de \mathcal{A} vaut 1 (ou **Vrai**), s'il détecte que l'entrée x a été échantillonnée à partir d'**une distribution plutôt que l'autre**
- Deux distributions D_n et E_n sont calculatoirement indistinguables si l'**avantage** $\delta(n)$ de l'adversaire est négligeable en fonction du « paramètre de sécurité » n :

$$\delta(n) = \left| \Pr_{x \leftarrow D_n} [\mathcal{A}(x) = 1] - \Pr_{x \leftarrow E_n} [\mathcal{A}(x) = 1] \right|$$

- Si la distribution est générée par un PRNG, n sera la longueur de la graine ;
Dans RSA, n sera la valeur du module

- **Indistinguabilité calculatoire**

- 🤔 La question posée par le calcul de l'avantage est : est-ce que A se comporte différemment selon qu'on lui donne une entrée échantillonnée à partir de D_n ou bien de E_n ? Si oui, cette différence est-elle négligeable (notation : $D_n \approx E_n$), c.-à-d. est que $\delta(n) \rightarrow 0$ quand $n \rightarrow \infty$?
- **Fonction négligeable** : une fonction $f(n)$ est considérée comme négligeable si, \forall polynôme positif $\text{poly}(n)$, \exists une valeur suffisamment grande de n tq $f(n) < \frac{1}{\text{poly}(n)}$
c.-à-d. si $f(n) \rightarrow 0$ plus vite que l'inverse de n'importe quel polynôme
- Indistinguabilité calculatoire est donc une relaxation de l'indistinguabilité statistique prenant en compte le fait que la puissance de calcul des algorithmes est limitée (notions de complexité)

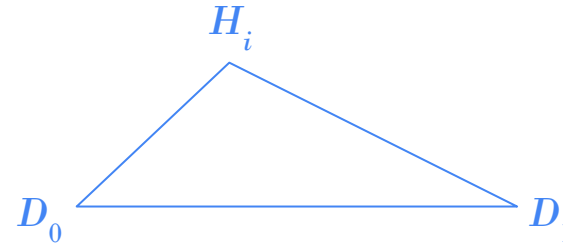
- **Argument hybride**

Pour montrer l'indistinguabilité calculatoire de deux distributions D_0 et D_1 , on définit une séquence de distributions hybrides tq $D_0 := H_0, H_1, \dots, H_n =: D_1$ qui permet de transformer graduellement la distribution D_0 en la distribution D_1 .

L'adversaire \mathcal{A} ne pourra lire que les n premières hybrides (entier au plus polynomial).

Si $H_i \approx H_{i+1}, \forall 0 \leq i < n$, alors $D_0 \approx D_1$

Note : conséquence de l'inégalité triangulaire



Dans un preuve de sécurité d'un chiffrement par flot, D_0 serait la sortie du PRNG et D_1 une séquence binaire parfaitement aléatoire.

- **Argument hybride**

La manière de définir les hybrides dépendra du PRNG dont on souhaite prouver la sécurité cryptographique.

À titre d'exemple, pour un générateur et sa graine $G(k)$:

- $H_0 : G(k)$
- $H_1 : G(k) \| G(G(k))$
- $H_2 : G(k) \| G(G(k)) \| G(G(G(k)))$
- ...
- $H_n : G(k) \| G(G(k)) \| \dots \| G^{(n)}(k)$
- $H_{n+1} : U \| U \| \dots \| U$

avec U représentant une séquence parfaitement aléatoire (uniforme)

2.5. Chiffrement de flux

- **Exemple** : soit $G : \{0, 1\}^n \rightarrow \{0, 1\}^l$ un PRNG sécurisé (avec $l = f(n)$, fonction polynomiale) ; soit un hybride $G'(k) = G(k) \| G(k)$ et soit la distribution uniforme $U(0^{2l}, 1^{2l})$
Question : la sortie de $G'(k)$ est-elle calculatoirement indistinguishable de U ?

Réponse : On peut définir un distingueur \mathcal{A} , qui détecte que la séquence $x \leftarrow G'(k)$ (et renvoie donc **Vrai**), en deux étapes :

1. Coupe x en 2 segments de même longueur tq $x_1 \| x_2 = x$
2. $\mathcal{A}(x) = \text{Vrai}$ si $x_1 = x_2$

Calcul de l'avantage de \mathcal{A} :

$$\delta(n) = | \mathbf{P}_{x \leftarrow G'(k)}[\mathcal{A}(x) = \text{Vrai}] - \mathbf{P}_{x \leftarrow U}[\mathcal{A}(x) = \text{Vrai}] |$$

Par la définition même de \mathcal{A} , on a : $\mathbf{P}_{x \leftarrow G'(k)}[\mathcal{A}(x) = \text{Vrai}] = 1$

$$\mathbf{P}_{x \leftarrow U}[\mathcal{A}(x) = \text{Vrai}] = ?$$

2.5. Chiffrement de flux

- **Intuition** avec les plus petites valeurs possibles : 2 bits

Un générateur aléatoire peut générer $x \in \{00, 01, 10, 11\}$

Mais $G'(k)$ ne peut générer que $x \in \{00, 11\}$

$$P_{x \leftarrow U}[A(x) = \text{Vrai}] = 2/4 = 1/2$$

$$\delta(n) = |P_{x \leftarrow G'(k)}[A(x) = \text{Vrai}] - P_{x \leftarrow U}[A(x) = \text{Vrai}]| = 1 - 1/2 = 1/2 \text{ non-négligeable}$$

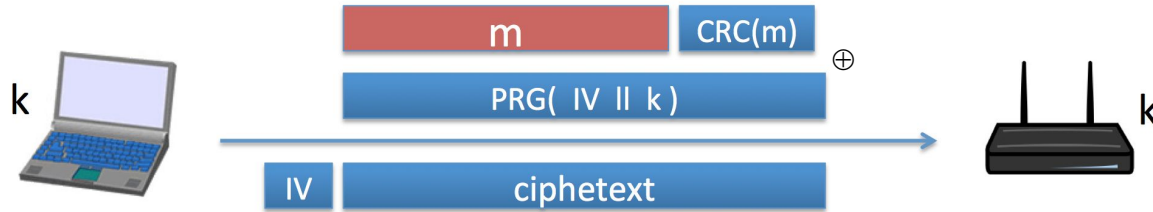
- Pour toute longueur l de x

$$P_{x \leftarrow U}[A(x) = \text{Vrai}] = 2^l / 2^{2l} = 2^{-l}$$

$$\delta(n) = |1 - 2^{-l}| = |1 - 2^{-f(n)}| > \frac{1}{\text{poly}(n)} \text{ non-négligeable ; } G'(k) \text{ n'est pas sécurisé}$$

- **Limitations**

- **802.11b WEP:**



La même clé k était réutilisée ; seul le IV changeait à chaque message, mais était trop court (24 bits) → Utiliser WPA2 ou WPA3 pour sécuriser les réseaux WiFi

- Chiffrement de disque :
 - Des erreurs dans le chiffré peuvent empêcher le déchiffrement
 - Beaucoup de données : la périodicité de la clé de flux ou sa réutilisation cause l'apparition de motifs répétés dans le chiffré, détectables en cryptanalyse
→ **Utiliser un chiffrement par bloc**

- **Confusion et diffusion**

Deux propriétés des opérations utilisées dans les méthodes de chiffrement, afin de contrecarrer les méthodes de cryptanalyse (Claude Shannon, 1949).

- **Confusion** : signifie que chaque bit du **texte chiffré** devrait dépendre de plusieurs parties de la **clé**, pour cacher la relation entre les deux. La confusion rend difficile la découverte de la clé à partir du chiffré : si un seul bit dans la clé est changé, le calcul de la majeure partie du chiffré sera affecté.
- **Diffusion** : signifie que si l'on change un seul bit du **texte clair** en entrée, alors environ la moitié du **texte chiffré** en sortie devrait changer de façon imprédictible et vice versa (voir **effet avalanche**). La diffusion aide à ce que la structure statistique du texte clair (les *patterns*) ne soit pas préservée dans le chiffré. En effet, des relations entre les bits en entrée et en sortie pourraient être utiles pour le cryptanalyste.

2.6. Chiffrement par bloc

- **Chiffrement par bloc**

Un chiffrement par bloc découpe le message clair m en blocs d'une certaine taille (p. ex. 128 bits) et produit en sortie le message chiffré c , constitué d'autant de blocs chiffrés, chacun de cette même taille.

💡 Dans un chiffrement continu, le caractère $c[i]$ résulte du chiffrement du caractère $m[i]$. Cela équivaut à une **taille de bloc de 1**.

- **Réseau de substitution-permutation (SP)**

- S-Box (*substitution box*)

Présentes dans beaucoup de chiffrements par bloc, elles sont utilisées pour effectuer des opérations de substitutions au sein de chaque bloc durant le processus de chiffrement. Ces substitutions ajoutent de la **confusion**.

Ce sont souvent des tables de correspondances (*Lookup Table*, LUT).

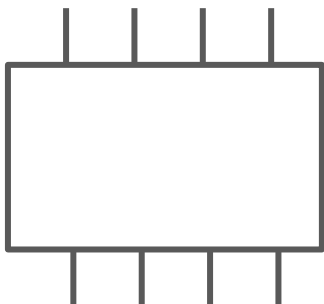


Illustration d'une S-Box : 4 bits en entrée et sortie.

La table de substitution comporte donc $2^4 = 16$ règles, pour des valeurs décimales de 0 à 15 : $0 \rightarrow 6$; $1 \rightarrow 13$; $2 \rightarrow 1$; ...

La taille des S-Boxes AES est de 8 bits. Dans DES, elle était de **6 bits en entrée, 4 bits en sortie**.

Note : les substitutions réciproques doivent être évitées.

- **Réseau de substitution-permutation (SP)**

- P-Box (*permutation box*)

Utilisées pour permuter (réarranger) les bits des données d'entrée, de manière systématique, selon un motif prédéfini. La P-Box introduit de la **diffusion**.

P-Box et S-Box sont déterministes et réversibles, par un adversaire qui en connaît possiblement les règles (voir principe de Kerckhoffs).

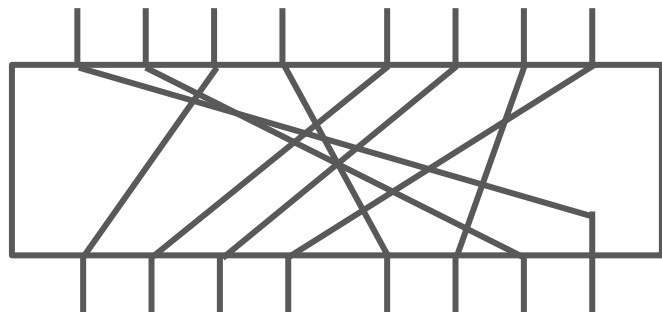
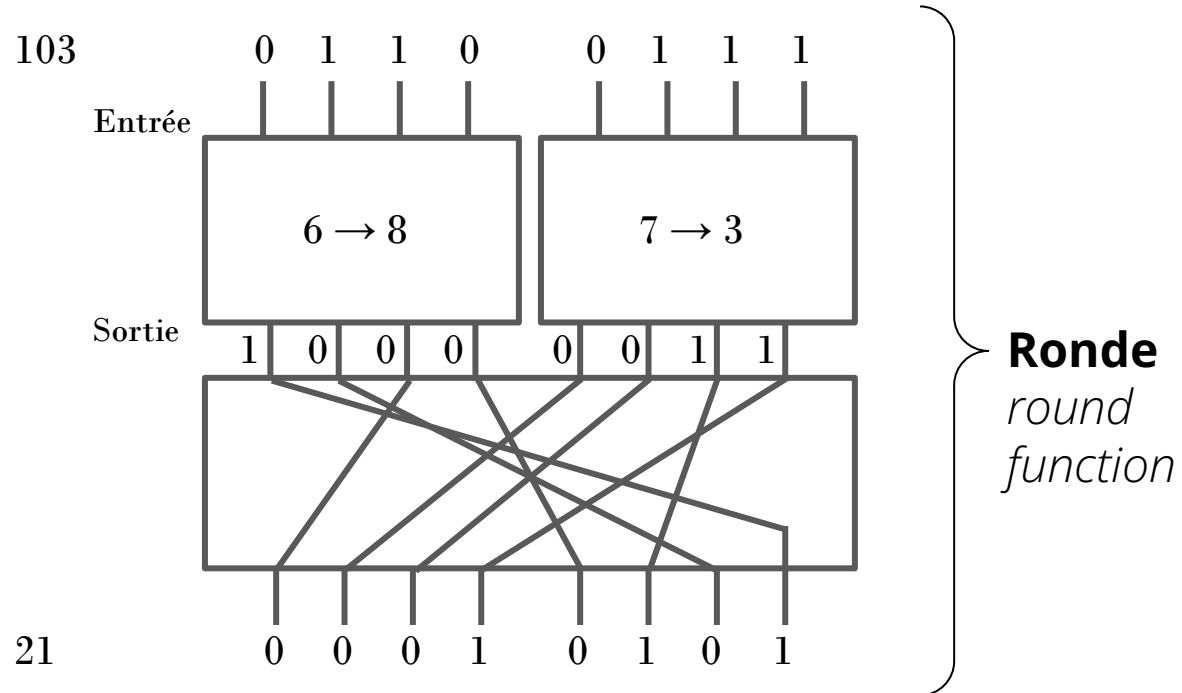


Illustration d'une P-Box : 8 bits I/O

- Réseau de substitution-permutation (SP)

- Exemple



2.6. Chiffrement par bloc

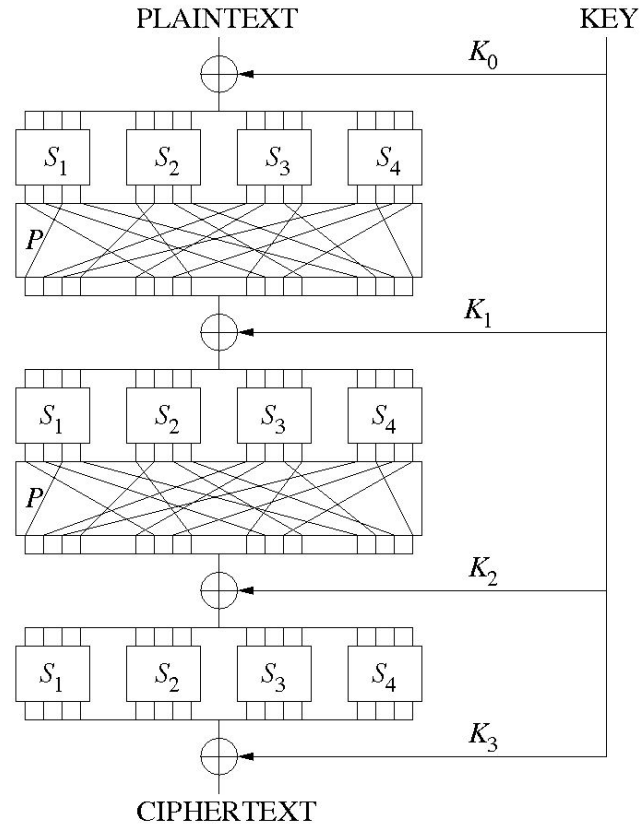
- **Réseau de substitution-permutation (SP)**

Plusieurs rondes doivent être effectués : leur nombre est un compromis en vitesse et sécurité

On introduit l'utilisation d'un clé secrète, afin que, sans elle, **on ne puisse plus inverser la fonction** combinant S-Box et P-Box (« boîtes blanches »).

Préparation des clés (*key schedule*) : on crée des sous-clés à partir de la clé principale (« expansion » et division en clés de rondes).

Chaque ronde se termine par un **XOR** avec une sous-clé (ci-contre : 3 rondes)



2.6. Chiffrement par bloc

• Rijndael

Réseau SP de Rijmen et Daemen, 1998

- Bloc de données arrangé en grille
P. ex. $4 \times 4 = 16$ octets = 128 bits

- **Substitutions** (S-Box) :

Table de correspondance → rapide

⚠ Pas de point fixe

Pas de point fixe opposé (p. ex. $1010 \Leftrightarrow 0101$)

- **Permutations** :

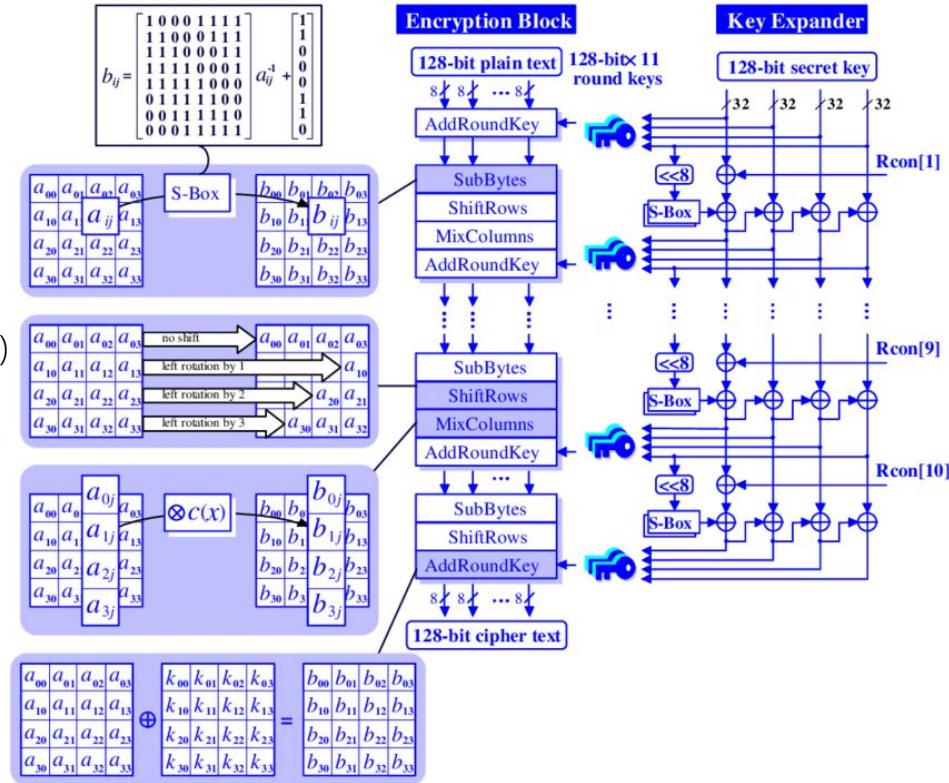
Décalages des rangs

Mélange des colonnes

- Multiplication par une matrice
→ Transformations linéaires

- **Addition de la clé de ronde**

- Dernière ronde : permutation inutile



- **Rijndael**

- Algorithme choisi par le NIST pour l'*Advanced Encryption Standard* (AES) pour remplacer (Triple) DES : devait être aussi sécurisé, mais beaucoup plus rapide
- L'AES fixe :
 - la longueur de bloc à 128 bits
 - la longueur des clés à 128, 192, ou 256 bits
 - et le nombre de rondes associé : 10, 12 et 14, respectivement
- Chaque étape peut être effectuée dans le **sens inverse** → **déchiffrement**
- https://formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng-html5.html
- **Arithmétique modulaire**

Toutes les opérations (addition, soustraction, multiplication, inversion) se font dans un « corps fini » (ou corps de Galois) de 2^8 éléments, soit 1 octet, les valeurs allant de 0000 0000 à 1111 1111. Les 256 éléments sont donc représentés par un polynôme irréductible de coefficients binaires $m(x)=x^8 + x^4 + x^3 + x + 1$ et les opérations se font modulo ce polynôme