Synthèse Théorique
Principes de cryptographie
Partie 1

Tout jusqu'à la distribution quantique des clés
Deuxième Bloc
Sécurité des systèmes
Année académique 2020-2021

Rédigé par
Sénéchal Julien

21 Décembre 2020

1 Histoire

1.1 Scytale

- Le message vient s'enrouler au tour de la Scytale
- Clé : l'épaisseur de la scytale

1.2 Atbash

- Chiffrement par substitution inversée
- Clé : Nombre de lettres et sens du décalage

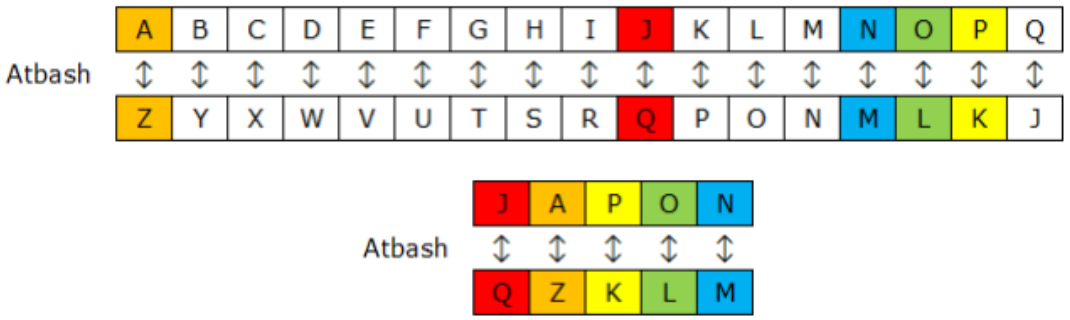

Figure 1 – Atbash

1.3 Chiffre de César

- Consiste à décaler les lettres de l'alphabet d'un nombre n
- Clé : Nombre de lettres de décalage
- Peu sûr car peu de configuration possible (25 possibilité)

1.4 Cryptanalyse : al-Kindi

- Analayse la fréquence des lettres du texte chiffré et le compare avec la moyenne des lettres utilisé dans la langue
- Utile pour une substitution monoalphabétique

1.5 Le cadran d'Alberti

- Chiffrement polyalphabétique
- 2 cadran, l'un avec les lettres dans l'ordre et l'autre dans le désordre
- On aligne les 2 "A" et toutes les 4 lettres on tourne d'une lettre le petit disque
- Suffit de posséder le cadran pour le décrypter

1.6 Vigenère

- Intersection entre la lettre en clair et celle de la clé
- Simple et sûr si la clé sécurisée
- Analyse de fréquence inutile
- Substitution polyalphabétique par bloc

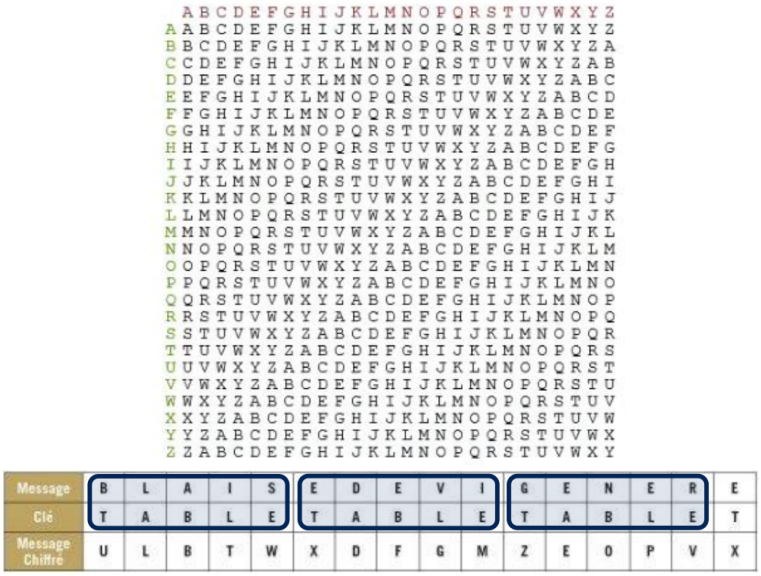

FIGURE 2 – Vigenère

1.7 Chiffre de Playfair

- Chiffrement parblocs de 2 lettres
- Clé : Phrase du tableau de 5 x 5
- Très dur a casser par analyses fréquentielles

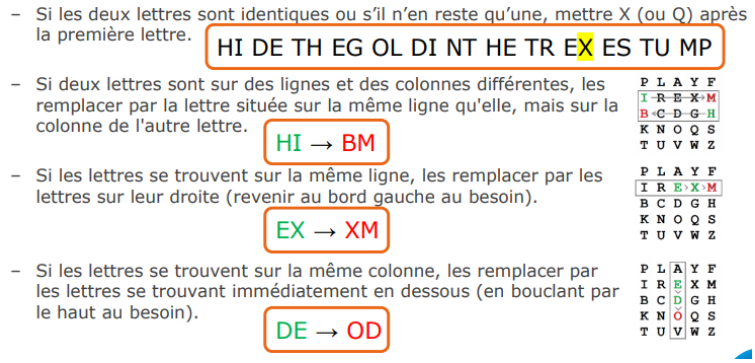

FIGURE 3 – Playfair

1.8 Principe de Kerckoffs

- Les systèmes doivent être totalement publics
- Le message est sûr tant que la clé est secrète
- Toute attaque contre le cryptosystème doit être envisagée en considérant que l'attaquant connaît tous les détails de conception du système cible

1.9 Chiffre de Vernam

- Chiffre parfaitement sûr (que même avec le message chiffré et une puissance de calcul infinie, il est impossible de retrouver le message en clair)
- 3 impératifs
  - ➤ Aussi longue que le texte à chiffrer

    - ➤ Parfaitement aléatoire
    - ➤ Utilisée pour chiffrer un seul message puis est immédiatement détruite
- Inconvénient
    - ➤ clé très longues
    - ➤ parfaite synchro des clés
    - ➤ échange des clés doit être sécurisé
    - ➤ parfaitement aléatoire > très difficile

1.10 Enigma

- Basé sur le chiffre de vigenère de longueur 26 (26 nombre de rotors)

2 Vocabulaire

- Cryptographie
    - ➤ Science consistant à écrire l'information pour la rendre inintelligiblle à ceux qui ne possèdent pas les capacités de la déchiffrer
- Chiffrement
    - ➤ Opération par laquelle on chiffre le message, c'est une opération de codage
    - ➤ Chiffrer = cryptographier $\neq$ crypter
- Déchiffrer
    - ➤ Se fait avec la clé de chiffrement
- Décrypter
    - ➤ Lorsqu'on a pas la clé
- Cryptanalyse
    - ➤ Ensemble des moyens permettant d'analyser une information chiffrée afin de la décrypter.

3 Principes de base

- Les algorithme publics sont plus robustes
- La taille de la clé est primordiale
- Le secret du message chiffré est basé sur le secret de la clé et non de l'algorithme

4 Outils de base

- XOR
    - ➤ Méthode fondamentale
    - ➤ Très simple
    - ➤ Pas sécurisé
    - ➤ Principe : Vrai si une seule des 2 entrées est vraie en binaire
- Substitution
    - ➤ Remplacer chaque lettre du texte en clair par une autre lettre
    - ➤ Le destinataire fait l'inverse
    - ➤ Substitution arbitraire : Mise en place d'une table de conversion, chaque lettre remplace une autre de façon arbitraire
    - ➤ Substitution par rotation : Code de césar avec un nombre n de rotation
- Hybride
    - ➤ Une des techniques ci-dessus $\rightarrow$ Facilement décryptable
    - ➤ Les différents outils de chiffrement sont combinés pour obtenir un chiffrement plus robuste

5 Chiffrement symétrique

- Base
  - ➤ Cryptographie a clé secrète
  - ➤ Même clé pour le chiffrement et le déchiffrement
- Longeur de la clé
  - ➤ L'attaque la plus simple pour récupérer la clé est l'attaque par Brute Force → Pour contrer : augmenter la taille de la clé
- Echange de la clé
  - ➤ Il faut un canal de transmission de clé très sécurisé
- Distribution des clés
  - ➤ Chaque couple d'interlocuteurs doit posséder sa clé secrète
  - ➤ n personnes nécessitent $\frac{n*(n-1)}{2}$ clés
- Avantages
  - ➤ Nécessite des clés de taille relativement faible (128 bits)
  - ➤ Consomme peu de ressources
  - ➤ Peut chiffré en temps réel ou différé
- Exemples
  - ➤ DES
  - ➤ 3DES
  - ➤ RC2, RC4, RC5
  - ➤ IDEA
  - ➤ Blowfish
  - ➤ AES

6 Chiffrement asymétrique

- Analogie
  1. Alice envoie de la clé publique c'est comme si on envoyait une valise avec un cadenas ouvert
  2. Bob mets son message dans la valise et la referme avec le cadenas (donc chiffre le message avec sa clé publique)
  3. Alice ouvre la valise avec sa clé secrète qui est la clé du cadenas
- Lien entre la clé privée et la clé publique
  - ➤ Les 2 clés sont liées par des problèmes mathématiques extrêmement difficiles à résoudre
  - ➤ Des fonctions trappes sont utilisées pour cela, elles ont la particularité d'être facile à calculer dans un sens mais presque impossible dans le sens inverse. Le sens moyen de faire le calcul inverse est de connaître la trappe
  - ➤ Exemple :
    - — Il est facile de faire $3^5 = 243$ mais beaucoup moins simple de faire le calcul inverse qui est $log_x y = 243$ car on ne connaît pas la paire de xy
  - ➤ Les algorithmes réels utilisent des nombres premiers, ils peuvent avoir plusieurs centaines de chiffres.
- Congruence

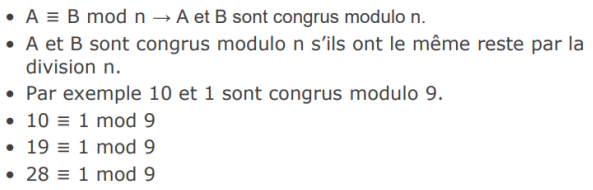

FIGURE 4 – Congruence

- Exemples :
  - RSA (Basé sur la factorisation en nombres premiers)
  - Diffie-Henman (Basé sur le calcul des logarithme discret)

7 Fonction de hachage

- Transforme la donnée initiale en une donnée numérique de taille fixée de faible longueur appellé hash
- Le hash est l'empreinte de la donnée initiale
- Généralement entre 128 et 256 bits
- Propriété :
  - La longueur de l'empreinte est toujours la même, quelle que soit la longueur du message en entrée.
  - Cette empreinte doit être unique. Deux messages, même très proches ont une empreinte différente.
  - La fonction de hachage doit être une fonction à sens unique afin qu'il ne soit pas possible, à partir de la signature, de remonter au message initial.
- Concepts
  - Si un bit de donnée en entrée est modifié, chaque bit de sortie a 50% de chance de changer
  - Le but du hash est de s'assurer de l'intégrité de la donnée
  - Font partie des signatures numériques et sont utilisé pour stocker les mdp
- Exemples :
  - SHA-1 SHA-256 SHA-384 SHA-512 (Secure Hash Algorithm)
  - HAVAL
  - RIPE-MD
  - MD2, MD4, MD5, MD6

8 Signature Numérique

- Principe
  - L'émetteur va envoyer un document et va envoyer son empreinte qui elle sera chiffrée par sa clé privée
  - Le destinataire va alors déchiffrer l'empreinte avec la clé publique (comme il arrive a le déchiffrer, l'emprunte vient bien du bon emetteur)
  - Le destinataire va ensuite recalculer l'emprunte du document et le comparer avec l'emprunte reçue

9 Clé de session

- Optimisation par clé de session
  - Chiffrement asymétrique lent (clé d'environ 2048 bits)
  - Chiffrement symétrique rapide mais problème de distribution des clés
  - Solution : On combine les deux
    1. Pour chiffrer un message de grande taille, on le chiffre avec une clé de session symétrique à usage unique
    2. La clé de session est chiffrée avec la clé publique du destinataire
    3. On envoie le message et la clé
    4. Le destinataire déchiffre la clé de session grâce à sa clé privée
    5. Il peut maintenant déchiffrer le message

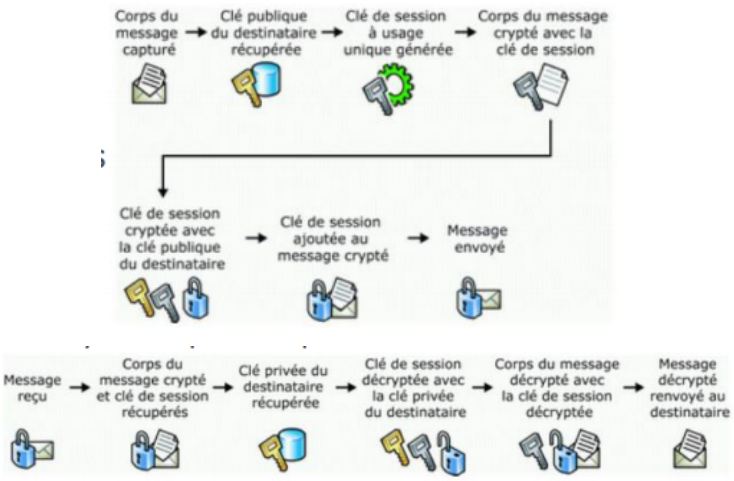

FIGURE 5 – Echange de la clé de session

10 Signature numérique et Clé de session

- L'expediteur
  1. On hash le message
  2. On chiffre la hash avec la clé privée de l'expéditeur
  3. On ajoute le hash chiffré au message et on chiffre le tout avec la clé de session
  4. On chiffre la clé de session avec la clé publique du destinataire et on l'envoie avec le pack "hash + message" chiffré
- Le destinataire
  1. La clé de session est déchiffrée grâce à la clé privée du destinataire
  2. On déchiffre le pack "hash + message" avec cette clé de session
  3. On calcule le hash du message
  4. On déchiffre le hash reçu grâce à la clé publique de l'expéditeur
  5. On compare les 2 hash et si ils correspondent, tout est bon

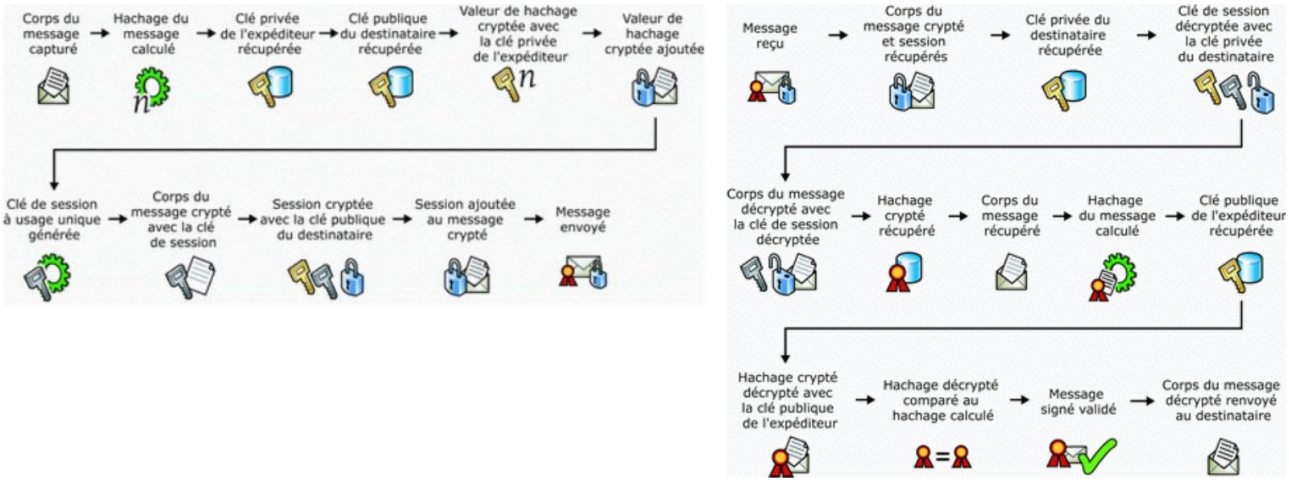

FIGURE 6 – Clé de session + signature

11 Avantages/Inconvénient

- Symétrique
  - ➤ Avantages
    - — + facile
    - — + rapide
    - — - de puissance nécessité
    - — Empêche les attaques généralisées étant donné que les clés secrètes sont différentes pour chaque communications
    - — Pas de lien entre les données et la clé
    - — Sans clé, pas moyen de décrypter
  - ➤ Inconvénients
    - — Manque de sécurité à l'échange des clés
    - — Difficile de gérer et de sécuriser un grand nombre de clés partagées
    - — Ne fournit aucune assurance de l'origine et de l'authenticité d'un message
    - — La même clé est utilisé par les 2
    - — Vulnérables aux attaques par dictionnaire et par force brute
- Asymétrique
  - ➤ Avantages
    - — Pratique pour la distribution de clé symétrique
    - — Sécurité renforcée car pas besoin de partager les clés privées
    - — Fourni des signatures numériques
  - ➤ Inconvénients
    - — Lent
    - — + grande puissance de calcul
    - — Si la clé privée est volée, l'intégralité des messages peuvent être déchiffré
    - — Les messages reçus sont perdu si la clé privée est perdue
    - — Vulnérable au MITM

12 Man in the middle

Je connais très bien donc pas de synthèse

13 Infrastructure à clés publiques

- Objectifs
  - ➤ Nécessaire pour mettre en place le chiffrement asymétrique
  - ➤ Permet de résourdre le problème d'association entre une entité et une clé publiqué
  - ➤ Désigné par IGC ou PKI (Infrastructure de Gestion de clés ou Public Key Infrastructure)
  - ➤ Fonctions des PKI :
    - — la génération de couple unique de clés (privée et publique)
    - — la création et la gestion de certificats numériques
    - — la diffusion des clés publiques aux ressources qui la solliciteraient
    - — la certification des clés publiques
- Certificats Electroniques
  - ➤ Attestent de l'identité numérique des détenteurs de clés publiques

    - 4 objectifs (ceux de la sécu)
        - confidentialité
        - authenticité
        - intégrité
        - non-répudiation
    - Le certificat est signé avec la clé privé de l'organisme de certification (chiffrement de l'empreinte du certificat avec la clé privée de l'organisme de certification)
    - Pour valider le certificat :
        1. Obtenir la clé publique de l'organisme de certification
        2. Déchiffrer la signature à l'aide de cette clé
        3. Calculer l'empreinte du certificat
        4. Comparer l'empreinte calculée et celle reçue (se trouve à la fin de la signature)
        5. Vérifier que la période de validité du certificat est correcte
    - Déjoue le MITM
    - Certificats auto-signé
        - Certificats signé avec la clé privée de l'expéditeur et non celle de l'Infrastructure
- Principaux types de certificats :
    - Certificats de messagerie
    - Authentication IPSec pour un accès distant par VPN
    - Authentication Internet pour l'HTTPS
    - Chiffrement des données avec EFS
    - Signature logiciel
- Composant d'une PKI
    - CA (Certificate Authority) : émet et révoque des certificats
    - RA (Registration Authority) : vérifie l'identité pour le CA
    - VA (Validation Authority) : détient les certificat accompagné de leur clé publique
    - End user : Demande, utilise et gère des certificat
    - Digital Certificates : identifie une personne lors de transactions en ligne
- Résumé :
    1. Utilisateur demande a RA un certificat
    2. RA vérifie son identité et demande à CA de lui donner le certificat de clé publique
    3. CA donne donne le certificat avec la clé publique de l'utilisateur a Utilisateur
    4. Utilisateur envoie les informations à VA
    5. Quand Utilisateur effectue une action, il signe le mesage avec le certificat de clé publique
    6. Utilisateur envoie son certificat pour prouver son identité au destinataire
    7. Le destinataire vérifie que le certificat est valide
    8. Le VA compare le certificat de clé publique de l'utilisateur avec celui qu'il détient et détermine le résultat (qu'il soit valide ou non).

14 Algorithmes de cryptographie

14.1 Chiffrement par flot et par bloc

- 2 types de chiffrement symetrique
    - Continu (par flot) : agit sur un bit à la fois du message en clair
    - Par bloc : opère sur le message en clair par groupe de bits (des blocs)

- Chiffrement par flot
  - ➤ Pas besoin de lire ni de connaître la longueur du message pour le chiffrer
  - ➤ Tente d'imiter le chiffre de Vernam
  - ➤ Généralement combiné par opération XOR avec flux de bits pseudo-aléatoire
  - ➤ Une clé de chiffrement ne doit jamais être utilisée plus de 2 fois
  - ➤ Technique pour ne pas échanger de nouvelles clés en continu :
    - — Synchronisation d'algorithme via horloge
    - — Vecteur d'initialisation renouvelé et échangé en clair (à ajouter à la clé)
  - ➤ Utilité :
    - — Téléphonie mobile
    - — Bluetooth
  - ➤ Très bonne performances
  - ➤ Problèmes de sécurité
  - ➤ Exemples :
    - — RC-4 (mal utilisé par le WEP du WiFi)
    - — A5/1 (Utilisé dans la téléphonie mobile GSM)
- Chiffrement par bloc :
  - ➤ Le message est découpé en blocs de n blocs de bit, tous de même taille, ensuite chaque blocs est chiffré
    - — Si la longueur du message n'est pas un multiple de la longueur du bloc, on utilise le padding (bourrage) pour compléter le dernier bloc
  - ➤ Existe plusieurs technique de chiffrement par bloc :
    - — EBC (Electronic CodeBook) :
      - — Chaque bloc est chiffré séparémment les uns après les autres
      - — Non recommendé car si même contenu alors même chiffrement
      - — On obtient un dictionnaire de codes avec les correspondances entre le clair et le chiffré

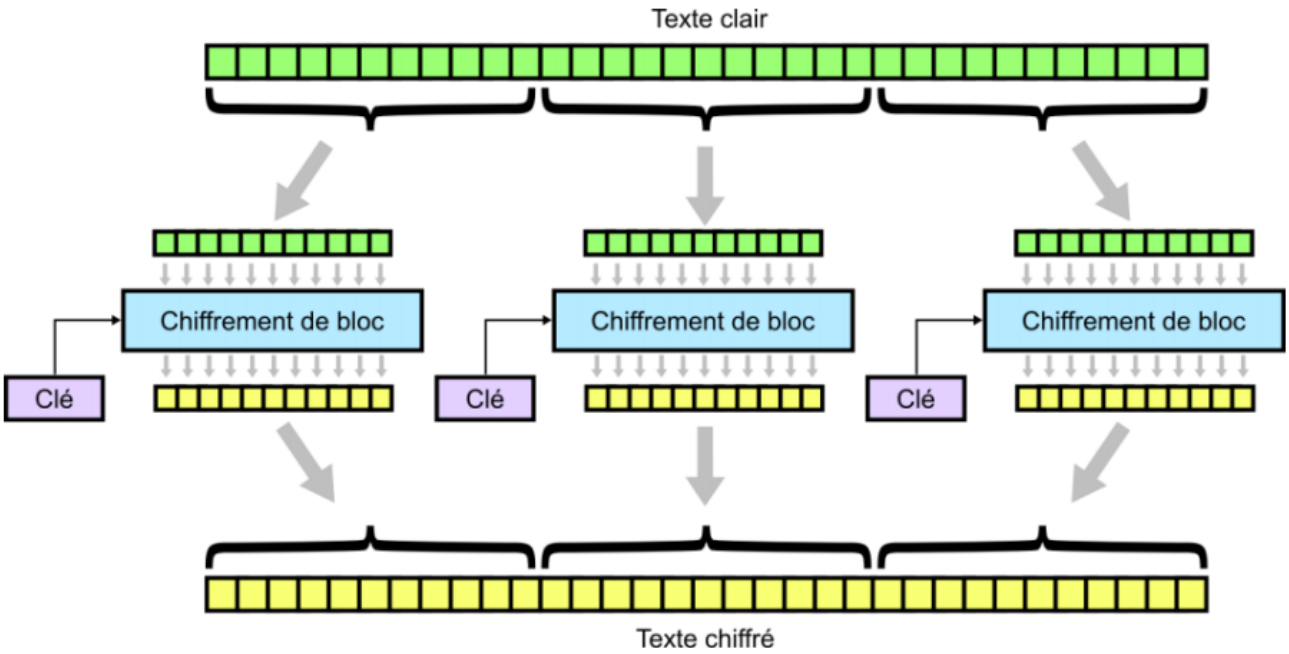

FIGURE 7 – EBC

— CBC (Cipher Block Chaining)
  — "OU exclusif" sur chaque block avec le chiffrement du bloc précent
  — Pour rendre chaque message unique, un vecteur d'initialisation (IV) est utilisé

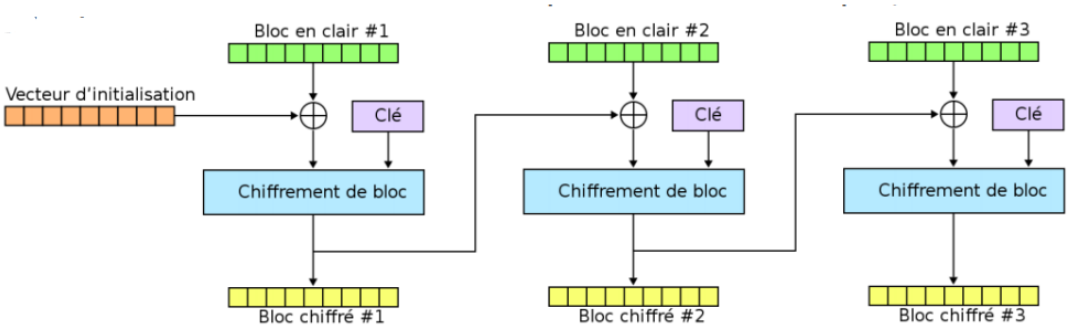

FIGURE 8 – CBC

— CFB (Cipher FeedBack) et OFB (Output FeedBack)
  — Utilisé comme un générateur pseudo-aléatoire de clés en essayant de simuler un chiffrement par masque jetable
  — Exemple : DES, AES

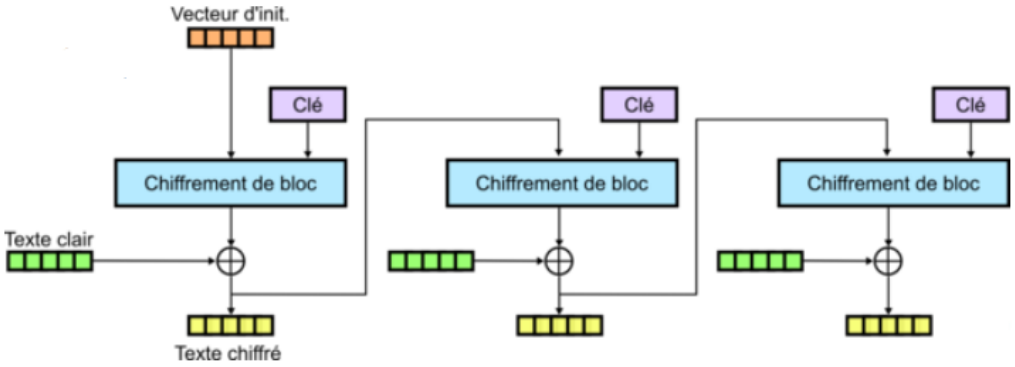

FIGURE 9 – CFB et OFB

14.2 DES

- Description de la clé
  - ➤ Clé = chaîne de 64 bits
  - ➤ Seuls 56 bits servent réellement à définir la clé
  - ➤ Les bits 8,16,24,32,40,48,56,64 = bits de parité
  - ➤ 2^{56} clés possibles = 72 millions de milliard de possibilités
- DES cassé en 3 semaines en 1997
- Fonctionnement :
  - ➤ 16 sous clés sont créées à partir de la clé de 56 bits

- ➤ Utilise des combinaisons, substitutions, permutations (entre texte et clé)
- ➤ Chaque bloc de 64 bits du texte est calculé une permutation et on divise les 64 bits en 32
- ➤ On applique 16 tours, chacun à l'aide d'une sous-clé et d'un même schéma de substitutions et de permutations
- ➤ On regroupe les 2 blocs de 32 bits et on permute
- ➤ Le déchiffrement ce fait dans l'ordre inverse

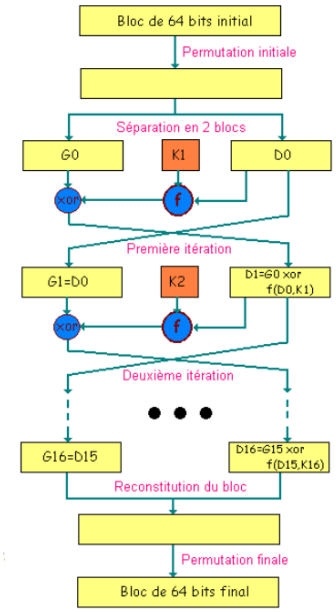

Figure 10 – DES

14.3 3DES

- Etant donné que DES ne pouvais plus être utilisé, 3DES est créé en solution provisoire
- Fonctionnement :
  - ➤ 2 clés de 56 bits
  - ➤ 3 chiffrement DES en chaîne
  - ➤ Augmente significativement la sécurité de DES (demande plus de ressources)
- Variante :

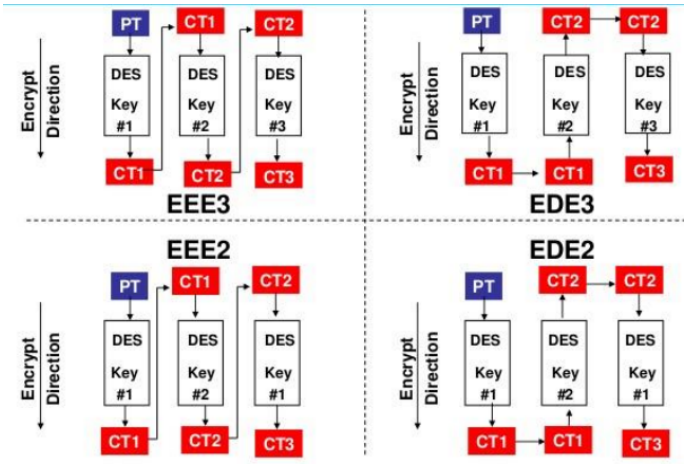

Figure 11 – Variante de 3DES - DES-E...

14.4 AES - Advanced Encryption Standard

- Belge
- Fonctionnement
  - ➤ Chiffrement par bloc
  - ➤ clés de : 128, 192, 256
  - ➤ Blocs de 128 bits
  - ➤ Les blocs sont encodés en plusieurs cycles consécutifs
  - ➤ A chaque cycle : permutation, substitution, XOR, ...
- Utilité
  - ➤ WPA2
  - ➤ SSH
  - ➤ IPSec
  - ➤ Chiffrement des archives compressés

14.5 RC

- Inventé par l'inventeur du MD5
- Employé en raison de leur vitesse et parce que la longeur de la clé est variable
- RC2
  - ➤ Conçu pour remplacer le DES
  - ➤ Chiffrement par bloc de taille de clé variable
- RC4
  - ➤ Chiffrement par flux le plus utilisé au monde
  - ➤ Utilisé pour le chiffrement de fichiers et pour les communications sécurisées (ex : SSL)
  - ➤ Considéré comme sécurisé bien qu'il puisse être implémenté de manière non sécurisé (WEP)

14.6 Diffie-Hellman

- Utilisé pour échanger des clés en toute sécurité
- Utiilité :
  - ➤ Algorithme mathématique qui permet de générer un secret partagé sur 2 systèmes sans avoir à le communiquer
  - ➤ Pas vraiment de clé symétrique échangée entre l'expéditeur et le destinataire
  - ➤ VPN IPsec, SSL TLS, quand des données SSH sont échangées
- Analogie :

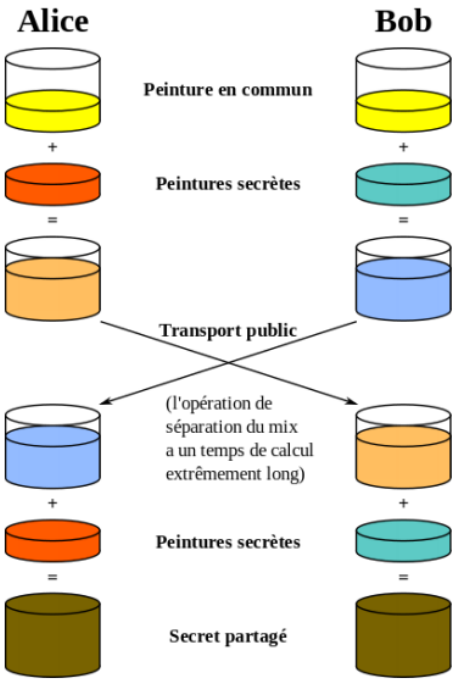

FIGURE 12 – Analogie Diffie-Hellman

- Fonctionnement :
  - ➤ Utilise les modulos
  - ➤ 38 modulo 7
  - ➤ 1ère étape :
    - — Alice et Bob choisissent ensemble 2 nombres (p : un nombre premier, g : tel quel 1<g<p (generator))
    - — Nombre public
  - ➤ 2ème étape :
    - — Alice génère aléatoirement un grand nombre A
    - — pareil pour Bob avec un nombre B
    - — A et B restent secret
  - ➤ 3ème étape :
    - — Alice calcule $P_A = g^A \ mod \ p$ et transmet le résultat à Bob
    - — Bob calcule $P_B = g^B \ mod \ p$ et transmet le résultat à Alice
  - ➤ 4ème étape :
    - — La clé secrète symétrique est $k = g^{AB} \ mod \ p$
    - — Alice peut calculer k à partir de A et $P_B(g^B \ mod \ p)$
    - — Bob peut calculer k à partir de B et $P_A(g^A \ mod \ p)$

Alice			
p=23	g=5	A=6	$P_A = 8$
			$P_B = 19$
$k = P_B^A \bmod p = 19^6 \bmod 23 = 2$ La clé secrète est donc **2**			

Bob			
p=23	g=5	B=15	$P_B = 19$
			$P_A = 8$
$k = P_A^B \bmod p = 8^{15} \bmod 23 = 2$ La clé secrète est donc **2**			

FIGURE 13 – Exemple Diffie-Hellman

- Secret
  - Si Eve écoute la conversation, elle ne pourra pas trouver k
  - Nombres utilisés sont en réalité d'environ 1024 bits (309 chiffres) donc complexe
- Utilisation
  - Exige la simultanéité des actions d'Alice et de Bob
  - Protocol surplanté par des méthodes de type RSA pour lesquels on met à disposition une clé publique
  - Utilisé pour l'appariement d'objets Bluetooth

14.7 RSA

- Très démocratisé et présent presque partout
- Système à clé publique
- Fonctionnement :
  - P et Q étant 2 nombres premiers aléatoire
  - $N = P.Q$ = Modulus
  - $\phi = (P-1)(Q-1)$
  - E aléatoire tel que $1 < E < \phi$ et que E et ϕ soient premier entre-eux (E n'est pas forcément premier mais doit être impair)
  - D calculé tel que $(D.E-1)$ divisible par ϕ
    - Il faut trouver un entier X tel que D soit entier et que :
    - $D = \frac{(X.\phi)+1}{E}$
  - E = exposant public ((N, E) = clé publique)
  - D = exposant privé ((N, D) = clé privée)
  - Taille de P & Q > 1000 bits (détruit après génération des clés)
  - Chiffrement :
    - $C = T^E \ mod \ N$
    - T = texte clair
    - C = texte chiffré
  - Déchiffrement :
    - $T = C^D \ mod \ N$
- Exemples avec petites valeurs :
  - P = 11
  - Q = 3
  - N = 33 (P.Q)
  - $\phi = (P-1).(Q-1) = 20$
  - E par exemple = 3 (3 < 20, 3 est premier et ne divise pas 20, E et ϕ sont premier entre eux)
  - D calulé :
    - $D.3-1$ doit être divisible par 20
    - $D = \frac{(X.20+1)}{3}$
    - Avec X = 1 $\rightarrow$ D = 7
  - Chiffrer 13 : $C = 13^3 \ mod \ 33 = 19$
  - Pour déchiffrer 19 : $T = 19^7 \ mod \ 33 = 13$

14.8 MD5 et SHA

- MD5
  - ➤ Séquence complexe d'opération binaires simples (XOR, rotations)
  - ➤ Empreinte de 128 bits
  - ➤ Déprécié
- SHA
  - ➤ Similaire à MD5
  - ➤ 3 générations : sha-1, sha-2 et sha-3
- Sécurité :
  - ➤ Failles de sécurité dans SHA-1 et MD5
  - ➤ Il faut utiliser SHA-256 ou version ultérieure

14.9 SSL et TLS

- Chiffrement asymétrique pour un échange de clé symétrique + certificats
- Fonctionnement :
  1. Le navigateur envoie une requête pour se connecter grâce au SSL ou TLS
  2. Le serveur envoie ses certificats qui contiennent sa clé publique
  3. Le client vérifie le certificat et si il est bon, il chiffre une clé de session avec la clé publique du serveur et lui envoie
  4. Le serveur déchiffre la clé de session avec sa clé privée
  5. Ils communiquent ensuite grâce au chiffrement de cette clé de session