Sécurité et Réseaux
Licence 3 Informatique

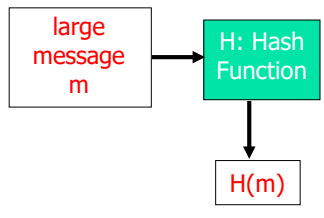
Cours 4: Introduction

Osman SALEM
Maître de conférences - HDR
osman.salem@parisdescartes.fr

MATHÉMATIQUES ET INFORMATIQUE
Sciences
Université de Paris

Fonctions de hachage

- Aussi appelée fonction de condensation
  - à partir d'un texte de longueur quelconque, calculer une chaîne de taille inférieure et **fixe**
  - appelé condensé ou empreinte (message digest ou hash en anglais)



Fonctions de hachage

- Utilisée seule, elle permet de vérifier *l'intégrité* d'un message
- Appliqué à un texte et à une clé privé, elle permet le calcul d'un MAC (Message Authentification Code), pour assurer :
  - *Intégrité* des données: garantit que le message n'est pas altéré
  - *Authentification* de la source
- Associé à un chiffrement asymétrique, elle permet le calcul de signatures, pour assurer :
  - *Intégrité* des données
  - *Authentification* de la source
  - *Non-répudiation* de la source

Fonction de hachage (fin)

- Une fonction de hachage doit être :
  - à *sens unique*, c'est à dire qu'il doit être impossiblede trouver m à partir de H(m)
  - *sans collisions ou presque*, impossibilité de trouver deux messages distincts tel que H(m)=H(m')
  - La moindre modification du message entraîne la modification de l'empreinte
- Exemples :
  - MD5 (Message Digest 5 - RFC 1321) : calcul une empreinte de 128 bits
  - SHA-1 (Secure Hash Algorithm 1 - NIST1994) : plus sûr que MD5 - empreinte de 160 bits

Fonction de hachage

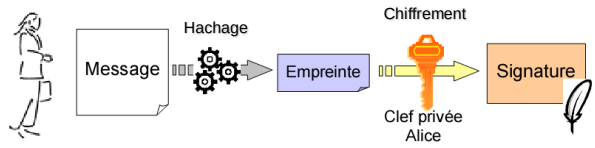
- Fonction mathématique qui
  - à un ensemble de nombres en entrée, fait correspondre un ensemble de nombres de cardinal plus petit en sortie
  - La modification d'un élément en entrée engendre une modification de sa fonction de hachage en sortie

M	h(M)
`remarquez la fin de cette ligne,`	`4b:2c:65:c0:e8:ee:95:5f:eb:05:9f:3c:6d:2f:2f :0f:9a:26:00:b7`
`remarquez la fin de cette ligne!`	`9e:e9:22:99:11:7f:41:23:7c:ce:38:3a:d8:05:18 :0c:4a:fc:ab:4c`
`??`	`75:cc:4b:e0:a9:7c:76:34:78:58:bf:04:db:3b:90 :2b:45:6a:2b:c0`

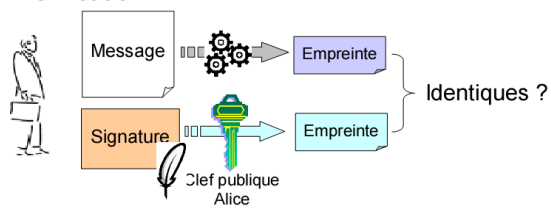
- Propriétés
  - Deux messages différents ont deux empreintes différentes
  - Connaissant h(M), il est impossible de trouver M

Signature numérique pour chiffrement asymétrique

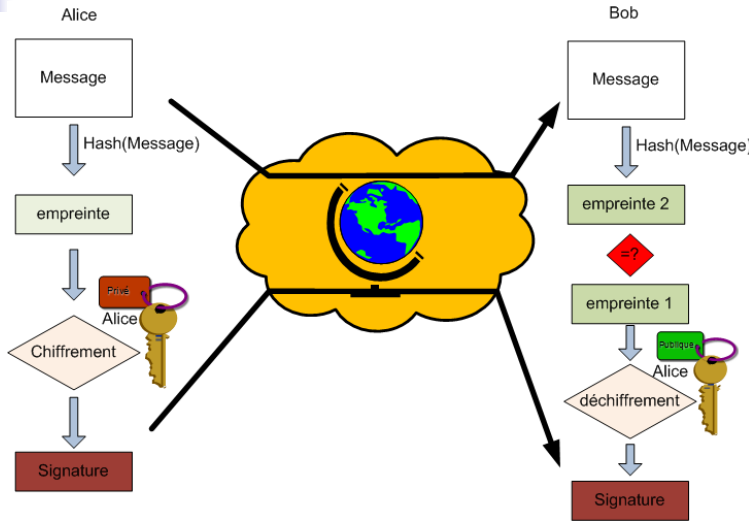
- Signature



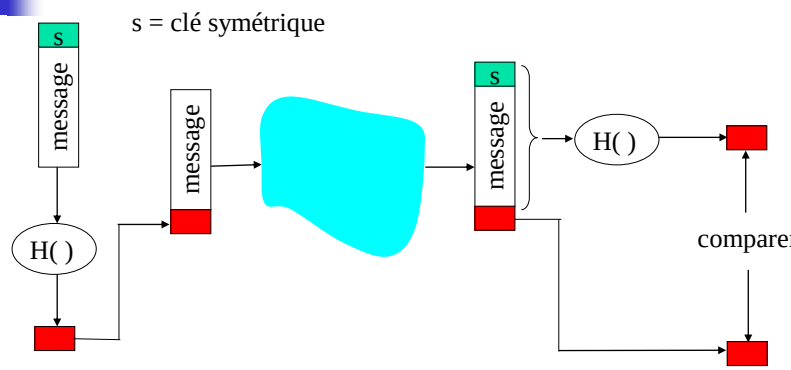
- Vérification



Signature numérique

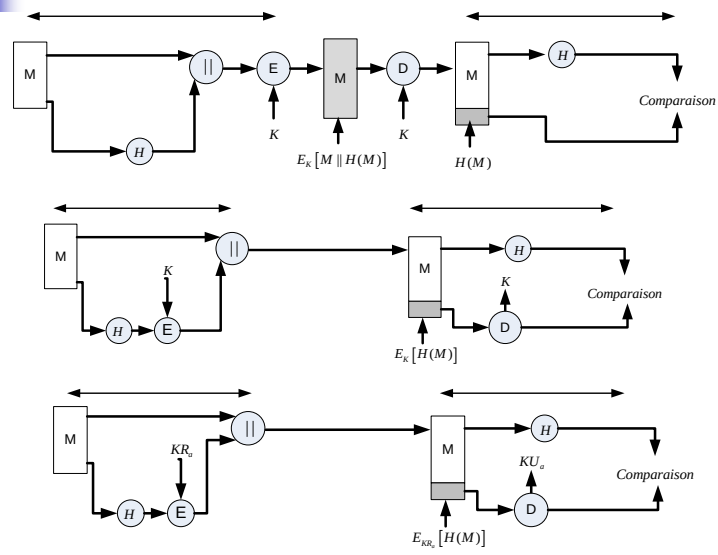


MAC: Message Authentication Code

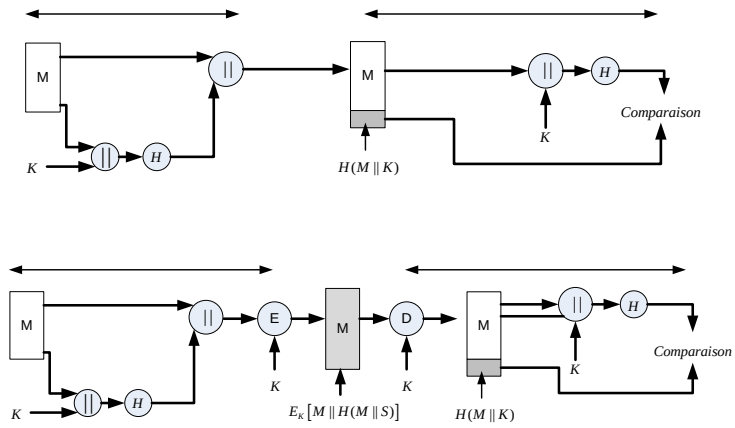


- Son rôle est d'assurer l'intégrité d'un message ***m***
- *S=H(m,s),* ***m*** est le message et ***s*** est la clé symétrique
- MAC (Message Authentication Code): ne fournit pas la propriété de non-répudiation

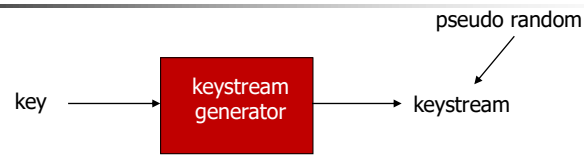
Utilisation d'une fonction de hachage (1)



Utilisation d'une fonction de hachage (2)



Chiffrement en continu



- Combinaison de bit de keystrem avec le bit du message en claire
- m(i) = i^{ième} bit du message
- ks(i) = i^{ième} bit de la clé
- c(i) = i^{ième} bit du message chiffré
- c(i) = ks(i) $\oplus$ m(i) ($\oplus$ = ou exclusif)
- m(i) = ks(i) $\oplus$ c(i)

Chiffrement par bloc

- Le message est divisé en bloc de k bits (64-bit par bloc).
- Chiffrement du k-bit du bloc en claire texte à k-bit bloc chiffré

Exemple avec k=3:

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

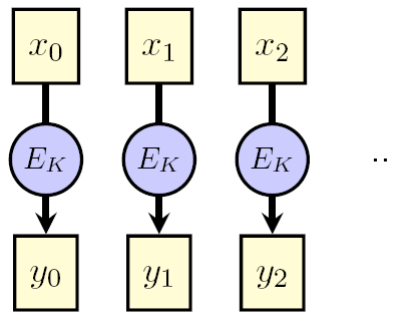
Quel est le message chiffré correspondant à 010110001111 ?

Chiffrement par bloc

- **Les quatre principaux modes de chiffrement**
  - ECB Electronic CodeBook
  - CBC Cipher Block Chaining
  - CFB Cipher FeedBack
  - OFB Output FeedBack

ECB : Electronic Code Book

- $C_i = E_k(M_i)$



- Le mode ECB n'assure aucune sécurité : ne pas l'utiliser.

Chiffrement par bloc (ECB)

Supposant le codage suivant:

input	output	input	output
A: 000	110	I: 100	011
K: 001	111	E:101	010
N: 010	101	L:110	000
O: 011	100	S:111	001

Quel est le message chiffré correspondant à « le soleil »

L	E	S	O	L	E	I	L
110	101	111	011	110	101	100	110

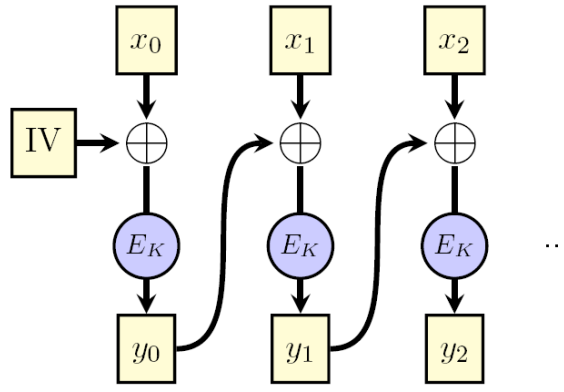
Texte chiffré:

000	010	001	100	000	010	011	000
A	N	K	I	A	N	O	A

Est-ce la fréquence de répétition des lettres est toujours la même avant et après le chiffrement ?

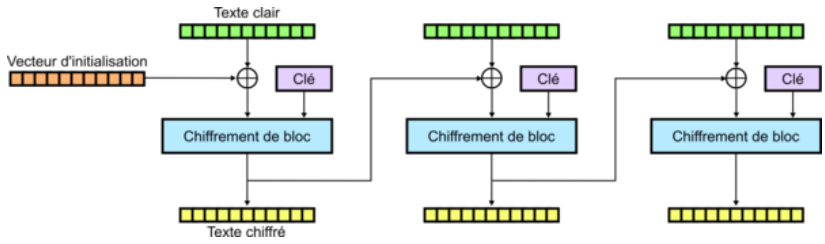
CBC : Cipher Block Chaining

- $C_0 = IV$
  $C_i = E_k(M_i \oplus C_{i-1})$



Mode opératoire: Cipher Block Chaining

- Le message est découpé en blocs de taille fixe
- Chaque bloc est chiffré de manière corrélée avec le bloc précédent en utilisant l'opération OU eXlusif (XOR (⊕)) entre le bloc de message i (M_i) et le résultat du chiffrement du bloc de Message M_{i-1}
  - à l'étape i,
    - On calcule: $M_i \oplus C_{i-1}$
    - Puis on chiffre le résultat: $C_i = E(M_i \oplus C_{i-1})$
    - Et on transmet C_i
  - pour l'étape 1 :
    - On introduit une valeur d'initialisation (appelé seed ou initialisation Vector (IV)) pour effectuer le premier XOR.



Chiffrement par bloc (CBC)

Supposant le codage suivant:

input	output	input	output
A: 000	110	I: 100	011
K: 001	111	E:101	010
N: 010	101	L:110	000
O: 011	100	S:111	001

Par exemple:
IV: 000

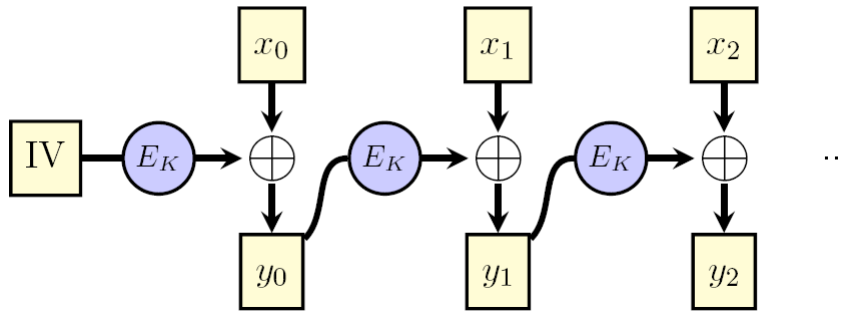
Quel est le message chiffré correspondant à « le soleil »

	L	E	S	O	L	E	I	L
	110	101	111	011	110	101	100	110
IV:	000	000	010	010	111	111	101	111
Xor:	110	101	101	001	001	010	001	001
Texte chiffré	000	010	010	111	111	101	111	…

Est-ce la fréquence de répétition des lettres est toujours la même avant et après le chiffrement ?

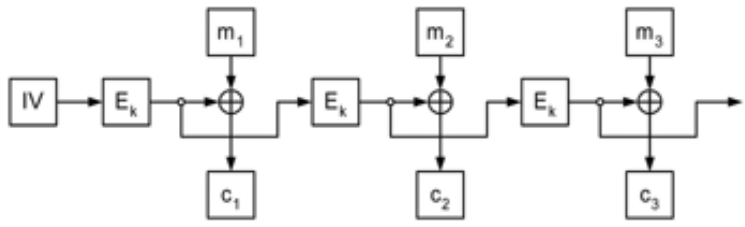
CFB : Cipher FeedBack

- $C_0 = IV$
  $C_i = M_i \oplus E_k(C_{i-1})$



OFB : Output FeedBack

- $Z_0 = IV$
  $Z_i = E_k(Z_{i-1})$
  $C_i = M_i \oplus Z_i$



Distribution des clés

- Les algorithmes de chiffrement symétriques nécessitent le partage d'une clé secrète
- Il faut donc assurer le transport sûr de cette clé
- Si la clé est compromise lors de la phase de distribution, toutes les communications le seront !

Distribution des clés

- Un utilisateur souhaitant communiquer avec plusieurs autres en assurant de niveaux de confidentialité distincts doit utiliser autant de clés qu'il a d'interlocuteurs
- Pour un groupe de N personnes utilisant un cryptosystème à clés secrètes, il est nécessaire de distribuer un nombre de clés égal à:

$$N * (N-1) / 2$$

Schémas de distribution de clés

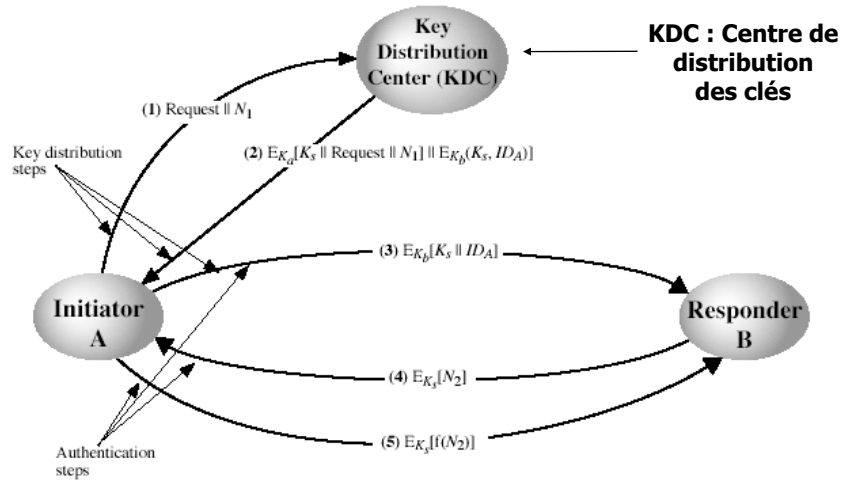
- Plusieurs variantes pour 2 partenaires
  - A choisit une clé et la transmet physiquement à B (valise diplomatique par exemple)
  - Une tierce partie (de confiance) C choisit un clé et assure la distribution à A et B
  - Si A et B ont déjà partagé une clé auparavant, ils peuvent utiliser l'ancienne clé pour chiffrer une nouvelle
  - Si A et B ont des communications sûres avec une tierce partie C, C peut relayer les clés entre A et B

Kerberos

Généralités

- Principes
  - Basé sur la notion de « Ticket »
  - Cryptographie à clé secrète (symétrique)
  - Authentification mutuelle
  - Tickets limités dans le temps
  - Mécanismes anti-rejeu (horodatage)
  - Pas de transmission de mot de passe sur le réseau
- Kerberos V5
  - Standards IETF : RFCs 1510 et 1964

Scénario de distribution



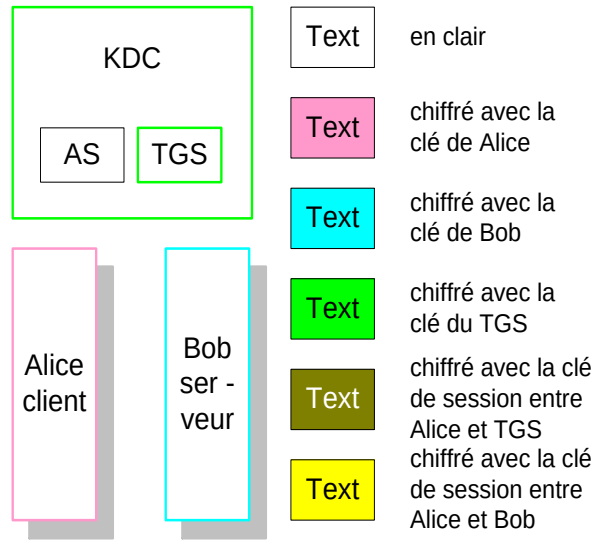
Scénario de distribution

L'utilisateur A (resp. B) dispose d'une clé K_a (resp K_b) avec le KDC

La requête envoyée par A au KDC contient les deux identités ID_a et ID_b

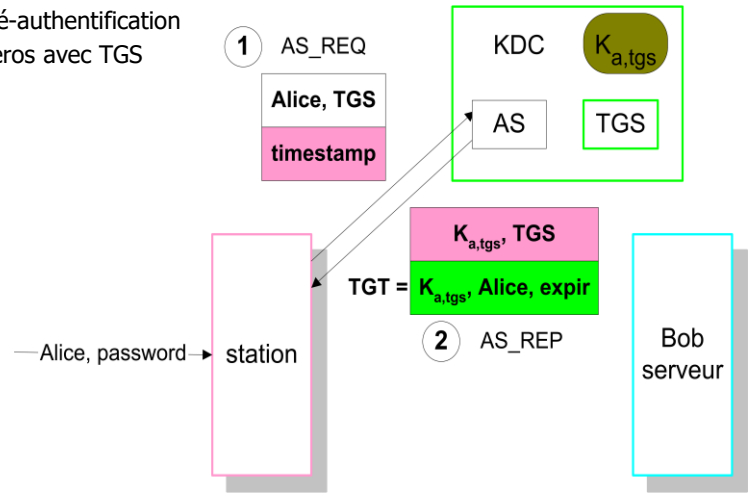
Le nombre aléatoire N_1 (resp. N_2) est destiné à la lutte anti rejeu

Kerberos



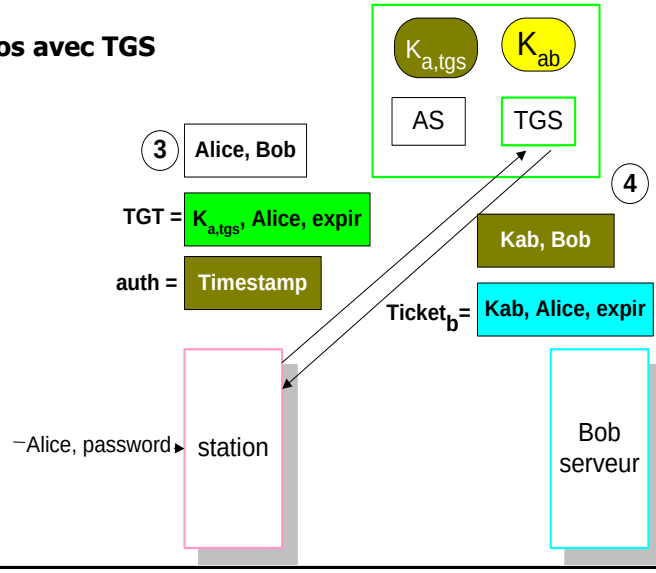
La pré-authentification avec kerberos

- La pré-authentification
- Kerberos avec TGS

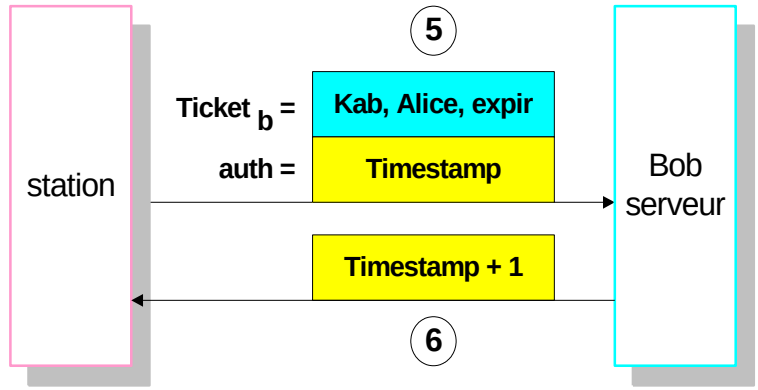


Kerberos

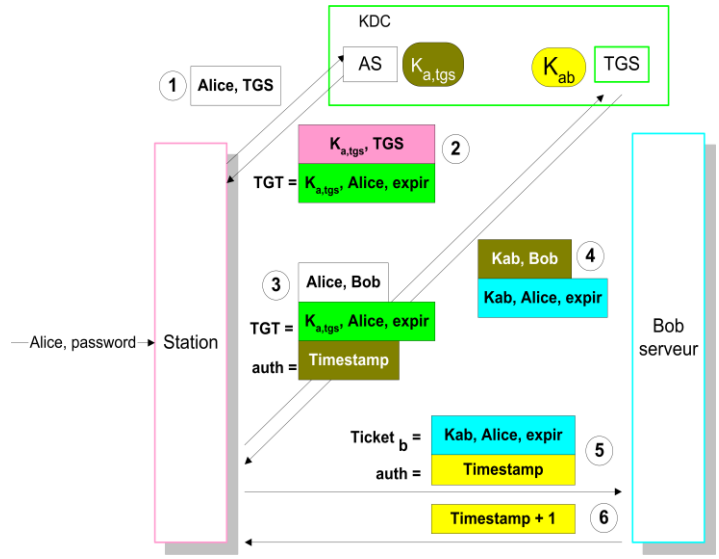
- **Kerberos avec TGS**



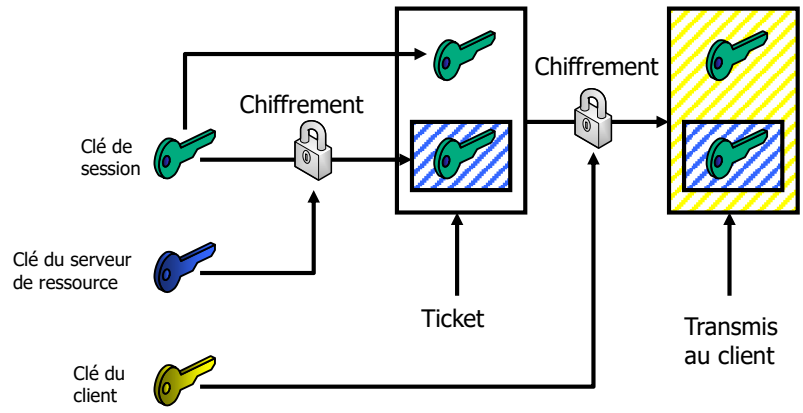
Kerberos



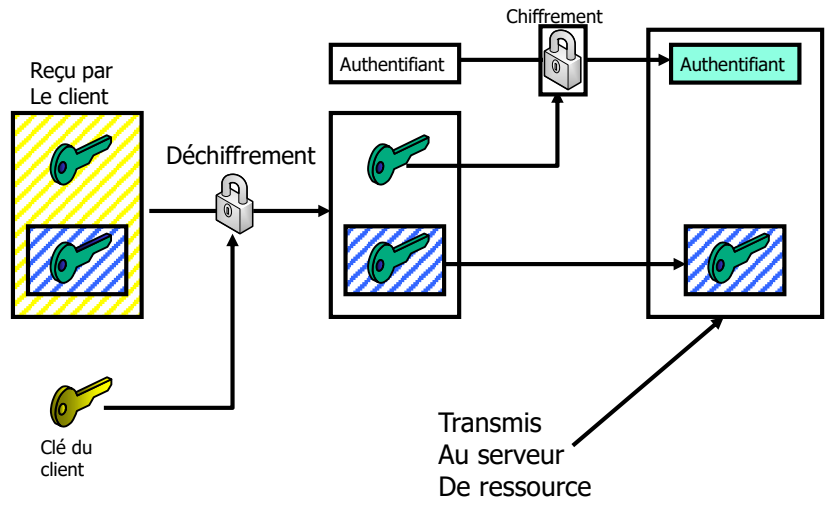
Kerberos



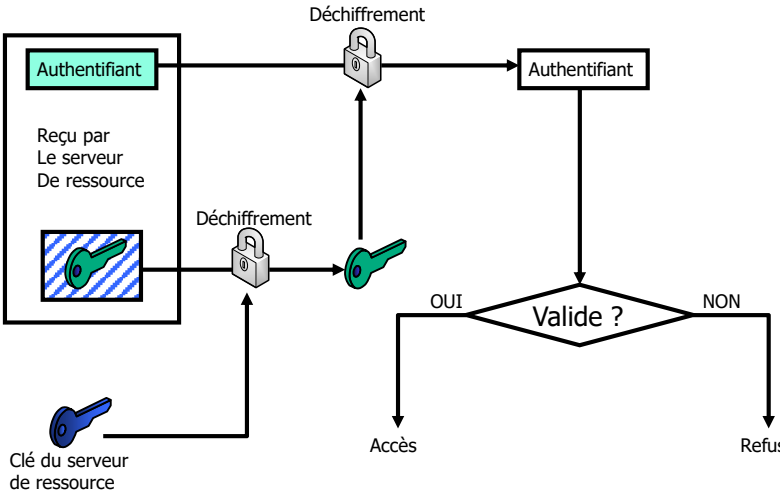
Génération d'un ticket



Traitement par le client



Traitement par le serveur



Accès à distance: FTP, telnet et SSH

- telnet (client et serveur)
  - yum install xinetd
  - yum install telnet-server telnet
  - nano /etc/xinetd.d/telnet
    - disable=no
  - /etc/init.d/xinetd restart

  ou

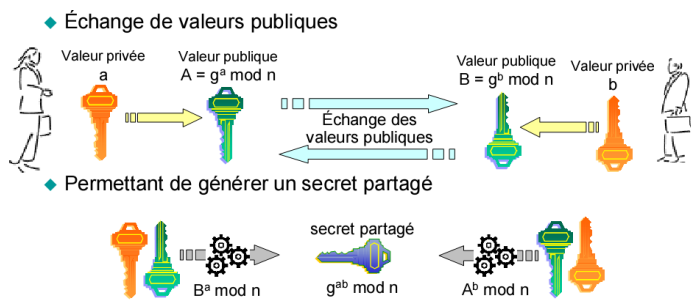
  service xinetd restart
  - telnet 192.168.0.13
    - Login:
    - Password:
    - Escape character is '^]'.

Accès à distance :FTP, telnet et SSH

- ftp (client et serveur)
  - Client en GUI: yum install gftp
  - Serveur: yum install vsftpd
    - Service vsftpd start
  - nano /etc/vsftpd/vsftpd.conf
    - #write_enable=YES => uncomment this option

Protocole d'échange de clés: Diffie-Hellman DH

- Qu'est ce que Diffie-Hellman (DH) ?
  - Inventé en 1976
  - permet à deux entités de générer un secret partagé
- Principe :
  - g et n sont deux valeurs publique (n est un nombre premier)
  - a et b deux valeurs choisis par Alice et Bob (privé!)
  - Alice et bob s'échangent (g^a mod n et g^b mod n)



Diffie-Hellman: Exemple

- n = 11 and g = 5
- Clés privées: a = 3 and n = 4

Clés publiques:

- $A = g^a \mod n = 5^3 \mod 11 = 125 \mod 11 = 4$
- $B = g^b \mod n = 5^4 \mod 11 = 625 \mod 11 = 9$

Clé partagée après l'échange

- $(T_B)^{S_A} \mod p = 9^3 \mod 11 = 729 \mod 11 = 3$
- $(T_A)^{S_B} \mod p = 4^4 \mod 11 = 256 \mod 11 = 3$

Clé partagée

- 3 = clé symétrique

Conclusion : une synthèse

- Bien que le *chiffrement* puisse rendre secret/confidentiel le sens d'un document, d'autres techniques cryptographiques sont nécessaires pour communiquer de façon sûre.
- Pour vérifier l'intégrité ou l'authenticité d'un document, on utilise respectivement un *Message Authentification Code* (MAC) ou une *signature numérique*.
- L'utilisation d'un *compteur* associé aux messages échangés permet de s'affranchir du problème du re-jeux
- On peut aussi prendre en considération *l'analyse de trafic* dont la communication peut faire l'objet, puisque les motifs provenant de la présence de communications peuvent faire l'objet d'une reconnaissance de motifs. Pour rendre secrète la présence de communications, on utilise la stéganographie.