CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Cryptographie

Marc GILG

19/06/2018

Plan du module

1. Pourquoi la cryptographie
2. Les fonctions de hachage
3. Le chiffrement symétrique
4. L'échange de clé Diffie-Hellman
5. Le chiffrement asymétrique
6. Les infrastructures de gestion de clés
7. The Gnu Privacy Gard
8. Transport Layer Security

CyberEdu
La sécurité par l'enseignement supérieur des NTIC

1. Pourquoi la cryptographie ?

a) Protection des informations et communications
b) Intégrité
c) Confidentialité
d) Authentification

1. Pourquoi la cryptographie ?

a. Protection des informations et des communications

- Un besoin très ancien
  - le plus ancien document chiffré est une recette secrète de poterie qui date du XVIe siècle av. J.-C., qui a été découverte dans l'actuelle Irak.
  - Chiffre de César lors de l'antiquité
- Protection de l'information
  - secret de fabrication
  - secret militaire
- Besoin croissant avec les moyens numériques
  - authentification des logiciels
  - Accès distants
  - Économie Numérique

1. Pourquoi la cryptographie ?

b. Intégrité

- Besoin de savoir si une donnée a été altérée lors
  - d'un traitement
  - de sa conservation
  - de sa transmission
- L'altération peut être
  - volontaire
  - accidentelle
- En cryptographie les fonctions de hachage permettent de détecter les altérations.

1. Pourquoi la cryptographie ?

C. confidentialité

- Une information a une certaine valeur
- Il faut en limiter l'accès
- Des algorithmes de chiffrements sont utilisés
- une clé permet rendre l'information lisible
- ce mécanisme ne fournit pas d'authentification

1. Pourquoi la cryptographie ?

d. Authentification

- Valider une identité :
  - pour accéder à l'information ou au système d'information
  - pour identifier la source de l'information
  - pour donner des droits
  - pour identifier une ressource, un serveur

- Faire le lien entre une personne physique et un moyen numérique
  - Ce lien ne peut être uniquement d'ordre algorithmique
  - Ce lien doit être établi par un humain

CyberEdu
La sécurité par l'enseignement supérieur des NTIC

2. Fonctions de hachage

a) Définition
b) Collisions
c) Propriétés
d) Exemples et utilisations
e) Salage

2. Les fonctions de hachage

a. définition

- Une fonction de hachage calcule une empreinte à partir d'un fichier ou plus généralement d'une donnée
- Une empreinte a une taille fixe de quelques octets
- Une fonction de hachage est à sens unique : difficulté de calculer la fonction inverse
- Une fonction de hachage a un pouvoir discriminant : deux données différentes ont en général des empreintes différentes, mais ce n'est pas toujours le cas

2. Les fonctions de hachage

b. Collisions

- On associe à une infinité de données une empreinte de taille fixe
- On aura forcément des données qui ont la même empreinte
- Une **collision** se produit si deux données ont la même empreinte
- Une fonction de hachage est parfaite s'il n'existe pas de collision, malheureusement, cela est impossible
- On cherche a minimiser les collisions

2. Les fonctions de hachage

c. propriétés

- Il est difficile de trouver le contenu de départ à partir de l'empreinte
- Il est difficile de générer une donnée pour créer une collision à partir d'un message existant, connaissant son empreinte et la fonction de hachage
- Il est difficile de trouver deux données qui ont la même empreinte

2. Les fonctions de hachage

d. Exemples et utilisations

- Exemples :
  - MD5
  - Sha1, Sha256, Sha512,...

- Utilisations :
  - Vérifier si un fichier n'est pas corrompu lors d'un téléchargement
  - Vérification de mots de passe
  - Vérification de l'intégrité

2. Les fonctions de hachage

e. salage

- Un mot de passe est sauvegardé par son empreinte
- On peut réaliser une recherche de collision sur l'empreinte en utilisant des tables de mots de passe connus (dictionnaire)
- On utilise la méthode de **salage** pour rendre cette attaque difficile :
  - Une chaîne pseudo-aléatroire est créée à partir de l'identifiant (le sel)
  - On concatène cette chaîne avec le mot de passe
  - On sauvegarde l'empreinte de cette concaténation

CyberEdu
La sécurité par l'enseignement supérieur des NTIC

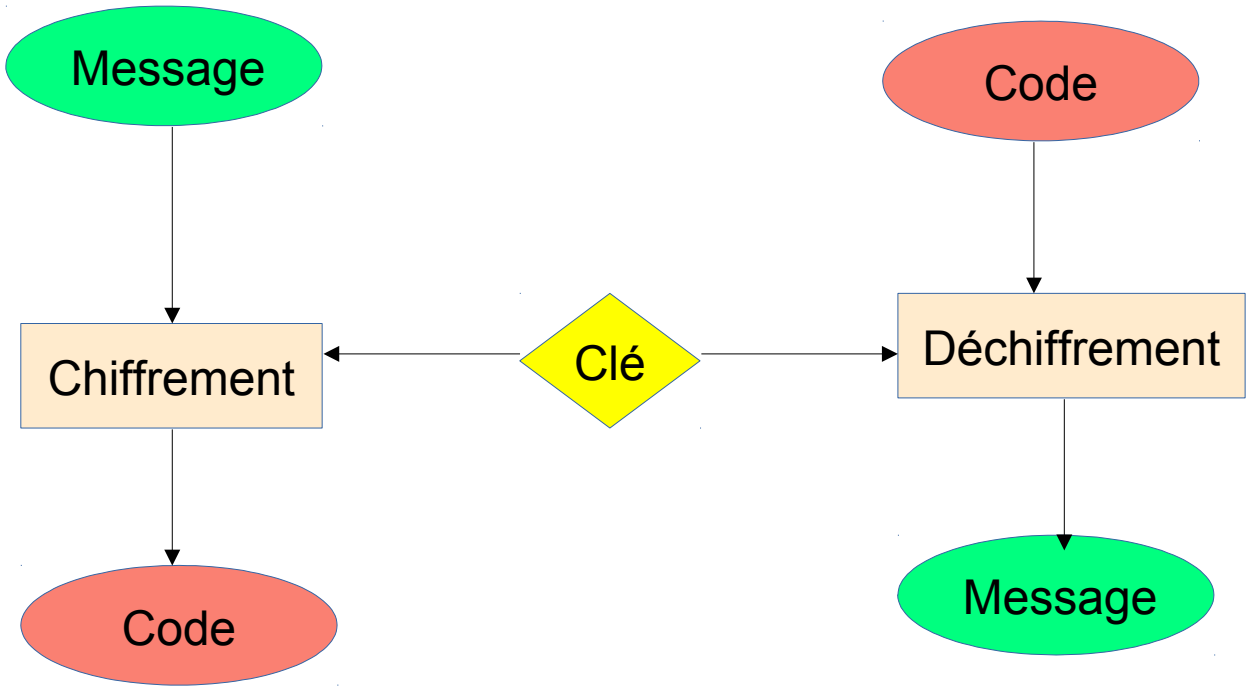
3. Chiffrement symétrique

a) Définition
b) Chiffrement par flots
c) Chiffrement par blocs

3. Chiffrement Symétrique

a. définition

- Un algorithme de **chiffrement symétrique** utilise **la même clé** pour chiffrer et déchiffrer



3. Chiffrement Symétrique

b. Chiffrement par flots

- Le chiffrement par flots est un chiffrement bits à bits.
- Convient à la voix sur IP
- Exemple : le chiffrement de Vernam :
  - On utilise une clé K de la taille du message M
  - Pour chiffrer, on calcule C = K xor M
  - Pour déchiffrer, on calcule M = K xor C
- Le chiffrement est très sûr, mais nécessite une clé très grande (de la taille du message)

3. Chiffrement Symétrique

c. Chiffrement par blocs

- Le chiffrement s'applique à des blocs de même taille
- Nécessité de découper le message en blocs de taille identique
- Exemples de chiffrement par bloc :
  - DES, 3DES
  - AES
  - Blowfish, serpent, towfish

3. Chiffrement Symétrique

d. Modes d'opération

- Comment appliquer le chiffrement aux blocs de message M_i ?
- **ECB : Electronique Code Book**
  $C_i=E_k(M_i)$: Chiffrement bloc par bloc

Image d'origine

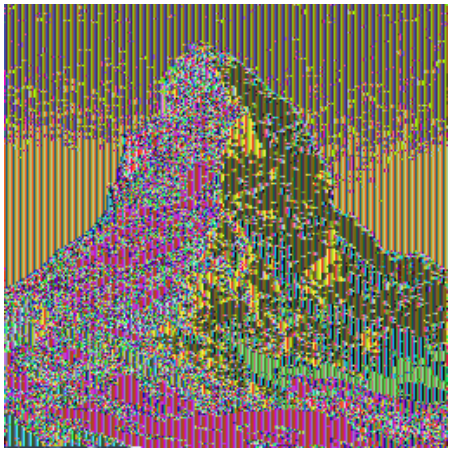


Image avec ECB

source wikipedia.org

3. Chiffrement Symétrique

d. Modes d'opération

CBC:Cipher Block Chaining

$$\begin{Bmatrix} C_0 = VI \\ C_i = E_k\left(M_i \otimes C_{i-1}\right) \end{Bmatrix}$$

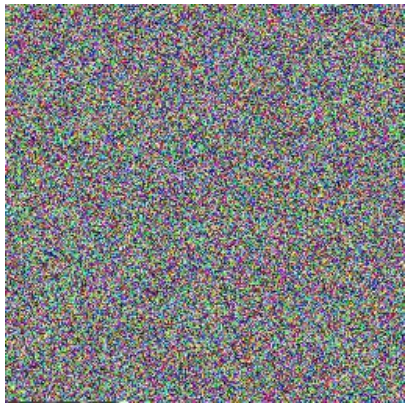


Image avec CBC

CFB : Cipher Feedback

$$\begin{Bmatrix} C_0 = VI \\ C_i = M_i \otimes E_k\left(C_{i-1}\right) \end{Bmatrix}$$

≈ Chiffrement par flot

4. Échange de Clés – Diffie-Hellman

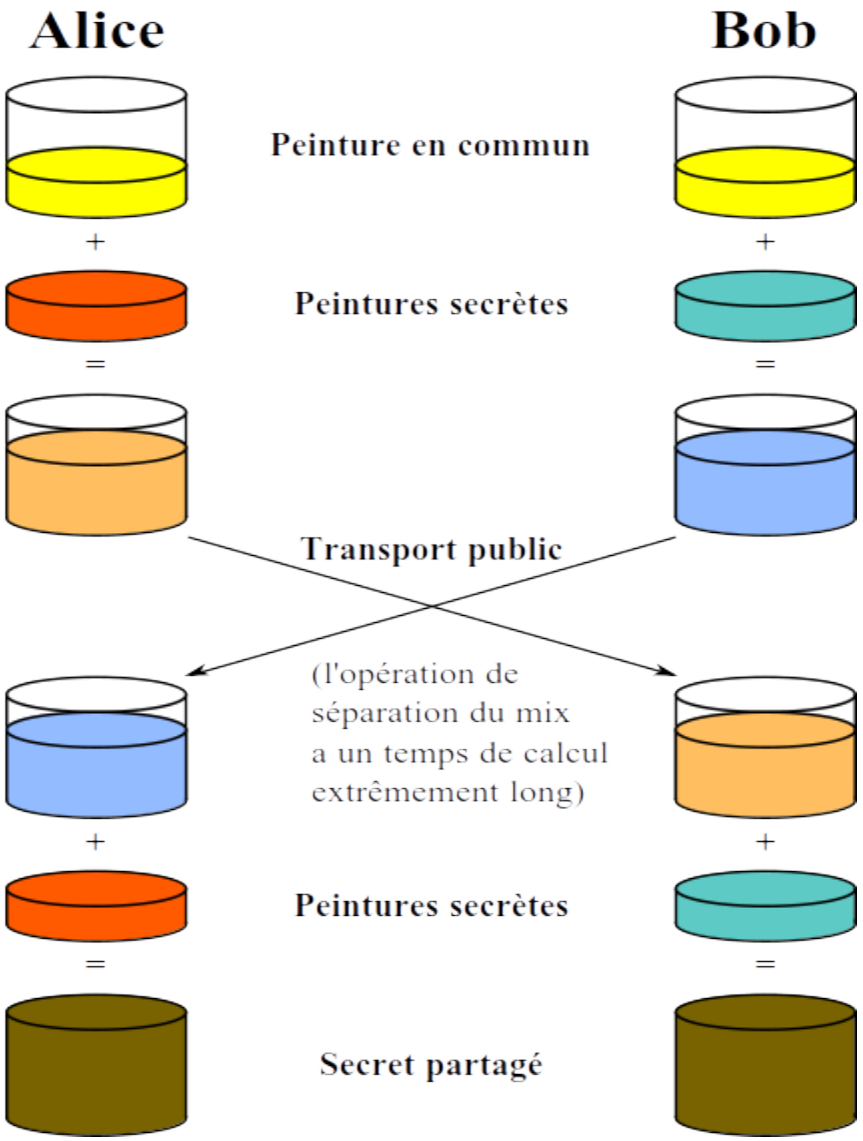
a) Définition
b) Initialisation
c) Transmission et calcul de la clé

4. Échange de clés Diffie-Hellman

a. définition

- Whitfield Diffie et Martin Hellman sont des mathématiciens américains
- Ils ont proposé une méthode d'échange de clés en 1976
- Le principe est que deux personnages Alice et Bob s'échangent des informations pour créer une clé
- Un troisième personnage Eve intercepte la communication
- **Les informations échangées, captées par Eve, ne permettent pas de déduire la clé**

4. Échange de clés Diffie-Hellman



wikipedia.org

4. Échange de clés Diffie-Hellman

b. Initialisation

- Alice et Bob se mettent d'accord :
  - sur un nombre premier **p**
  - sur un générateur **g**
- Alice choisi un nombre secret **a**
- Bob choisi un nombre secret **b**

4. Échange de clés Diffie-Hellman

b. Transmission

- Alice calcul **$A = g^a \bmod p$** et le transmet à Bob.
- Bob calcul **$B = g^b \bmod p$** et le transmet à Alice.

c. Calcul de la clé

- Alice calcul **$K = B^a \bmod p$**
- Bob calcul **$K = A^b \bmod p$**

On ne peut pas déduire **K** de **g,p,A,B** car il n'y a pas de **logarithme** modulo **p**

CyberEdu
La sécurité par l'enseignement supérieur des NTIC

5. Chiffrement asymétrique

a) Définition
b) Chiffrement
c) Signature
d) Exemples

5. Chiffrement asymétrique

a. définition

- Utilisation de deux clés **différentes**
- Les clés sont liées mathématiquement
- Une clé est :
  - **privée**
  - l'autre **publique**
- **Si on chiffre avec la clé privée, on déchiffre avec la clé publique, et vice-versa**

5. Chiffrement asymétrique

b. Chiffrement

- Alice chiffre avec la clé **publique** de Bob
- Bob déchiffre avec **sa** clé **privée**
- **Bob est seul a connaître la clé pour déchiffrer**
- **On authentifie le destinataire**

5. Chiffrement asymétrique

c. signature

- Alice calcul l'**empreinte** de son message
- Alice signe l'**empreinte** avec **sa** clé **privée**
- **Bob authentifie** le message :
  - il vérifie l'***empreinte*** avec la clé **publique** de Alice
  - il recalcule l'**empreinte** du message et la compare à l'***empreinte*** qui a été calculée par Alice
- **On authentifie la source**

5. Chiffrement asymétrique

d. exemples

- RSA : Ronald **R**ivest, Adi **S**hamir, Leonard **A**dleman

- ElGamal

6. Infrastructure de Gestion de Clés

a) Besoin d'une IGC
b) Certificats
c) Acteurs d'une IGC
d) Schéma de principe
e) Validation d'un certificat

6. Infrastructure de Gestion de Clés

a. Besoin d'une Infrastructure de Gestion de Clés

- Une clé est une donnée abstraite, mathématique
- il faut faire le lien entre cette donnée et un propriétaire
- C'est comme mettre une étiquette à une clé

6. Infrastructure de Gestion de Clés

b. Certificat

- Un **certificat** contient :

  - Une clé publique
  - Des informations sur le propriétaire
  - Une date de début de validité
  - Une date de fin de validité
  - Un numéro de série
  - Des informations de l'autorité de certification

  - **Une signature des toutes les informations ci-dessus par l'autorité de certification**

6. Infrastructure de Gestion de Clés

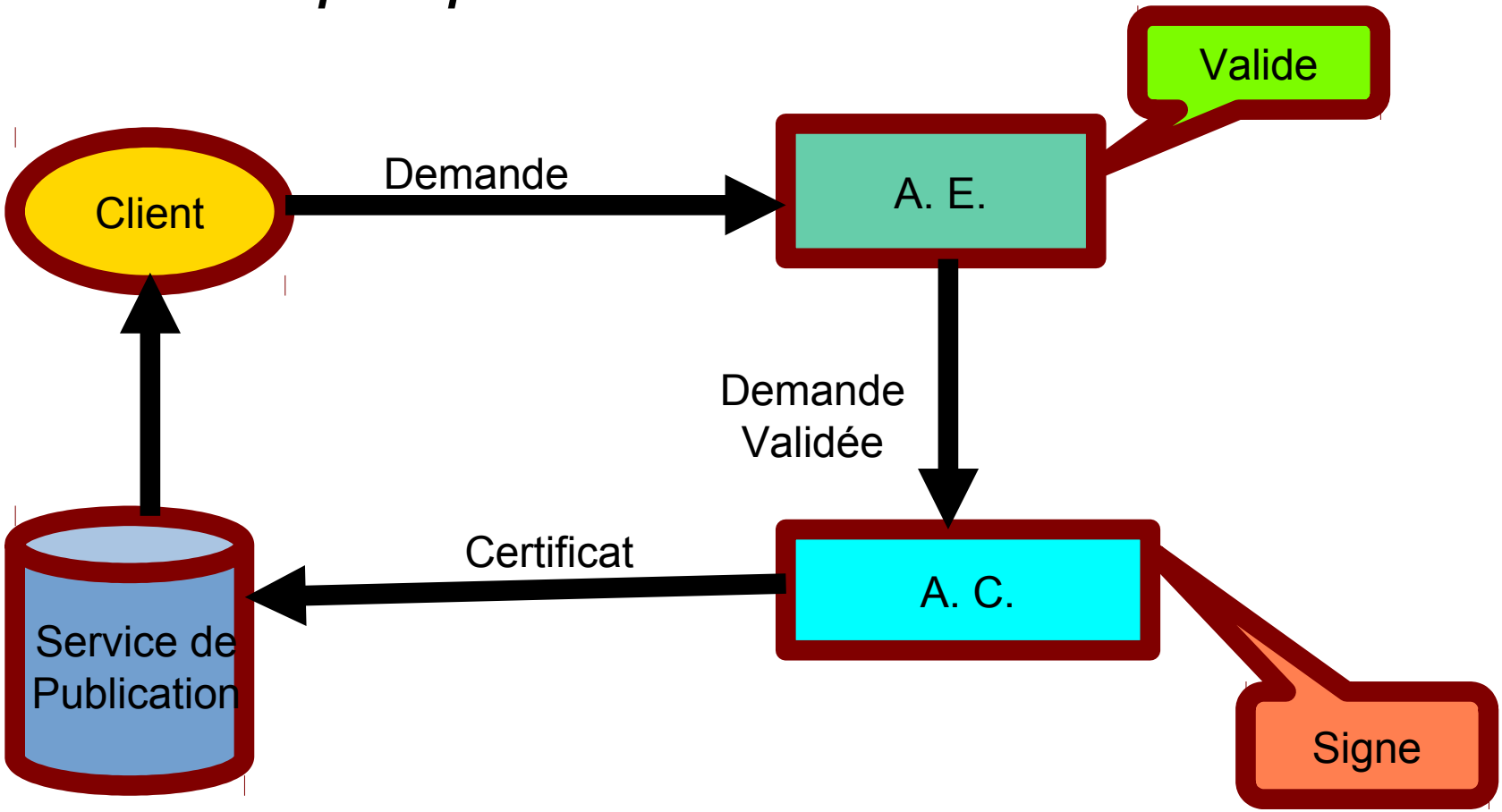
c. Acteurs d'une IGC

- Le **Client** qui demande le certificat
- L'**autorité d'enregistrement** (A.E.) qui valide la demande
- L'**autorité de certification** (A.C.) qui signe le certificat
- Un **service de publication**
- Un **service de révocation**

- Options :
  - **Service d'horodatage**
  - **Service de séquestre**

6. Infrastructure de Gestion de Clés

d. Schéma de principe



6. Infrastructure de Gestion de Clés

e. Validation d'un certificat

- Un **certificat** est **validé** en **vérifiant la signature** de l'A.C.
- **Pour vérifier** la signature **il faut le certificat de l'A.C.**
- Pour le Web, le certificat de l'A.C. est présent dans le navigateur

Peut-on avoir confiance dans le certificat d'autorité ?

CyberEdu
La sécurité par l'enseignement supérieur des NTIC

7. Gnu Privacy Gard

a) PGP
b) Signature de clés
c) Toile de confiance
d) Validation de clés
e) Utilisations

7. Gnu Privacy Gard

a. PGP

- **Pretty Good Privacy** (PGP) a été crée en 1991

- **PGP** offre des services :
  - d'**Authentification** :
    signature de l'empreinte par une clé privée
  - de **Confidentialité** :
    chiffrement du message par une clé symétrique (clé de session), elle-même chiffrée par une clé publique
  - de **Compression** :
    compression zip du message avant le chiffrement : augmentation de l'entropie
  - de **Segmentation** : découpage du message en bloc
  - de **Compatibilité** : utilisation d'un code ASCII sur 8 bits

7. Gnu Privacy Gard

b. Signature de clés

- Détecter la falsification d'un trousseau de clé
- Authentifier le propriétaire d'une clé publique
- Signature direct d'une clé :
  - On vérifié l'identité du propriétaire
  - On vérifie l'empreinte de la clé publique avec les données fournies par le propriétaire
  - On signe la clé publique
  - On a un rôle d'autorité de certification

- **Si on ne connaît pas directement le propriétaire, il faut un autre mécanisme.**

7. Gnu Privacy Gard

c. Toile de confiance

- On ne peut pas signer directement toutes les clés
- Une clé qu'on a signée permet de signer d'autres clés.
- Niveau de confiance de la clé publique pour la signature :
  - **unknown** : on ne sait rien sur la validation, niveau par défaut
  - **none** : la vérification n'est pas consciencieuse
  - **marginale** : les clés sont validées avant signature
  - **full** : signature identique que par sa propre clé (~ autorité)

- **Le niveau de confiance en une clé est personnel et privé**

7. Gnu Privacy Gard

c. Validation de clés.

- Une clé est valide si :
  - elle a été **signé par sa propre clé**
  - elle a été **signé par une clé** de confiance **full**
  - elle a été **signé par trois clés** de confiance **marginale**
  - **si la distance** (nombre de signatures intermédiaires) **entre sa clé et la clé à valider est inférieure à 5**

- **Ces valeurs sont des valeurs par défaut et peuvent être modifiées**

7. Gnu Privacy Gard

c. Utilisation de gpg sous linux :

- Créer une clé : **gpg –gen-key**
- Afficher les clés sécretes : **gpg –list-secret-keys**
- Afficher les clé publiques : **gpg --list-keys**
- Exporter une clé publique vers un serveur :
  gpg --keyserver localhost –send-key keyID
- Récupérer une clé d'un serveur :
  gpg --keyserver localhost –recv-key keyID
- Signer une clé : **gpg –sign-key keyID**
- Chiffrer un fichier :
  gpg --output essai.gpg --encrypt –recipient marc.gilg@uha.fr essai.txt
- Déchiffrer un fichier :
  gpg --output essai.ok --decrypt essai.gpg

8. Transport Layer Security

a) Services de sécurité
b) Modèle OSI
c) Types de paquets
d) Négotiations
e) Utilisations

8. Transport Security Layer

a. Services de sécurité :

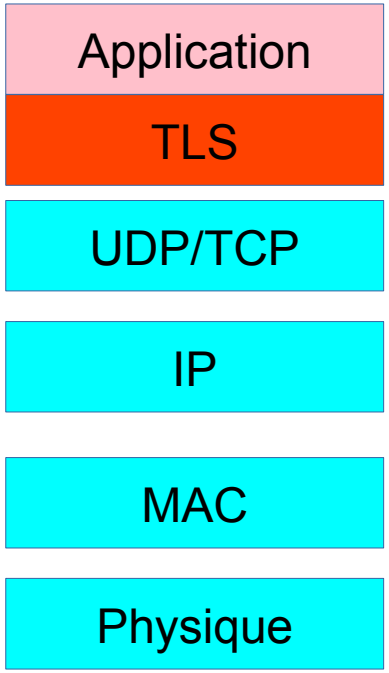
TLS est un protocole de communication client-serveur pour sécuriser les communications. Son ancêtre SSL a été créé dans le but de sécuriser les communications Web en offrant les services de sécurité suivant :

- Intégrité
- Confidentialité
- Anti-rejeu
- Authentification du serveur
- Option : authentification des clients

8. Transport Security Layer

b. Modèle OSI :

TLS s'intercale entre la couche transport et applicatif. Pour le programmeur il suffit d'appeler la bibliothèque TLS au lieu des bibliothèques TCP ou UDP



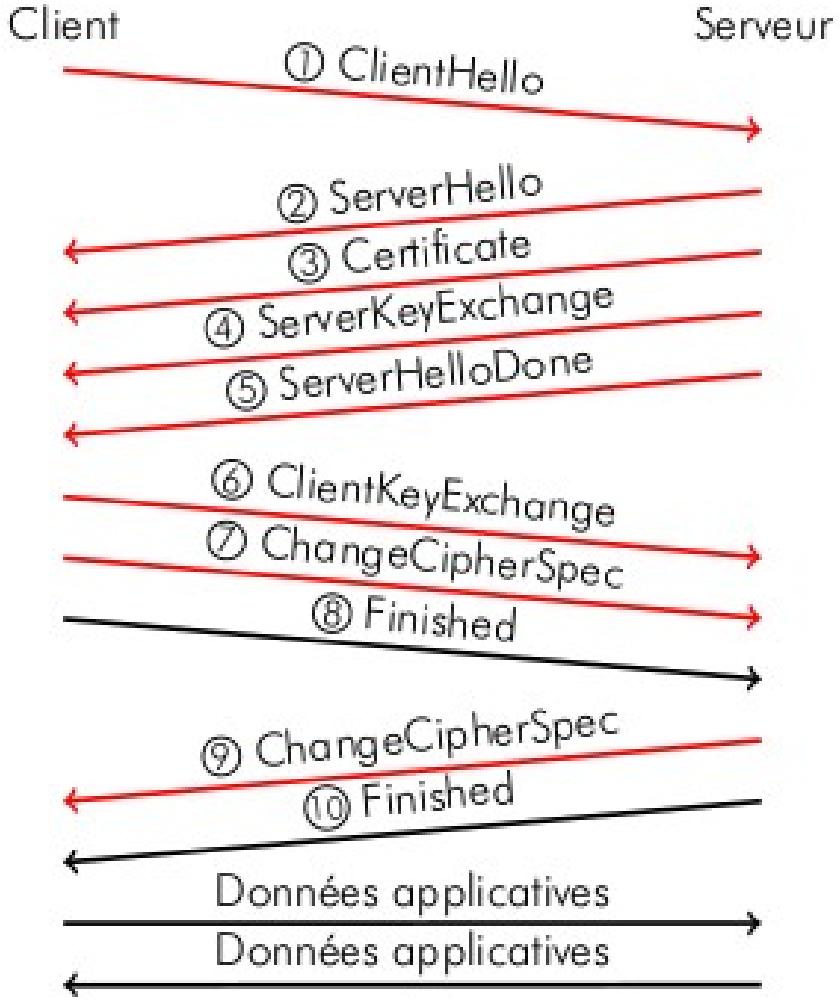
8. Transport Security Layer

c. Types de paquets TLS :

- Handshake : négotiation
- Alerte : contrôle
- ChangeCipherSpec : changement de chiffrement, de clés
- Application Data : les données applicatifs

8. Transport Security Layer

d. Négociation TLS :



Source ANSSI

8. Transport Security Layer

d. Négociation TLS :

1) ClientHello : Le client envoie une liste de suite chiffrements
2) ServerHello : le serveur choisi une suite de chiffrement
3) Certificate : le serveur envoie son certificat contenant sa clé publique
4) ServerKeyExchange : le serveur envoie un secret chiffré avec sa clé privée
5) ServerHelloDone : mise en attente du serveur, pas d'authentification client
6) ClientKeyExchange : envoie d'un secret par le client chiffré avec la clé publique du serveur
7) ChangeCipherSpec : le client envoie les informations sur le chiffrement (algo + clés)
8) Finished : le client envoie de manière chiffré l'empreint des échanges précédents
9) ChangeCipherSpec : le serveur confirme les infos sur le chiffrement
10) Finished : le serveur envoie l'empreinte des échanges

8. Transport Security Layer

e. Utilisations :

- Protocoles https, imaps, pops, smtps, etc ….
- Vérification d'un site par un navigateur :
  - Vérifie le certificat à l'aide d'une autorité connu par le navigateur
  - Vérifie si le nom du certificat correspond au nom DNS du site

CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Merci de votre attention