Introduction aux Réseaux

DIU « Enseigner l'Informatique au Lycée »

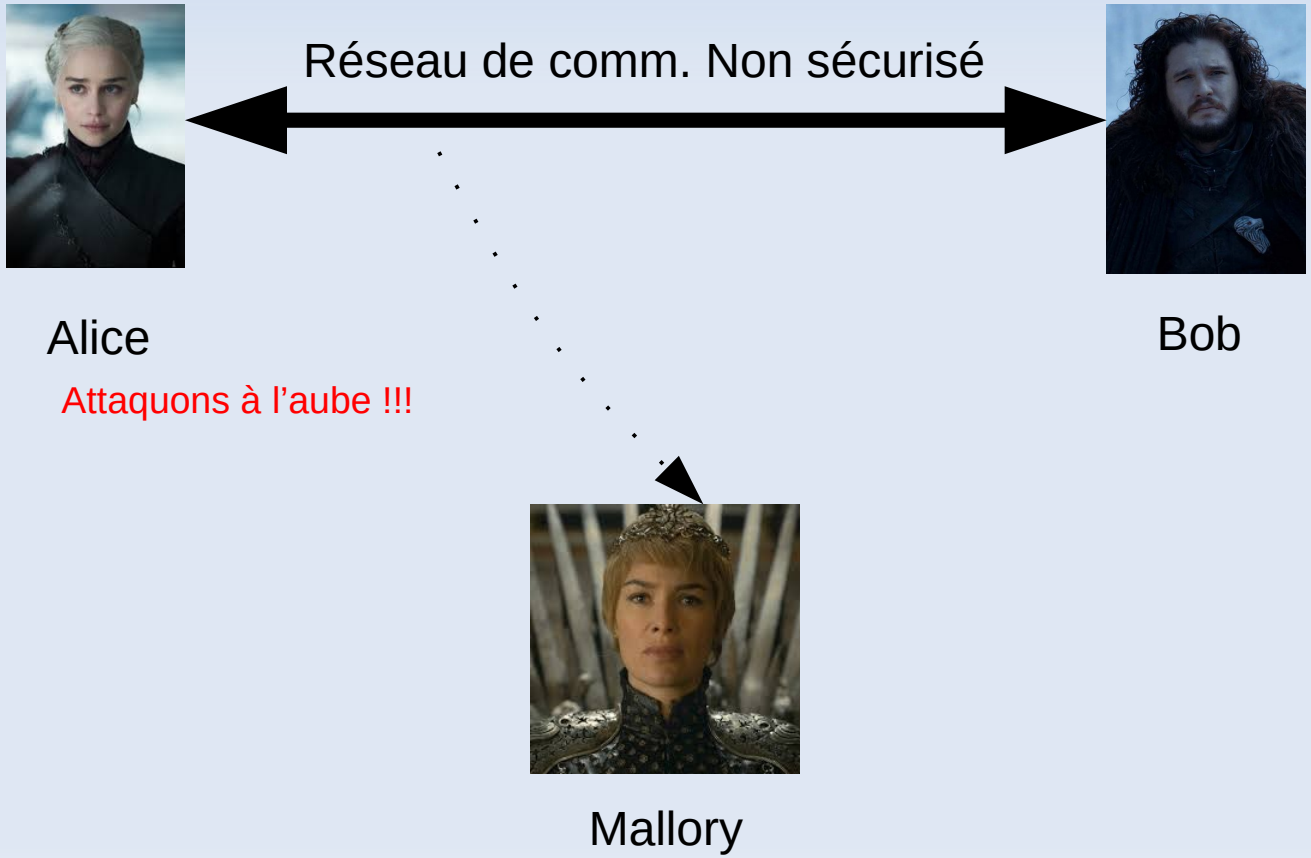
Université de Bordeaux

aurelien.esnard@u-bordeaux.fr
abdou.guermouche@u-bordeaux.fr

Sécurité des Communications

Contexte

- Alice veut transmettre une information secrète à Bob (et seulement a Bob) en utilisant un réseau non sécurisé.
- Mallory veut avoir accès à cette information.

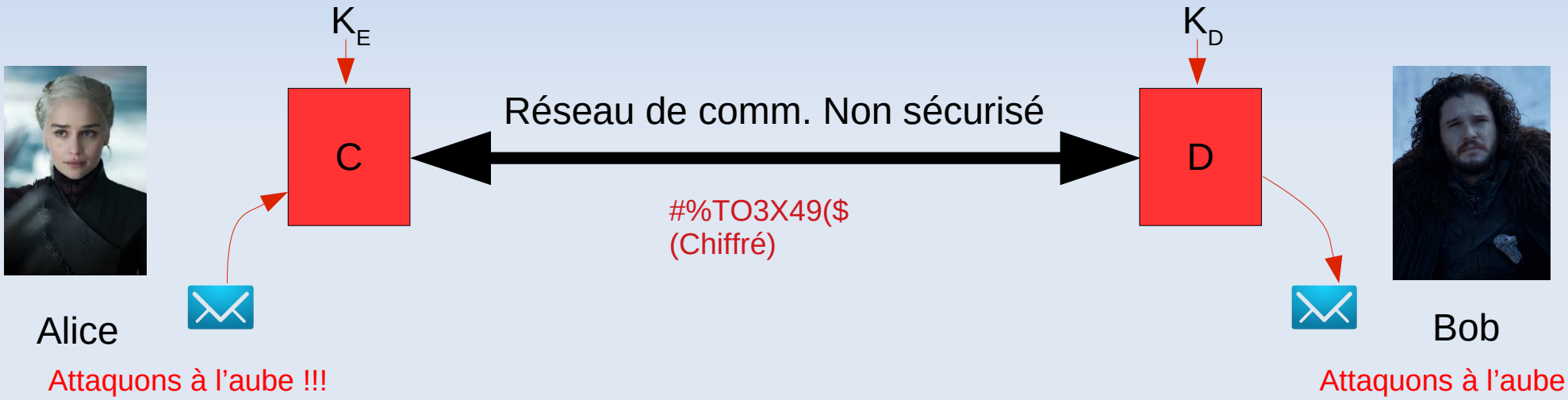


Cryptographie

- Un élément clé dans tous les système de sécurité
- Élément essentiel pour assurer
  - Confidentialité
    - Seules les personnes autorisées ont accès aux données
  - Intégrité des données
    - Seules les personnes autorisées peuvent modifier les données
  - Authentification
    - Prouver l'identité
  - Non répudiation
    - L'émetteur d'un message ne peut pas dire qu'il ne l'a pas fait

Utilisation du chiffrement

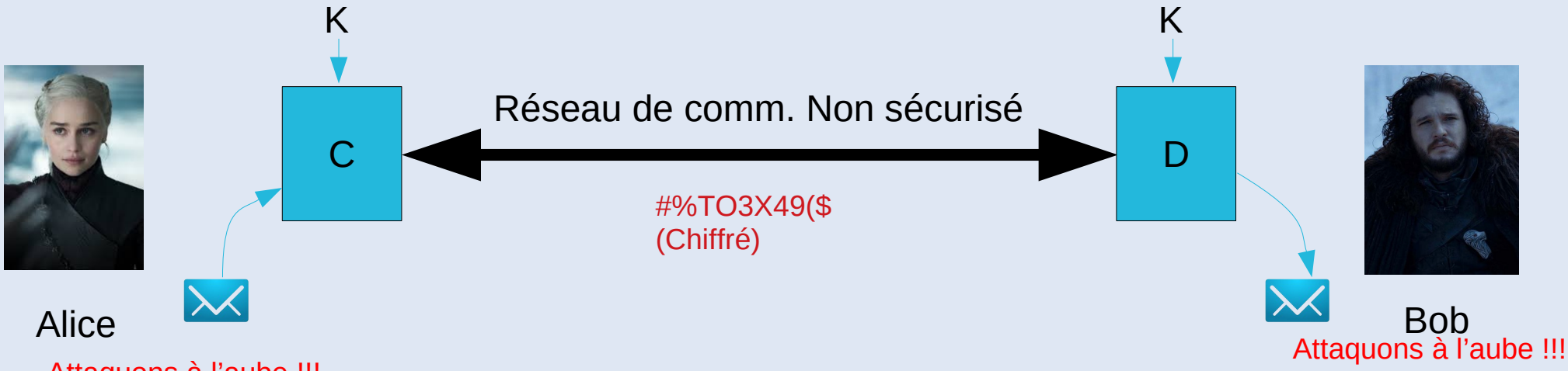
- Alice veut transmettre une information secrète à Bob (et seulement a Bob) en utilisant un réseau non sécurisé.



- Comment gérer les clés ?
- Quel algorithme utiliser ?

Chiffrement symétrique

- Chiffrement de déchiffrement avec la même clé
- i.e. $K_E=K_D$
- La clé doit être connue d'Alice et de Bob.
- Exemples : AES, DES, …

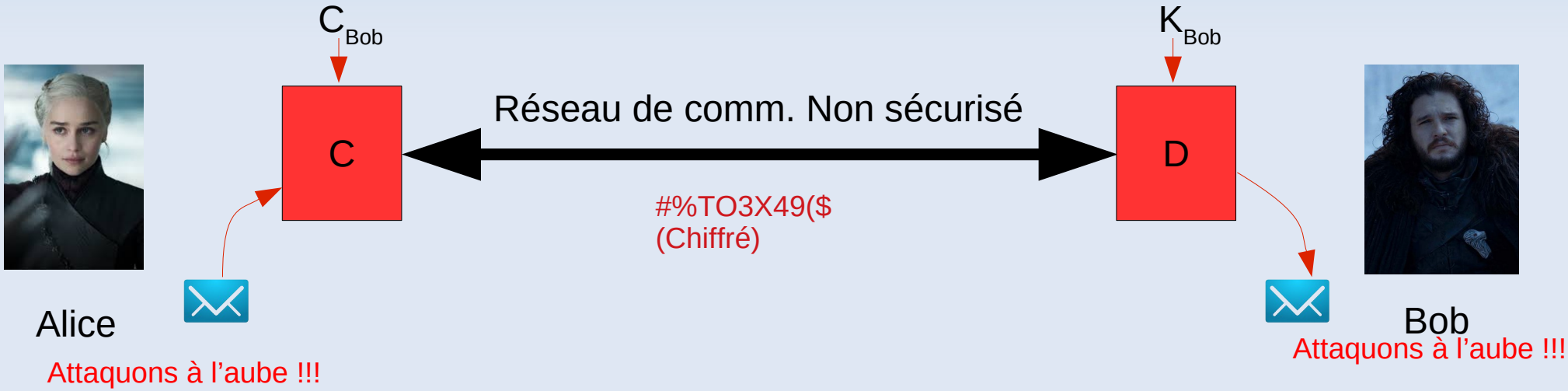


Chiffrement asymétrique

- Clé de chiffrement et de déchiffrement différente
- i.e. $K_E \neq K_D$
- Alice et Bob possèdent chacun une paire de clé C,K telles que
  - K_{Alice} est privée à Alice (resp. Bob)
  - C_{alice} est publique
  - Tout ce qui est chiffré avec C_{Alice} peut être déchiffré avec K_{Alice} et réciproquement.
- Exemples : RSA, ECC, …

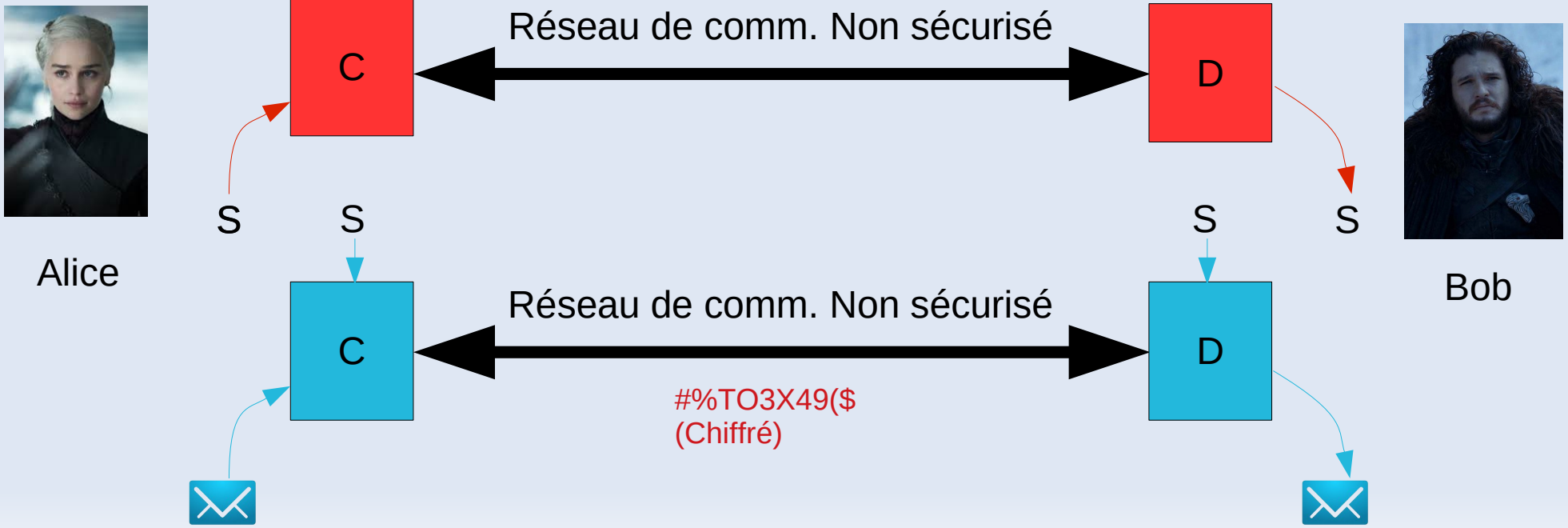
Chiffrement asymétrique

- Scénario simple
  - Chiffrer avec la clé publique C
  - Déchiffrer avec la clé privée K



Chiffrement asymétrique

- Scénario réaliste
  - Générer une clé aléatoire secréte S (symétrique)
  - Chiffrer S avec C et l'envoyer
  - Déchiffrer S avec K
  - Utiliser S pour chiffrer le trafic

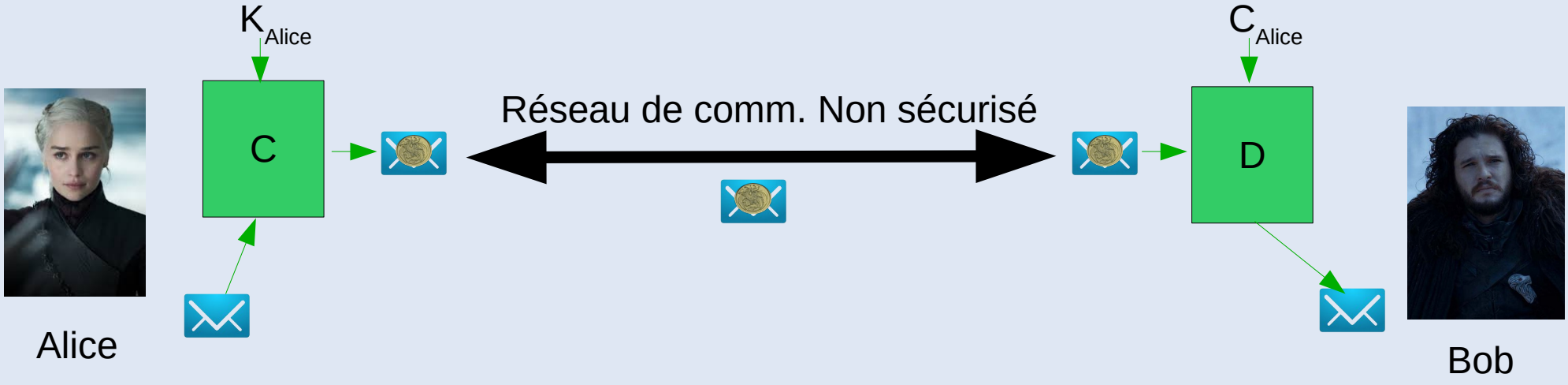


Algorithmes de hachage

- Permettent la vérification de l'intégrité du message
- Fonctions à sens unique calculant une empreinte du message
  - Facilité de calcul du hachage d'un message
  - Impossibilité de retrouver le message à partir du hachage
  - Impossibilité de construire deux messages ayant le même hachage
  - Impossibilité de modifier un message sans mise à jour du hachage
- Sha256, Sha1, MD5, ...
- Exemples
  - `echo "bonjour" | sha1sum`
    `1F71E0F4AC9B47CD93BF269E4017ABAAB9D3BD63`
  - `echo "Attaquons à l'aube!!!" | sha1sum`
    `8073B9D9B2EB74F31F9AE87359AF440883380D7E`

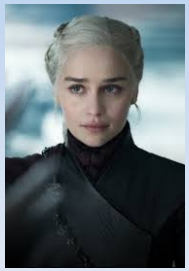
Signature électronique

- Permet de vérifier l'authenticité du message
- Générer le hachage H du message
- Chiffrer H avec K_{alice} et envoyer le résultat avec le message
- Bob peut vérifier la signature en utilisant C_{alice}
  - Bob est sûr que le message n'est pas corrompu si le résultat du déchiffrement est identique au hachage qu'il calcule
  - Bob est sûr qu'Alice est l'émetteur du message



Certificats électroniques

- Que se passe-t-il si Alice n'a pas C_{bob} initialement

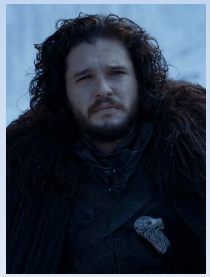


Alice

C_{bob} ?

C_{bob}

.
.
.



Bob

- Problème du Man-In-The-Middle

Alice

C_{bob} ?

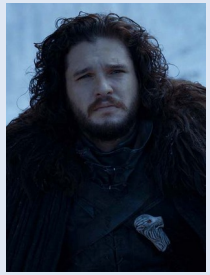
$C_{Mallory}$

Mallory

.
.
.

C_{bob} ?

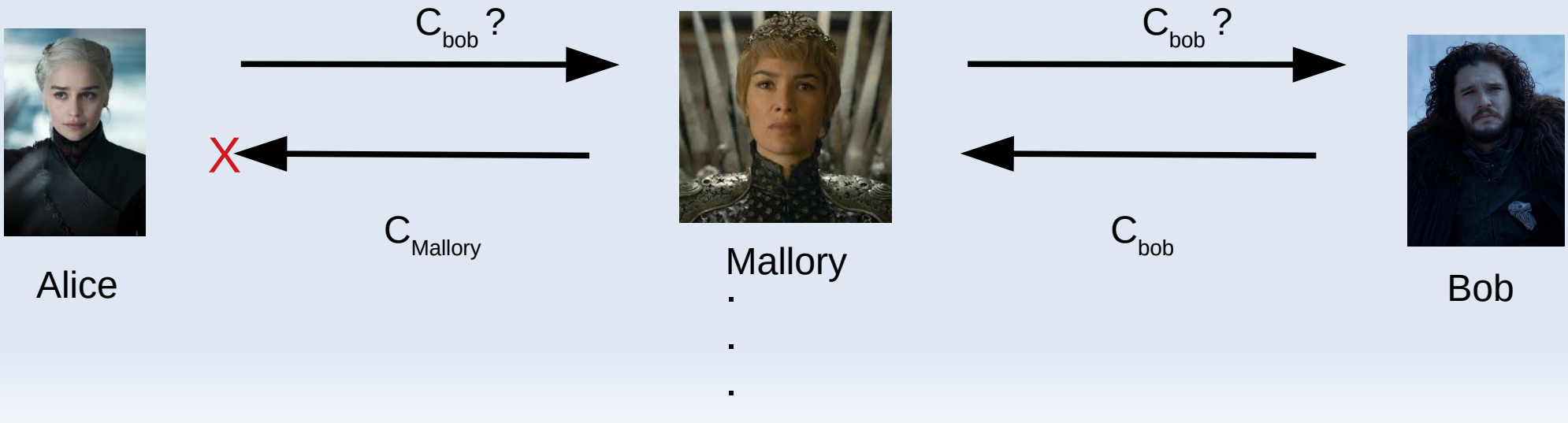
C_{bob}



Bob

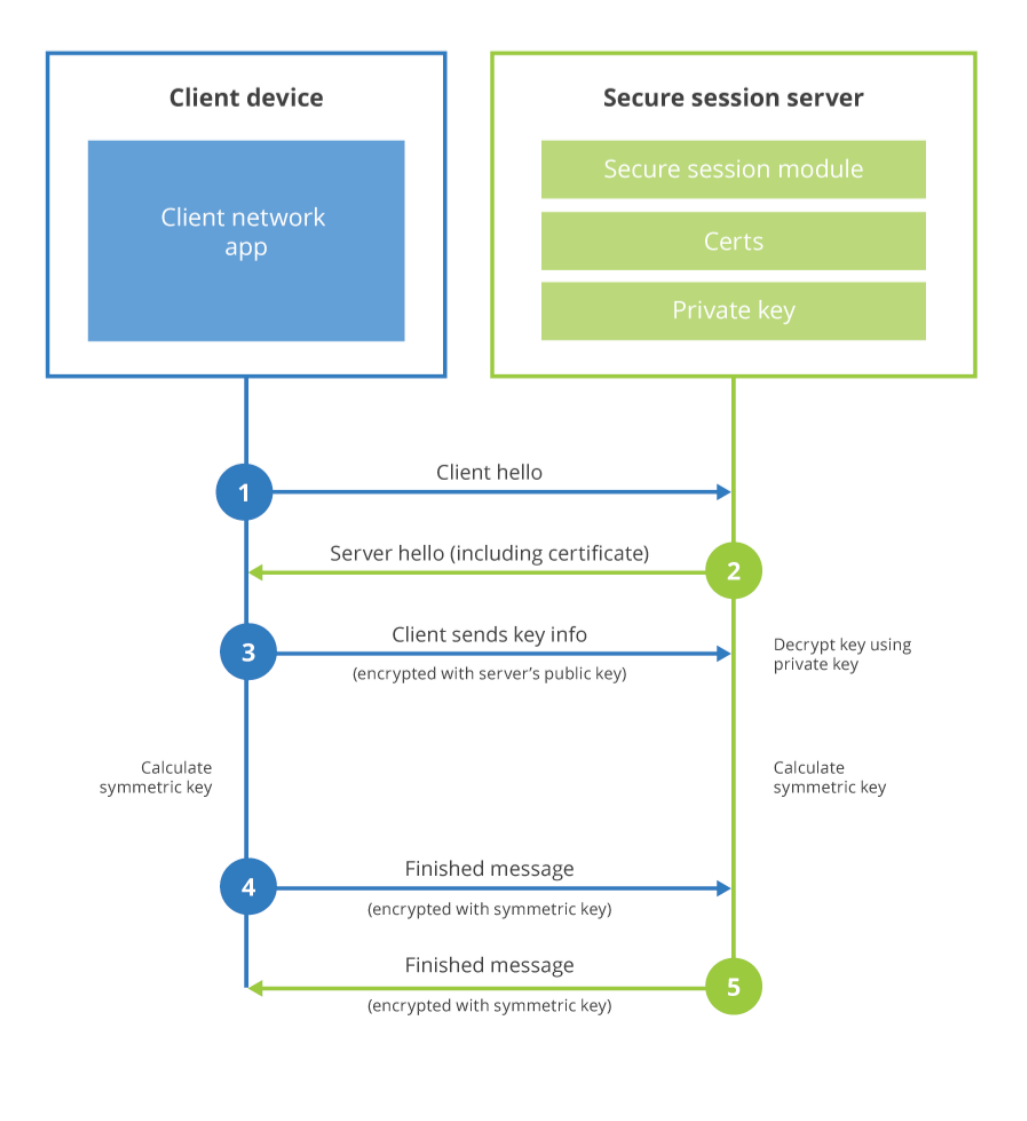
Certificats électroniques

- Un certificat contient
  - Une clé publique
  - Une identité (dans un format clé/valeur)
  - Une signature par une autorité de confiance dont la clé publique est connue
- À la réception du certificat de Bob, Alice peut vérifier que le certificat appartient bien à Bob
- Mallory ne peut plus usurper l'identité de Bob



SSL/TLS

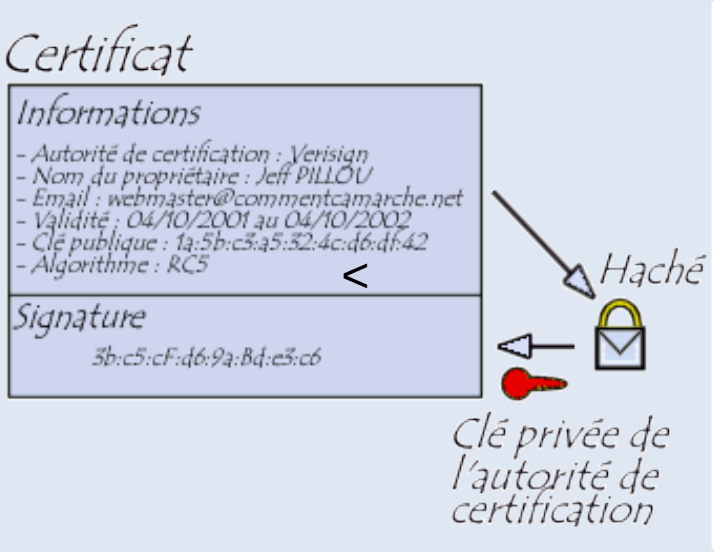
- Protocole de sécurisation des échanges sur Internet
- Basé sur l'utilisation de certificats
- Utilisé pour l'implémentation de versions sécurisées des protocoles standards (HTTPS, SMTPS, IMAPS, ...)

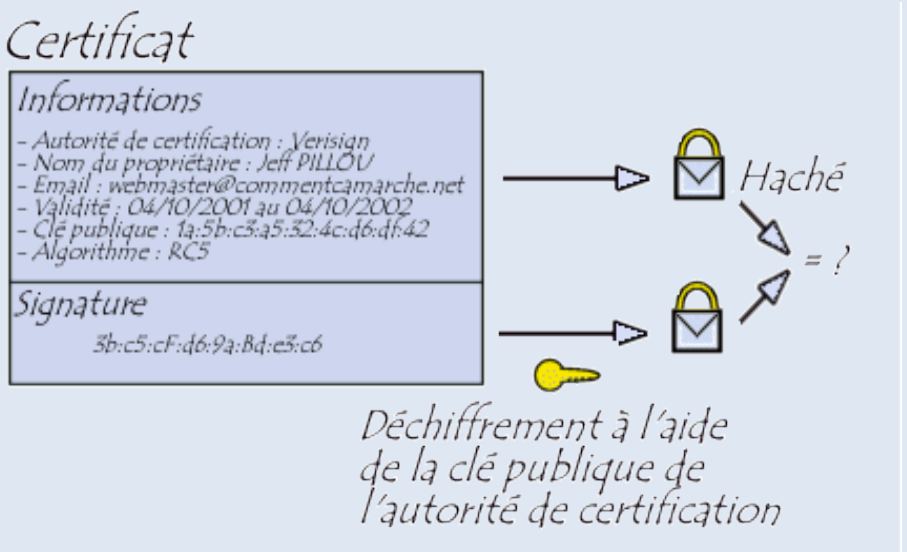


Source : https://blog.cloudflare.com/protecting-the-origin-with-tls-authenticated-origin-pulls/

HTTPS

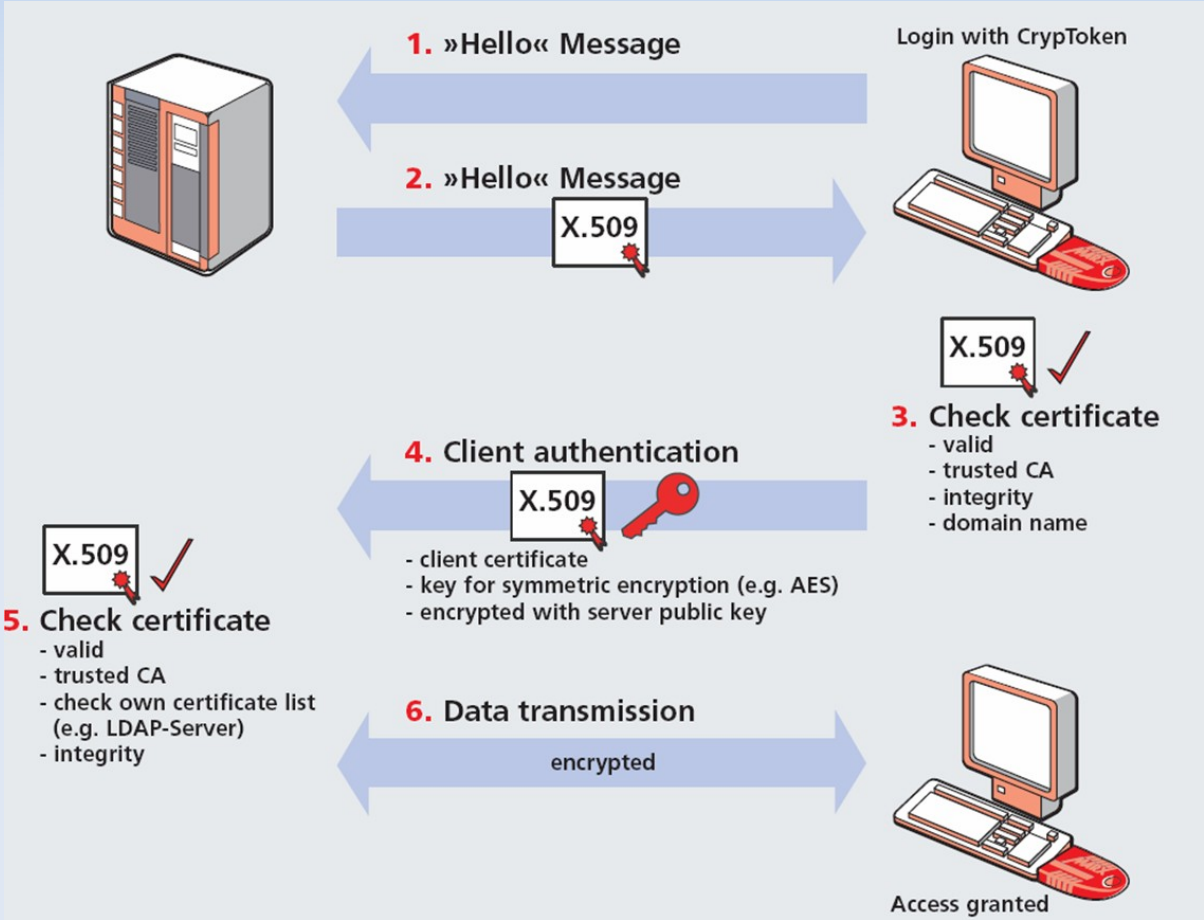
- HTTPS (HTTP Secure)
  - Utilisation transparente du protocole HTTP au-dessus de TLS/SSL (port 443 au lieu de 80)
  - Authentification du serveur web via son certificat (signé du CA)
  - Confidentialité et intégrité des données envoyées au serveur
  - Authentification du client facultative





HTTPS

- Authentification du serveur et du client avec SSL/TLS



Source : wikipedia