

Sécurité des communications

Jonathan Lejeune

Sorbonne Université/LIP6-INRIA

SRCS – Master 1 SAR 2019/2020

sources :

cours de Bénédicte Le Grand

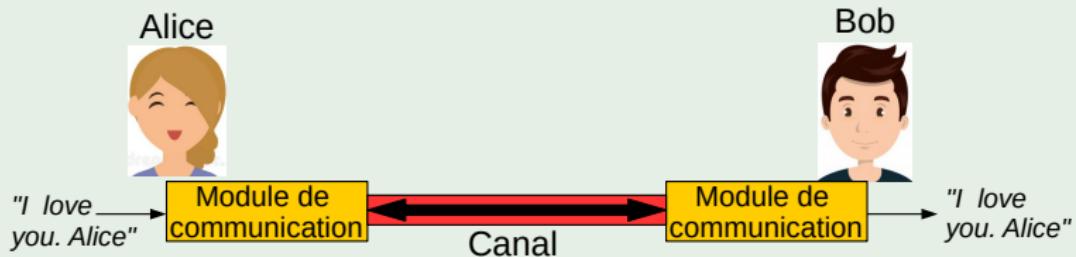
cours de Fabrice Legond-Aubry

computer networks, Tanenbaum and Wetherall

Contexte illustratif

Objectif

Alice et Bob souhaitent communiquer en toute sécurité.



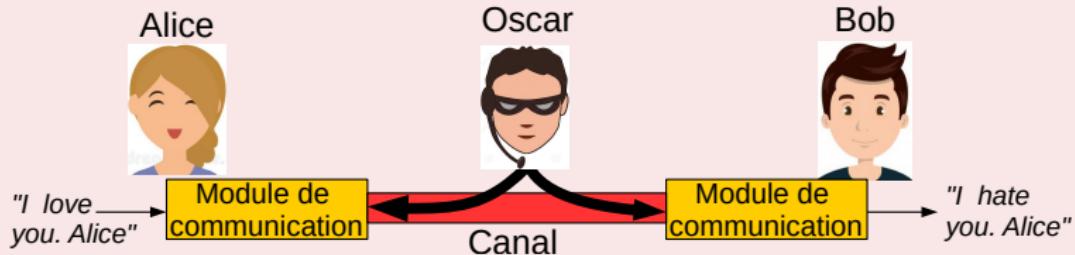
Que représentent Alice et Bob ?

- Des utilisateurs humains
- Client et serveur (ex : banque en ligne, achat en ligne ...)
- Des routeurs échangeant des mises à jour de tables de routage
- Des nœuds d'un système pair à pair
- Des serveurs mails

Contexte illustratif

Mais il existe un adversaire

Oscar, un pirate qui interfère dans la communication entre Alice et Bob



Comment Oscar peut-il interférer dans la communication ?

- Écouter et interpréter la communication
- Usurper une identité
- Modifier/altérer des messages
- Empêcher la communication

Confidentialité

Assurer que seules les personnes habilitées puissent lire une donnée

- Seuls Alice et Bob doivent pouvoir comprendre le contenu du message
- Oscar ne doit pas être en mesure de comprendre le message

Authentification

Assurer qu'une personne est bien ce qu'elle prétend être

- Alice doit s'assurer de bien s'adresser à Bob
- Bob doit s'assurer que c'est bien Alice qui s'adresse à lui
- Oscar ne peut ni se faire passer pour Alice ou pour Bob
- Attention à ne pas confondre avec l'**identification** qui ne prouve rien

Intégrité

Assurer que seules les personnes habilitées puissent modifier une donnée

- A la réception, Alice et Bob doivent retrouver le même message qui a été envoyé
- Oscar ne doit pas être en mesure d'altérer le contenu du message sans que Alice ou Bob ne le détecte

Disponibilité

Assurer qu'un accès à un service soit rapide, permanent et sans faille

- Alice doit pouvoir communiquer avec Bob en permanence et vice-versa
- Oscar ne doit pas pouvoir empêcher que Alice et Bob communiquent

La traçabilité

Pouvoir retracer tous les événements au cours d'une certaine période

- si Oscar lance une attaque, il faut en garder une trace pour la détecter a posteriori

Non-répudiation

Assurer que l'auteur d'un acte ne peut ensuite dénier l'avoir effectué

- Alice, Bob et Oscar ne doivent pouvoir nier aucun envoi de message qu'ils ont effectivement envoyé
- Alice, Bob et Oscar ne doivent pouvoir nier aucune réception de messages qu'ils ont effectivement reçus



Comment assurer ces propriétés dans un système distribué ?

- **Confidentialité** : clés de chiffrement
- **Authentification** : clés de chiffrement
- **Intégrité** : signature numérique (clés de chiffrement + hachage)
- **Disponibilité** : machines de contrôle (proxy, firewall, surveillant ...)
- **Traçabilité** : logging des événements sur un support fiable
- **Non-répudiation** : certificat + signature numérique

La cryptologie

La science du secret qui englobe :

- **la cryptographie** : discipline qui étudie la transformation d'un texte en clair en un texte chiffré via des clés de chiffrement
- **la cryptanalyse** : technique qui consiste à déduire un texte en clair d'un texte chiffré sans posséder la clé de chiffrement

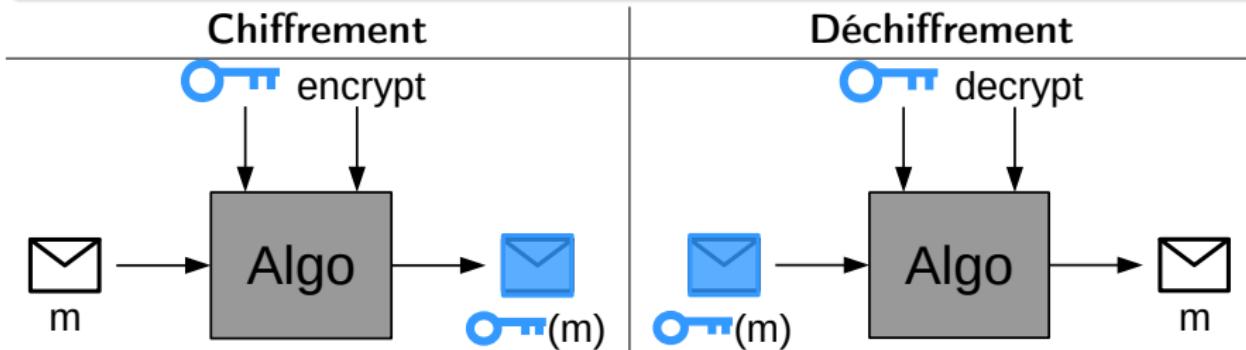
Terminologie de la cryptographie

- **Chiffrement** : transformation de données en clair en des données chiffrées afin qu'il soit difficile voire impossible à exploiter sans effectuer l'opération de déchiffrement
- **Déchiffrement** : opération inverse qui permet le retour des données en clair à partir de données chiffrées
- **Cryptogramme** : un message chiffré

Opérations de chiffrement et déchiffrement

Algorithme cryptographique

- Algorithme décrivant la transformation de chiffrement/déchiffrement.
- En entrée :
 - Une **clé de chiffrement** : paramètre secret de l'algorithme qui définit la transformation (suite d'octets)
 - Un **message** à transformer (suite d'octets)
 - Un **sens** : chiffrage ou déchiffrage (booléen ou enum)
- En sortie : le message chiffré/déchiffré



Opérations de chiffrement et déchiffrement

Comment évaluer l'efficacité de ces opérations ?

Une opération de chiffrement idéale rend impossible le déchiffrage d'un message sans connaître au préalable la clé

Comment assurer l'efficacité de ces opérations ?

- Confidentialité des clés : ne pas les diffuser/les divulguer
- Taille de la clé : plus la clé est grande, plus il est difficile de la trouver par brute force
- Avoir un algorithme difficile à casser :
Cassage = inverser le chiffrement de l'algorithme sans la clé
- Connaître l'existence de backdoor ou des failles de l'implantation de l'algorithme
accès au code source nécessaire ?

Chiffrement symétrique

Principes

- Une seule **clé secrète** est utilisée pour le chiffrement/ déchiffrement
- L'émetteur et le récepteur doivent disposer de la même clé pour déchiffrer et déchiffrer
- Une clé symétrique nommée \mathcal{K} assure la propriété suivante :

$$\mathcal{K}(\mathcal{K}(m)) = m$$

- Les algorithmes se basent sur une suite substitution monoalphabétique et permutation.

Exemple de chiffrement symétrique (simple)

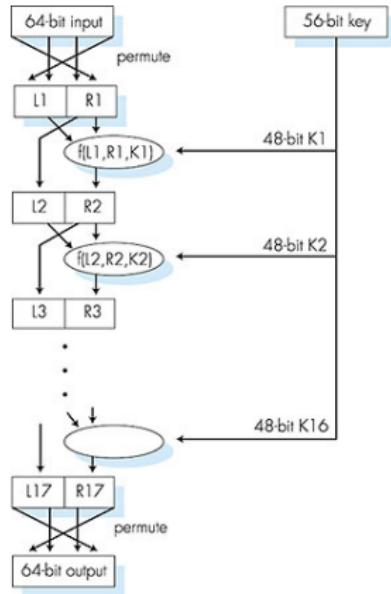
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

clé $\Rightarrow 11$, chiffrement $\Rightarrow + \text{ mod } 26$, déchiffrement $\Rightarrow - \text{ mod } 26$

Chiffrement symétrique : algorithmes

DES (Data Encryption Standard)

- Algorithme historique
- Cryptage par bloc de 64 bits
- clé de 56 bits
- 16 étages de transformations et de permutation
- obsolète car il existe des algorithmes efficaces de cassage



Algorithmes utilisés de nos jours

- **3-DES** : Utiliser 3 clés successivement sur chaque donnée
- **AES (Advanced Encryption Standard)**

Données par blocs de 128 bits , Clés de 128, 192 ou 256 bits

Avantages

- ✓ Les transformations sont peu coûteuses en ressources de calcul (temps CPU et Mémoire)
- ✓ Difficile à déchiffrer même si la clé est relativement petite

Limites

- ✗ L'échange de la clé entre le récepteur et l'émetteur peut être une faille de sécurité
- ✗ Il faut une clé distincte pour chaque paire d'interlocuteurs ce qui peut rendre le nombre de clés à gérer important.

Chiffrement asymétrique (ou à clé publique)

Principes

- Chaque utilisateur possède une paire de clés qui sont liées entre elles :
 - une **clé privée** \mathcal{K}^- qui ne doit être connue que de son propriétaire
 - une **clé publique** \mathcal{K}^+ qui peut être connue de tout le monde
- Il doit être impossible de calculer \mathcal{K}^- à partir de \mathcal{K}^+
- Il faut respecter la propriété suivante

$$\mathcal{K}^-(\mathcal{K}^+(m)) = m = \mathcal{K}^+(\mathcal{K}^-(m))$$

⇒ À partir d'un message chiffré avec \mathcal{K}^- , on le déchiffre avec \mathcal{K}^+

⇒ À partir d'un message chiffré avec \mathcal{K}^+ , on le déchiffre avec \mathcal{K}^-

⇒ Les deux clés ne sont pas symétriques individuellement :

$$\mathcal{K}^+(\mathcal{K}^+(m)) \neq m \text{ et } \mathcal{K}^-(\mathcal{K}^-(m)) \neq m$$

- Se base sur la factorisation de grands nombres qui est un problème mathématique difficile à résoudre

Comment choisir les clés ?

- 1) Choisir 2 très grands nombres premiers p, q (1024 bits chacun)
- 2) Calculer $n = pq$ et $z = (p - 1)(q - 1)$
- 3) Choisir e (avec $e < n$) n'ayant aucun facteur commun avec z . (e et z sont premiers entre eux).
- 4) Choisir d tel que $ed - 1$ soit divisible par z ($ed \bmod z = 1$).

$$\mathcal{K}^+ = (n, e) \text{ et } \mathcal{K}^- = (n, d)$$

Chiffrement/Déchiffrement d'un message m

$$\mathcal{K}^+(m) = m^e \bmod n \text{ et } \mathcal{K}^-(m) = m^d \bmod n$$

Exemple : $p = 5, q = 7 \Rightarrow n = 35, z = 24, e = 5, d = 29$ et $m=12$

$$\mathcal{K}^+(m) : m^e = 12^5 \bmod 35 = 17$$

$$\mathcal{K}^-(m) : m^d = 12^{29} \bmod 35 = 17$$

Chiffrement symétrique

Avantages

- ✓ La gestion des clés est plus facile :
 - en symétrique : si N utilisateurs alors il faut $n(n-1)/2$ clés
 - en asymétrique : si N utilisateurs alors il faut N paires de clés
- ✓ L'échange de la clé publique n'a pas besoin de transiter via un canal sécurisé

Limites

- ✗ Les tailles des clés doivent être grandes pour avoir un niveau de sécurité satisfaisant
 - une clé symétrique de 128 bits a le même niveau de sécurité qu'une clé asymétrique de 2048 bits
- ✗ Les algorithmes de chiffrement sont coûteux en ressource
 - grand nombre \Rightarrow plusieurs milliers de bits
- ✗ Il faut s'assurer de l'authenticité de la clé publique

Les limites de la cryptographie

La vulnérabilité des algorithmes

- Certains algorithmes sans failles connues ne sont plus utilisés car la puissance de calcul des machines actuelles permet de trouver leur clé
- La vulnérabilité d'un algorithme dépend du temps et de la puissance du hardware

Ce que ne permet pas la cryptographie

- Empêcher l'effacement des données par un pirate
- Protéger le programme de chiffrement et son exécution
- Empêcher un déchiffrement par hasard
- Empêcher une attaque par brute force
- Empêcher la lecture avant chiffrement ou après déchiffrement

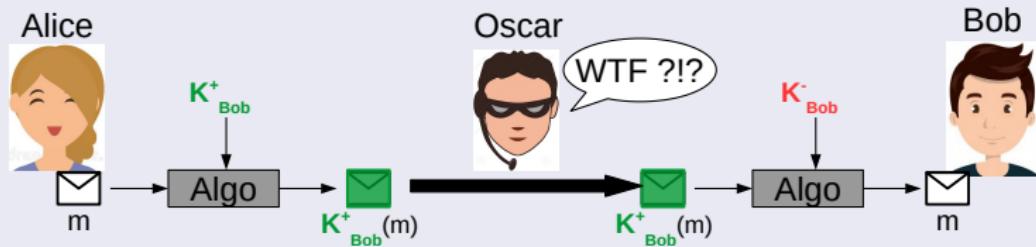
Assurer la confidentialité

Avec clé symétrique

- **Hypothèse** : la clé \mathcal{K} est exclusivement partagée entre Alice et Bob
- Alice chiffre son message avec \mathcal{K} , Bob déchiffre avec \mathcal{K}
- Oscar n'ayant pas la clé, il ne peut pas déchiffrer le message

Avec clé publique

- **Hypothèse** : Alice possède l'authentique clé publique de Bob
- Alice chiffre le message avec \mathcal{K}^+_{Bob} , Bob déchiffre avec \mathcal{K}^-_{Bob}
- Oscar n'ayant pas \mathcal{K}^-_{Bob} , il ne peut pas déchiffrer le message



Assurer l'authentification

Objectif : Bob veut que Alice lui "prouve" son identité

Solution 1

Alice envoie le message "Je suis Alice"



Échec de la solution 1

Oscar peut se faire simplement passer pour Alice puisque Bob ne peut pas "voir" réellement qui est l'émetteur

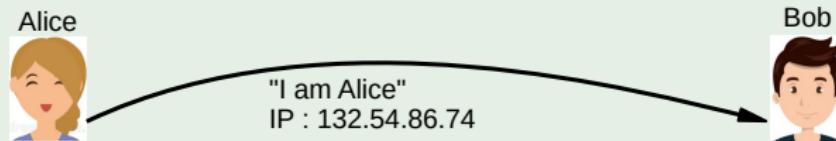


Assurer l'authentification

Objectif : Bob veut que Alice lui "prouve" son identité

Solution 2

Alice envoie le message "Je suis Alice" + son adresse IP



Échec de la solution 2

Oscar peut spoofe l'adresse IP de Alice



Assurer l'authentification

Solution 3

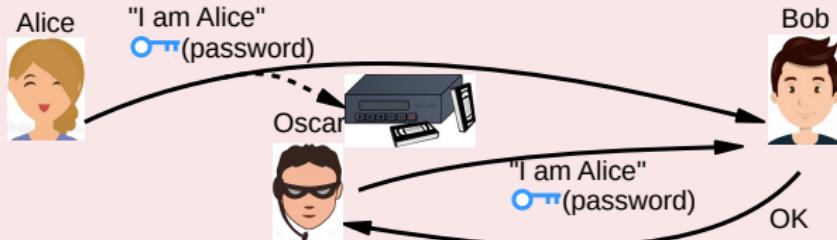
- Alice envoie le message "Je suis Alice" + son mot de passe chiffré
- Bob déchiffre le mot de passe, vérifie et renvoie une confirmation



Échec de la solution 3

Oscar peut faire une **attaque par rejetu**

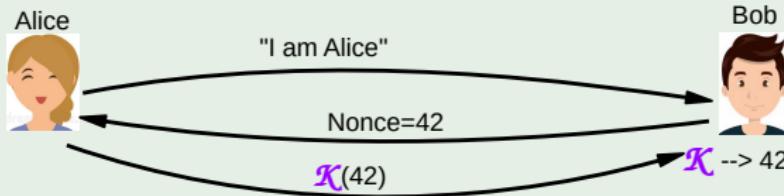
⇒ il enregistre le message de Alice et le renvoie plus tard à Bob



Assurer l'authentification

Une bonne solution avec clé secrète

- **Hypothèse** : Alice et Bob partagent une clé secrète symétrique \mathcal{K}
- Protocole :
 - Alice envoie le message "Je suis Alice"
 - Bob tire un nombre aléatoire et le renvoie à Alice (= **nonce**)
 - Alice renvoie le nonce chiffré avec \mathcal{K}
 - Bob déchiffre le nonce avec \mathcal{K}
- Oscar ne peut plus rejouer le message lors d'une prochaine connexion car le nonce change à chaque fois et il ne possède pas \mathcal{K}

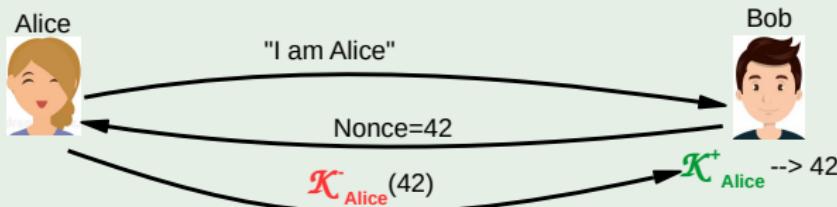


Note : le Nonce peut également être chiffré par Bob avec \mathcal{K} , pour que Alice authentifie Bob.

Assurer l'authentification

Une bonne solution avec clé publique

- Hypothèse : Bob possède l'authentique clé publique de Alice
- Protocole :
 - Alice envoie le message "Je suis Alice"
 - Bob tire un nombre aléatoire et le renvoie à Alice (= **nonce**)
 - Alice renvoie le nonce chiffré avec sa clé privée K^-_{Alice}
 - Bob déchiffre le nonce avec la clé publique de Alice K^+_{Alice}
- Oscar ne possède pas K^-_{Alice} pour répondre correctement à Bob



Note : le Nonce peut également être chiffré par Bob avec K^-_{Bob} , pour que Alice authentifie Bob.

Signature numérique

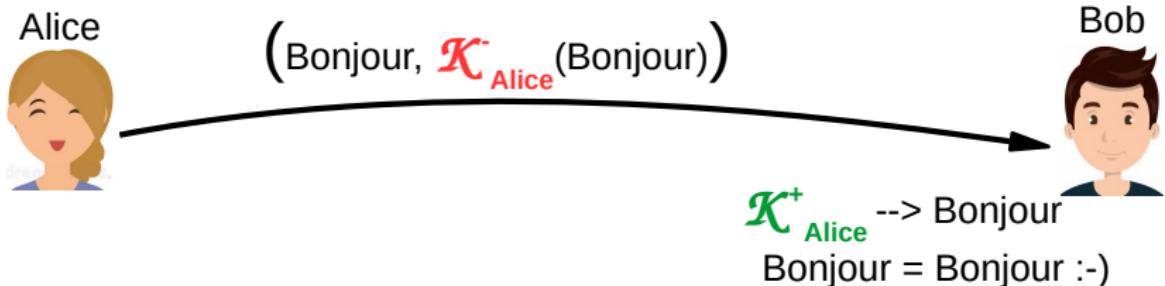
- Mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur.
- Alice doit envoyer un message à Bob qui doit pouvoir contrôler que :
 - I) le message a bien été créé par Alice
 - II) le message n'a pas été altéré par Oscar

Solution

- pour I), on a les clés de chiffrement
 - en clé symétrique : Alice chiffre le message avec K_{A-B}
 - en clé publique : Alice chiffre le message avec K^-_{Alice}
- Pour II), Bob a besoin d'avoir un point de comparaison
 - Alice englobe dans son envoi : le message en clair **et** le message chiffré
 - Bob déchiffre la partie chiffrée et vérifie la correspondance

Si Oscar modifie la partie en clair, Bob pourra le détecter

Vers l'assurance de l'intégrité



Inconvénient non négligeable de cette solution

Les messages peuvent être en pratique de taille conséquente :

- La taille des communications est potentiellement doublée
⇒ surcoût bande passante
- La vérification nécessite un déchiffrement systématique
⇒ surcoût temps CPU et RAM

Fonction de hachage

Définition

Fonction notée \mathcal{H} qui pour un objet de taille quelconque produit une **empreinte** de taille fixe.

Propriétés d'une fonction de hachage sécurisé

- \mathcal{H} doit être à sens unique \Rightarrow il est quasi-impossible de déduire \mathcal{H}^{-1} .
- \mathcal{H} est à collision faible difficile
 - \Rightarrow Pour y donné, il est difficile de trouver x tel que $\mathcal{H}(x) = y$
- \mathcal{H} est à collision forte difficile
 - \Rightarrow Il est difficile de trouver x_1 et x_2 tel que $\mathcal{H}(x_1) = \mathcal{H}(x_2)$

Informations complémentaires

- Les algorithmes de hachage sont publics (MD5, SHA-2, CRC32 ...)
- Synonymes de *empreinte* : *haché*, *digest*, *condensé*, *résumé*

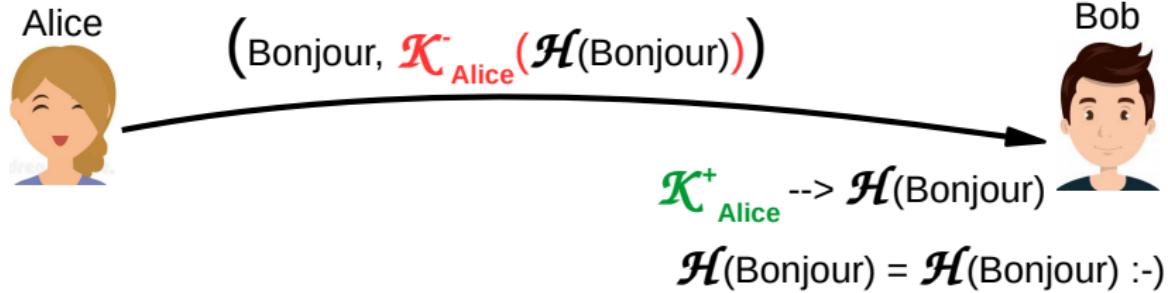
Assurer l'intégrité

Apport d'une fonction de hachage sécurisée

Elle permet de quasi-identifier un message à coût constant et sans connaître son contenu.

Retour à l'exemple

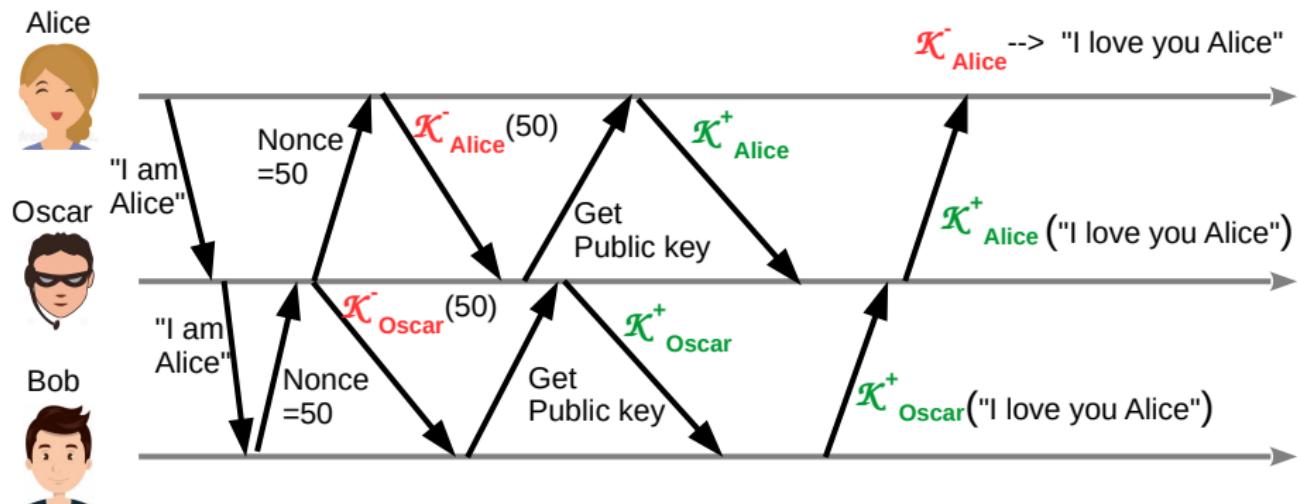
- Alice doit donc signer son message m avec $\mathcal{K}^-_{Alice}(\mathcal{H}(m))$
- \mathcal{H} et \mathcal{K}^+_{Alice} étant connus de tout le monde (y compris Oscar) n'importe qui peut vérifier l'authenticité du message



Risque de la non-authenticité des clés publiques

L'attaque Man in the Middle

- Oscar fait passer sa clé publique \mathcal{K}^+_{Oscar} comme étant la clé publique de Bob (resp. Alice) aux yeux d'Alice (resp. de Bob)
 - Arrive quand on n'a pas authentifié l'entité qui envoie la clé publique
- ⇒ **Authentification et confidentialité violées**



Nestor : un nouveau protagoniste



- Entité qui est supposée ne pas être corrompue
- Son identité et sa clé publique sont connues et reconnues de tout le monde
- Il peut générer et distribuer des clés secrètes pour deux entités qui souhaitent communiquer
 ⇒ rôle d'un **Key Distribution Center (KDC)**
- Il peut certifier ou révoquer le lien entre une entité et sa clé publique
 ⇒ rôle d'un **Certification Authority (CA)**

Enregistrer sa clé publique

- Bob (ou Alice) fournit une "preuve d'identité" à Nestor
- Nestor fait un lien entre Bob et \mathcal{K}^+_{Bob}
- Nestor peut délivrer un certificat sur l'identité de Bob en **signant numériquement** le lien $Bob \leftrightarrow \mathcal{K}^+_{Bob}$

Informations contenues dans un certificat

- La clé publique de l'entité
- Information sur l'entité : nom, pays, algo de chiffrement
- Période de validité du certificat
- Information sur l'autorité de certification
- **signature numérique du CA**

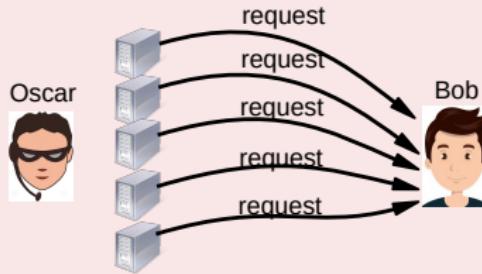
Assurer la disponibilité

Risque : attaque Deny Of Services (DOS)

- Objectif de Oscar : Générer un nombre important de requêtes sur Alice ou Bob pour que ces derniers ne puissent plus remplir correctement leur service



- Version distribué(D-DOS) : Oscar a corrompu des entités zombies pour qu'elles inondent de requête Alice ou Bob



Assurer la disponibilité

Mise en place de machines contrôle (firewall)

- Permet de contrôler les communications :
 - Qui s'adresse à qui ?
 - À quelle fréquence ?
 - Sur quels ports ?
 - ...
- Peut bannir un attaquant potentiel pour protéger son système

