Introduction

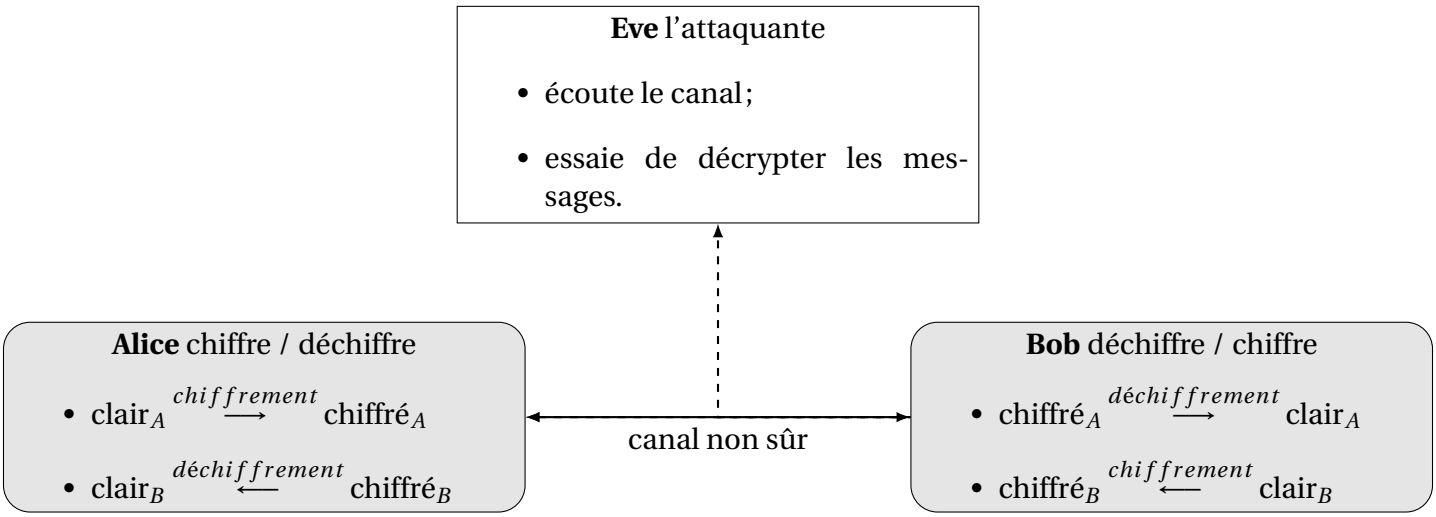
Comment sécuriser les échanges entre client et serveur sur le Web ? La cryptographie étudiée depuis des siècles pour protéger la correspondance secrète des états et des militaires, est désormais utilisée par tous lors d'une connexion en HTTPS sur un serveur Web.
On présentera les grands principes des méthodes de chiffrement et on découvrira que la sûreté des communications dépend de la difficulté de certains problèmes mathématiques.

Sources :

- « *Manuel de Terminale NSI* » de Michel Baudouin-Lafon aux éditions Hachette. et « *Manuel de Terminale NSI* » de T. Balabonski et coauteurs aux éditions Ellipse.
- « *Documents ressources Eduscol* »
- « *Cours de Gilles Lassus* » https://glassus.github.io/terminale_nsi
- *Ressources du concours Al Kindi* https://www.concours-alkindi.fr/#/pageDiscover.

1 Enjeux et histoire de la cryptographie

1.1 Enjeux et vocabulaire



Graphique 1 : situation problème

Point de cours 1

- **Situation :** Alice veut transmettre à Bob un message dont le contenu doit rester **secret** pour Eve qui écoute le canal de communication.

  Trois problèmes se posent :

  ☞ **Problème 1 :** garantir la **confidentialité** : le contenu du message même s'il est intercepté doit rester secret pour Eve, seul Bob doit pouvoir déchiffrer le message d'Alice ;

  ☞ **Problème 2 :** garantir l' **authentification** de l'émetteur : Bob doit pouvoir s'assurer que l'émet-

teur du message reçu est bien Alice. Un problème corrélé est la garantie de **l'intégrité** du message (qu'il n'a pas été modifié).

☞ **Problème 3 : l'efficacité** des méthodes pour dissimuler/révéler le message secret peut être critique si le volume d'informations à traiter est important (serveur, flux audio/vidéo).

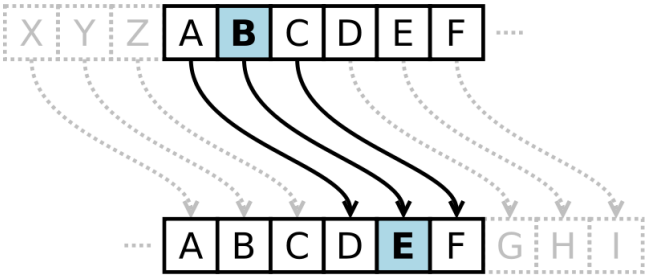
- Alice et Bob ont d'abord envisagé la **stéganographie** : cacher le message dans un contenant. Le codage binaire étant universel, ils ont remplacé les 4 bits de poids faibles d'un pixel par 4 bits d'un caractère pour cacher le texte de leur message dans une image. Malheureusement Eve a intercepté l'image et après quelques minutes de recherche découvert le procédé et révélé le message.
- Après quelques recherches sur Wikipedia, ils ont découvert les principes de Kerckhoffs énoncés pour garantir la confidentialité des communications en période de guerre :
  - *il faut que le système n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;*
  - *la sécurité du système doit reposer non pas sur le secret du procédé qui doit pouvoir être renseigné au grand jour mais sur le secret d'une clef qui doit pouvoir être partagée facilement entre l'émetteur et le destinataire.*

  Ces principes fondent la **cryptographie** moderne c'est-à-dire la science de la sécurisation des communications, en particulier le fait que la sécurité d'une méthode cryptographique repose sur un paramètre secret appelé **clef**.
- Avant d'aller plus loin, Alice et Bob décident de fixer précisément les définitions :
  - ☞ **Coder** un message consiste à le représenter dans un alphabet de symboles, tel que tout le monde peut **décoder** le message à l'aide d'un algorithme **public**. On peut citer le codage binaire d'un entier ou le codage Unicode pour un caractère.
  - ☞ **Chiffrer** un message consiste à le représenter dans un alphabet de symboles, tel que seul une personne possédant une information secrète, la **clef** peut **déchiffrer** le message chiffré. Selon les principes de Kerckhoffs, l'algorithme de déchiffrement peut être public mais ne peut fonctionner qu'avec la connaissance d'un paramètre secret, la **clef**.
  - ☞ La **cryptanalyse** est la science qui a pour but de **décrypter** un message chiffré sans posséder la clef de chiffrement.

Objectif

Dans les parties suivantes nous étudierons les deux principales méthodes de chiffrement, symétrique et asymétrique et leur application à la sécurisation des communications sur le Web à travers le protocole `HTTPS`.

1.2 Cryptographie de l'antiquité à nos jours



Graphique 2 : le chiffrement de César

Histoire 1

1. Jules César chiffrait ses messages secrets avec la méthode suivante :

- *on recopie chaque caractère du message en clair qui n'est pas une lettre majuscule ;*
- *chaque lettre majuscule est remplacée par la lettre située* `clef` *places plus loin dans l'ordre alphabétique, en recommençant par 'A' si le décalage dépasse 'Z'.*

a. Chiffrer le message `'AVE CESAR'` avec un chiffrement de César de clef 4.

'EZI GIMEV'

b. Déchiffrer le message `"RJXXFLJ XJHWJY"` sachant qu'il a été chiffré avec un chiffrement de César de clef 5.

'MESSAGE SECRET'

c. Combien existe-t-il de clefs pour un chiffrement de César ? Que se passe-t-il si la clef est 13 ?

25 clefs si on enlève le décalage de 0
Si la clef est 13, alors la clef de déchiffrement est la même que la clef de chiffrement

2. Al-Kindi (801-8773) a écrit l'un des premiers ouvrages connus de cryptanalyse, le « *Manuscrit sur le déchiffrement des messages codés* ». Quel type d'attaque a-t-il utilisé pour décrypter des messages chiffrés et à quelle catégorie de méthodes de chiffrement s'applique-t-elle ?

> **Qui était Al Kindi ?**
>
> Al Kindi est un savant arabe du IXe siècle qui s'est intéressé à de nombreuses sciences, allant de la géométrie à la médecine et à la chimie. Dans le « Manuscrit sur le chiffrement des messages cryptographiques », il explique comment casser les meilleurs codes connus à son époque, à l'aide de la technique de l'analyse de fréquence. C'est la première trace connue de cryptanalyse. Par conséquent, il est considéré comme l'un des fondateurs de la discipline.

3. A l'aide de recherches sur le Web ou dans les pages de cours du manuel (pages 200 à 204) ; localiser dans le temps les différents méthodes de chiffrement énumérées dans le tableau ci-après.

4. Une attaque exhaustive du système de chiffrement DES, qui utilise une clef secrète de 56 bits, a été réalisée en janvier 1998 en 39 jours sur 10 000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée comportant 1500 composants DES.

Combien de temps aurait-il fallu pour casser le système si la taille de la clef secrète avait-été de 57 bits ? de 64 bits ?

2 fois plus de temps pour une clef de 57 bits
$2^{64-56} = 2^8$ fois plus de temps pour une clef de 64 bits

5. La sécurité des chiffrements asymétriques repose sur la difficulté de certains problèmes mathématiques (logarithme discret, factorisation d'entiers). Des attaques à l'aide de méthodes mathématiques sont beaucoup plus efficaces qu'une recherche exhaustive. Le temps de recherche d'une clef de 65 bits n'est donc est donc pas doublé par rapport à une clef de 64 bits.

Quelles sont les recommandations de l'ECRYPT https://www.keylength.com/en/3/ pour les tailles de clefs de chiffrements asymétriques RSA (*Factoring modulus*) et Diffie-Hellman (*Discrete logarithm key and group*) ?

Protection	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
Legacy standard level *Should not be used in new systems*	80	1024	160	1024	160	160
Near term protection *Security for at least ten years (2023-2028)*	128	3072	256	3072	256	256
Long-term protection *Security for thirty to fifty years (2023-2068)*	256	15360	512	15360	512	512

Méthode de chiffrement	Epoque / date	Type de clef	Symétrique / Asymétrique	Sûreté	Efficacité	Utilisation HTTPS
Scytale	…	clef physique (bâton)	symétrique	non sûr	Rapide	Non
César	…	clef numérique (décalage alphabétique)	symétrique	non sûr	Rapide	Non
Vigenère	…	décalages variables	symétrique	non sûr	Rapide	None
Vernam	…	XOR avec clef aléatoire de la longueur du message	symétrique	incassable	clef jetable de la taille du message, une seule utilisation	Non
Enigma (Allemagne nazie)	…	caractères	symétrique	cassé par les Anglais à Bletchey Park (Turing)	rapide, avec machine	Non
DES	…	numérique (56 bits)	symétrique	cassé	rapide	Non
Diffie-Hellman	…	paire de clefs (publique, privée), numérique (… … bits)	asymétrique	sûr pour une clef assez longue (problème du logarithme discret)	calculs coûteux (exponentiation modulaire)	Oui échange de clefs
RSA	…	paire de clefs (publique, privée), numérique (… … bits)	asymétrique	sûr pour une clef assez longue (problème de la factorisation en facteurs premiers)	calculs coûteux (exponentiation modulaire)	Oui échange de clefs et authentification
AES	…	numérique (128 bits)	symétrique	sûr pour une clef assez longue	rapide	Oui chiffrement des données

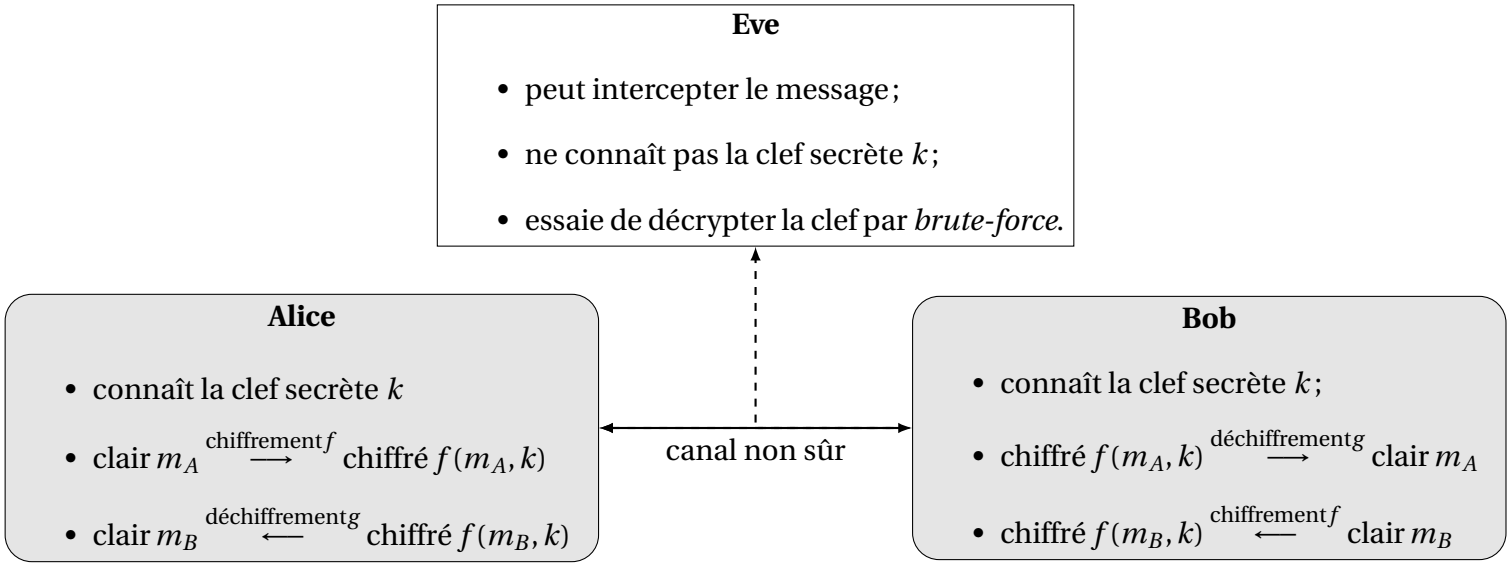
2 Chiffrement symétrique

2.1 Principe

Point de cours 2

- Un **chiffrement symétrique** a pour but de garantir la confidentialité d'un message transmis par un émetteur Alice à un destinataire Bob sur un canal non sécurisé écouté par Eve.
- Un chiffrement **symétrique** nécessite trois éléments :
  - ☞ une **clef secrète de chiffrement** k partagée symétriquement par l'émetteur et le destinataire ;
  - ☞ une *fonction de chiffrement* f qui prend en paramètre le message en clair m et la clef k et renvoie un chiffré $f(m,k)$.
  - ☞ une *fonction de déchiffrement* g qui prend en paramètre le message chiffré $f(m,k)$ et la clef k et renvoie le message en clair $g(f(m,k),k) = m$.

    Souvent les fonctions de chiffrement et déchiffrement sont les mêmes (chiffrement XOR) ou quasi-identiques (Cesar).



Graphique 3 : chiffrement symétrique

Analogie 1 *Chiffrement symétrique et coffre fort*

On peut faire l'analogie entre le chiffrement symétrique et l'utilisation d'un *coffre fort.*

- *Un coffre fort est verrouillé par une serrure dont la* ***clef secrète*** *est possédée uniquement par Alice et Bob.*
- *Pour transmettre un message secret à Bob, Alice le dépose dans le coffre fort et le verrouille*

*avec sa **clef secrète**.*

- *Eve ne possède pas la clef et ne peut donc lire le message même si elle intercepte le coffre.*
- *À réception du coffre, Bob le déverrouille avec la **clef secrète** et peut lire le message.*

2.2 Chiffrements de César et Vigenère

Exercice 1

Faire l'activité pages 198 et 199 du manuel dans le carnet Capytale :
https://capytale2.ac-paris.fr/web/c/3ed5-1533001.

2.3 Chiffrement XOR et de Vernam

Exercice 2 *Voir TP …*

Nous avons déjà étudié en TP le **chiffrement XOR**. On superpose le message en clair et la clef éventuellement répétée. On remplace les caractères par leurs ordinaux Unicode et on applique l'opérateur logique XOR bit à bit aux paires (ordinal clair, ordinal clef) :

```
clair XOR clef = chiffré
```

Par propriété de l'opérateur XOR, le chiffrement XOR est *symétrique* :

```
chiffré XOR clef = (clair XOR clef ) XOR clef = clair
```

Exemple de chiffrement/déchiffrement de `'PYTHON'` avec la clef `'NSI'` :

```
>>> clair = [ord(c) for c in 'PYTHON']
>>> clair
[80, 89, 84, 72, 79, 78]
>>> clef = [ord(c) for c in 'NSINSI']
>>> clef
[78, 83, 73, 78, 83, 73]
>>> chiffre = [clair[k] ^ clef[k] for k in range(len(clair))]
>>> chiffre
[30, 10, 29, 6, 28, 7]
>>> dechiffre = [chiffre[k] ^ clef[k] for k in range(len(chiffre))]
[80, 89, 84, 72, 79, 78]
```

Le **chiffrement de Vernam** est un chiffrement XOR qui doit vérifier les trois conditions suivantes :

- (C1) la clef a la même longueur que le message ;
- (C2) la clef est aléatoire ;
- (C3) la clef n'est utilisée qu'une seule fois (masque jetable).

1. Compléter les fonctions ci-dessous dans le carnet Capytale :
   https://capytale2.ac-paris.fr/web/c/3ed5-1533001

```python
import random

def masque_jetable(long:int)->list[int]:
    """Renvoie un masque jetable, tableau d'entiers aléatoires
    entre 0 et 255  de longueur long"""
    return [random.randint(0, 255) for _ in range(long)]

def chiffre_vernam(clair:list[int], masque:list[int])->list[int]:
    """Renvoie le chiffre XOR de Vernam sur les tableaux d'entiers
    clair et clef de même longueur"""
    assert len(clair) == len(masque)
    return [clair[k]^masque[k] for k in range(len(clair))]
```

2. On donne ci-dessous les étapes d'un chiffrement de Vernam du message clair `'VERNAM'`.

```
>>> clair_vernam = [ord(c) for c in "VERNAM"]
>>> clair_vernam
[86, 69, 82, 78, 65, 77]
>>> masque = masque_jetable(len(clair_vernam))
>>> masque
[117, 113, 99, 133, 16, 179]
>>> chiffre = chiffre_vernam(clair_vernam, masque)
>>> chiffre
[35, 52, 49, 203, 81, 254]
```

a. Construire une clef qui permet d'obtenir le même chiffré à partir du message clair `'PYTHON'`.

```python
clair_vernam = [ord(c) for c in "VERNAM"]
masque1 = masque_jetable(len(clair_vernam))
chiffre1 = chiffre_vernam(clair_vernam, masque1)
clair_python =  [ord(c) for c in "PYTHON"]
masque2 = chiffre_vernam(chiffre1, clair_python)
chiffre2 = chiffre_vernam(clair_python, masque2)
assert chiffre2 == chiffre1
```

b. **Claude Shannon** a démontré que le chiffrement de Vernam, parce qu'il vérifie les trois conditions précitées, est *inconditionnellement sûr* c'est-à-dire inattaquable par recherche exhaustive de la clef. Expliquer cette affirmation à l'aide de l'exemple précédent.

Par recherche brute on peut obtenir plusieurs messages clairs cohérents pour le même chiffré selon la clef choisie. Le choix de la clef étant aléatoire il n'est pas possible de déterminer le message clair.

2.4 Sécurité d'un chiffrement symétrique

Méthode *Attaque brute-force et taille d'une clef*

- Pour décrypter la clef d'un chiffrement symétrique à partir d'un chiffré, il est possible de procéder par recherche exhaustive de la clef. Sur n bits on peut coder 2^n clefs et par attaque *brute-force* il faut tester en moyenne la moitié des clefs possibles soit 2^{n-1}.

- Un algorithme de chiffrement symétrique est considéré comme robuste si on ne connaît pas de meilleure attaque que la recherche exhaustive.
- Actuellement, on considère qu'il faut au moins 2^{127} essais en moyenne à un attaquant pour qu'une clef soit sûre, donc on recommande des clefs d'au moins 128 bits pour les chiffrements symétriques.
- Pour chiffrer les échanges d'information dans le protocole `HTTPS`, l'un des chiffrements symétriques les plus courants est `AES` (`Advance Encryption Standard`) qui opère de façon itérée sur des blocs de 128 bits avec des clefs de taille 128 ou 256 bits .

  Le chiffrement AES utilise uniquement des opérations élémentaires sur les bits (permutation, XOR), il est donc très efficace pour chiffrer rapidement d'où son utilisation dans le protocole HTTPS.

Exercice 3 ***Recherche exhaustive de collisions***

Faire l'exercice 19 p. 213.

3 Chiffrement asymétrique

3.1 Enjeux

Exercice 4 ***Gestion des clefs de chiffrement symétrique***

1. Dans un groupe de 4 personnes, combien faut-il générer de clefs symétriques si on souhaite établir des communications confidentielles pour chaque paire d'individus (une clef ne doit pas être partagée entre plus de 2 personnes) ?

   $\frac{4\times3}{2} = 6$

   Et si on a n personnes dans le groupe ?

   $\binom{n}{2} = \frac{n(n-1)}{2}$ complexité quadratique en le nombre n de personnes

2. Les n personnes du groupe ne se sont jamais rencontrées et ne pourront pas se rencontrer (éloignement géographique, situation de guerre). Quel(s) problème(s) se pose dans la gestion des clefs ?

   Comment distribuer / modifier une paire de clefs sans que les deux entrémités du canal de communication se rencontrent ?

3.2 Un article fondateur

Correction des exercices 18 et 19 p. 213.

18 Fonctions de hachage

Une fonction de hachage est un algorithme qui prend en entrée une chaîne de caractères de longueur arbitraire et qui retourne en sortie une chaîne de caractères de longueur n fixée, appelée empreinte. Deux applications fréquentes sont la vérification de l'intégrité d'un fichier après un téléchargement et le stockage d'une empreinte des mots de passe sur un serveur (sans stocker les mots de passe en clair). Trois propriétés de sécurité classiques demandées aux fonctions de hachage sont les suivantes :

- résistance aux collisions : il est difficile de trouver deux messages m_1 et m_2 différents tels que $h(m_1) = h(m_2)$;
- résistance à la seconde préimage : il est difficile, étant donné m_1, de trouver m_2 différent de m_1 tel que $h(m_1) = h(m_2)$;
- résistance à la préimage : il est difficile, étant donné x, de trouver m tel que $h(m) = x$.

a. Expliquer pourquoi la première propriété (souvent considérée comme étant la plus forte) et la troisième sont importantes dans le cadre des deux applications évoquées.

b. Montrer qu'une fonction de hachage résistante aux collisions est également résistante à la seconde préimage. On pourra raisonner par contraposée en prouvant qu'un attaquant qui peut casser la seconde préimage peut également trouver des collisions.

a) Résistance aux collisions :
- 2 fichiers ≠ auront des empreintes ≠
- 2 mots de passe ≠ auront des empreintes ≠

Résistance à la préimage :
- Ne pas pouvoir reconstruire un fichier à partir de son empreinte (protection code source)
- Ne pas pouvoir déchiffrer un mot de passe à partir de son empreinte.

b) Résistance aux collisions ⇒ Seconde préimage (moins forte)

par contraposition car :

Non (Seconde préimage) ⇒ Non (Résistance aux collisions)

---

19 Recherches exhaustive et de collisions

On considère qu'un protocole est sûr s'il nécessite au moins 2^{128} opérations de la part d'un attaquant.

a. Pour les algorithmes de chiffrement symétrique, on espère que la meilleure attaque possible est l'attaque par recherche exhaustive. Expliquer ce qu'est une attaque par recherche exhaustive. On teste toutes les clefs possible

b. Justifier que la plupart des algorithmes de chiffrement symétriques considérés sûrs ont des tailles de clefs de plus de 128 bits.

c. Pour les algorithmes de fonction de hachage, on espère que la meilleure attaque possible pour la recherche de collisions est celle reliée au problème appelé *paradoxe des anniversaires*, qui a une complexité en $\sqrt{2^n}$, où n est la taille de la clef.

b) Sur 128 bits on a 2^{128} clefs. Si la clef a été choisie aléatoirement et qu'on les teste dans l'ordre croissant on a 1 chance sur deux que la clef soit dans la première moitié (2^{127} clefs) donc il faut 2^{127} tests en moyenne pour trouver la bonne clef. Il faut donc une clef

de plus de 128 bits. Avec 129 bits il faut $\frac{2^{129}}{2} = 2^{128}$ tests pour trouver la clef.

c)

En se reportant aux définitions de l'exercice **18**, expliquer ce qu'est une recherche de collisions.

d. Expliquer pourquoi la plupart des algorithmes de fonctions de hachage considérés sûrs ont des tailles de clefs de plus de 256 bits.

$$\sqrt{2^n} \geqslant 2^{128} \iff 2^n \geqslant 2^{128 \times 2}$$

$$\iff n \geqslant 256$$

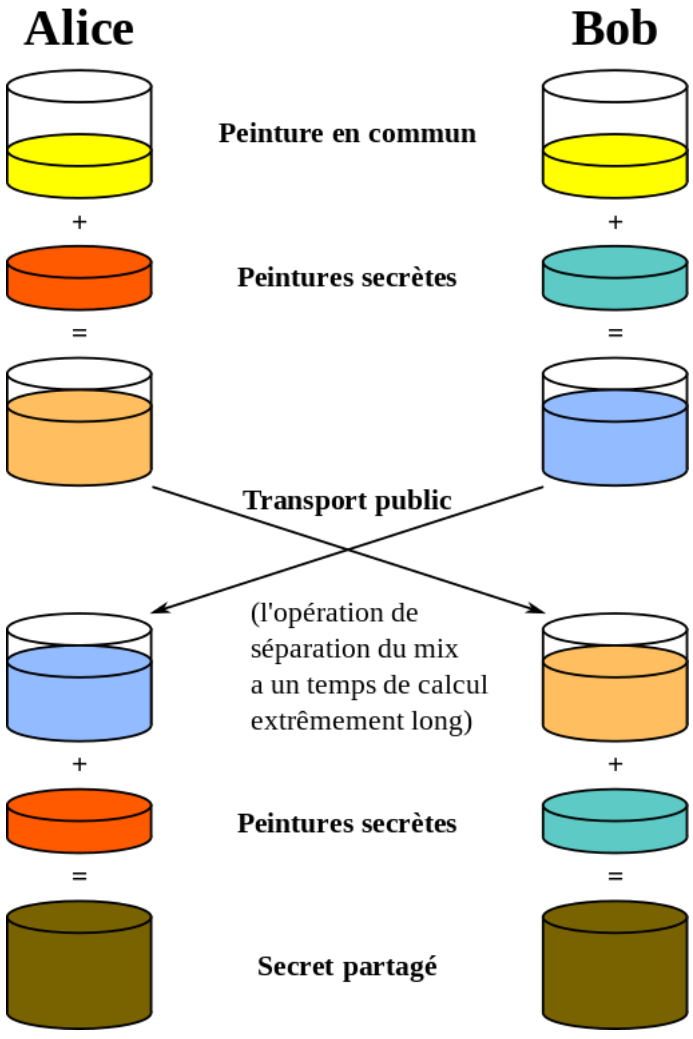
Histoire 2

En 1976, dans l'article « *New directions in cryptography* », **Whitfield Diffie** et **Martin Hellman**, introduisent le concept révolutionnaire de chiffrement asymétrique à clef publique . Même si les auteurs ne donnent pas de réalisation pratique d'un tel système, ses propriétés sont clairement énoncées : résolution du problème de l'échange de clefs de chiffrement symétrique sans secret partagé et possibilité de signature numérique infalsifiable.
En outre, ils présentent un protocole par lequel deux entités peuvent convenir d'une clef secrète à partir de la connaissance préalable de seules donnéees publiques. La première réalisation d'un chiffrement à clef publique est due à **Ronald Rivest**, **Adi Shamir** et **Leonard Adleman**, en 1978 : c'est le chiffrement RSA.
Diffie et Hellman ont reçu le prix Turing, plus haute distinction en informatique, en 2015, et Rivest, Shamir, Adleman en 2002.

3.3 Protocole d'échange de clef de Diffie-Hellman

Point de cours 3

Le **protocole d'échange de clef Diffie-Hellman** permet à deux utilisateurs, Alice et Bob, qui doivent communiquer sur un canal non sûr, d'échanger de façon confidentielle une clef secrète de chiffrement symétrique à partir de données publiques.
On peut expliquer le protocole par analogie avec des pots de peinture.



Graphique 4 : source Wikipedia

- **Étape 1 :** Alice et Bob conviennent d'une couleur de peinture commune, ici le jaune. Cette couleur est connue de tous, y compris d'Eve qui écoute.
- **Étape 2 :** Alice choisit une autre couleur secrète (ici du rouge). Elle mélange la peinture commune et sa couleur secrète et obtient de l'orange. Alice envoie la couleur orange à Bob. La couleur orange est connue d'Eve.
- **Étape 3 :** Bob fait de même : il choisit une couleur secrète (ici du cyan) qu'il mélange à la peinture commune et il obtient du bleu. Bob envoie sa couleur bleu à Alice. La couleur bleue est connue d'Ève.
- **Étape 4 :** Alice prend la couleur reçue (le bleu) qu'elle mélange avec sa couleur secrète rouge. Elle obtient une couleur brune. Bob prend la couleur reçue (le orange) qu'il mélange avec sa couleur secrète cyan. Il obtient la même couleur brune.
- **Étape 5 :** A la fin du protocole, Alice et Bob possèdent la même couleur brune, qui représente la couleur secrète partagée. La sécurité de l'échange repose sur le fait qu'il est difficile pour Eve d'extraire les couleurs utilisées pour obtenir les couleurs publiques orange et bleue, Eve ne connaît pas la couleur brune finale.

On parle d'échange de clef mais en fait il s'agit plutôt de générer une clef commune sans partager de secret.

Méthode *L'arithmétique au service de la sécurité informatique (1/2)*

La connaissance des fondements mathématiques de l'échange Diffie-Hellman est hors-programme.

L'échange Diffie-Hellman met en jeu l'arithmétique des entiers naturels. Sa sûreté reposé sur l'équivalence entre décryptage et résolution d'un problème mathématique difficile.

- **Étape 1 :** Alice et Bob conviennent de deux entiers rendus publics un grand nombre premier p par exemple 541 et un générateur g du *groupe cyclique multiplicatif* $(\mathbb{Z}/p\mathbb{Z})*$ qui doit posséder la propriété de générer avec ses puissances g^x tous les restes non nuls possibles de la division euclidienne par p (de 1 à $p-1$).

  Le calcul du générateur g d'un groupe cyclique est difficile!

- **Étape 2 :** Alice choisit un entier naturel secret $a = 177$ inférieur à p, calcule le reste de la division euclidienne de g^a par p et le transmet à Bob :

```
>> (2 ** 177) % 541
219
```

- **Étape 3 :** Bob fait de même : il choisit un entier naturel secret $b = 292$ inférieur à p, calcule le reste de la division euclidienne de g^b par p et le transmet à Alice :

```
>> (2 ** 292) % 541
69
```

☞ **Étape 4 :** Alice élève la valeur g^b (public) transmise par Bob à la puissance a (privé), c'est-à-dire $(g^b)^a = g^{ba}$ et garde le reste de la division euclidienne par p :

```
>>> (69 ** 177) % 541
174
```

Bob opère de façon symétrique : il calcule, c'est-à-dire $(g^a)^b = g^{ab}$ et garde le reste de la division euclidienne par p :

```
>>> (219 ** 292) % 541
174
```

☞ **Étape 5 :** Puisque $g^{ab} = g^{ba}$ (importance de la *commutativité* de la multiplication des entiers!), Alice et Bob obtiennent le même nombre chacun de leur côté, ce sera la clef secrète partagée.

Pour calculer cette clef secrète, Eve doit d'abord extraire la valeur de a (privée) à partir de g^a (public) ou la valeur de b (privée) à partir de g^b (public). Ce calcul doit s'effectuer dans l'ensemble $1, \ldots, p-1$ des restes non nuls de la division euclidienne par p, c'est **le problème mathématique difficile du logarithme discret**. Pour des valeurs assez grandes de p, il n'est pas réalisable dans un temps raisonnable, ce qui garantit la sûreté du protocole.

Exercice 5 *Attaque de l'homme du milieu*

Le protocole d'échanges de clef de Diffi-Hellman est vulnérable à l'attaque de l'homme du milieu. Un attaquant peut intercepter les messages entre Alice et Bob, les remplacer par les siens et se faire passer pour Bob auprès d'Alice et pour Alice auprès de Bob.
Cette vulnérabilité peut être corrigée si Alice et Bob apposent une signature numérique infalsifiable sur leurs messages, ce qui est rendu possible par un chiffrement asymétrique à clef publique comme RSA. Reste à vérifier l'authenticité de la signature … C'est le rôle des autorités de certification.
Traiter les questions a., b. et c. de l'exercice 30 p.217.

3.4 Principe d'un chiffrement asymétrique à clef publique

Point de cours 4

☞ Un **chiffrement asymétrique à clef publique** repose sur l'attribution d'une paire de clefs **(clef publique, clef privée)** à chaque utilisateur.

☞ La **clef publique** peut être publiée dans un annuaire, la **clef privée** doit rester secrète.

☞ Chaque utilisateur peut alors définir :

- une fonction de *chiffrement* avec sa **clef publique**, K^{pub}_{Alice} pour Alice et K^{pub}_{Bob} pour Bob ;
- une fonction de *déchiffrement* avec sa **clef privée**, K^{priv}_{Alice} pour Alice et K^{priv}_{Bob} pour Bob.

☞ Pour chaque paire de clefs un message chiffré avec la clef publique peut être déchiffré avec la clef privé et réciproquement. Autrement dit chaque fonction K^{pub}_{Alice} et K^{priv}_{Alice} peut être utilisée pour chiffrer ou déchiffrer, cela dépend de l'application souhaitée :

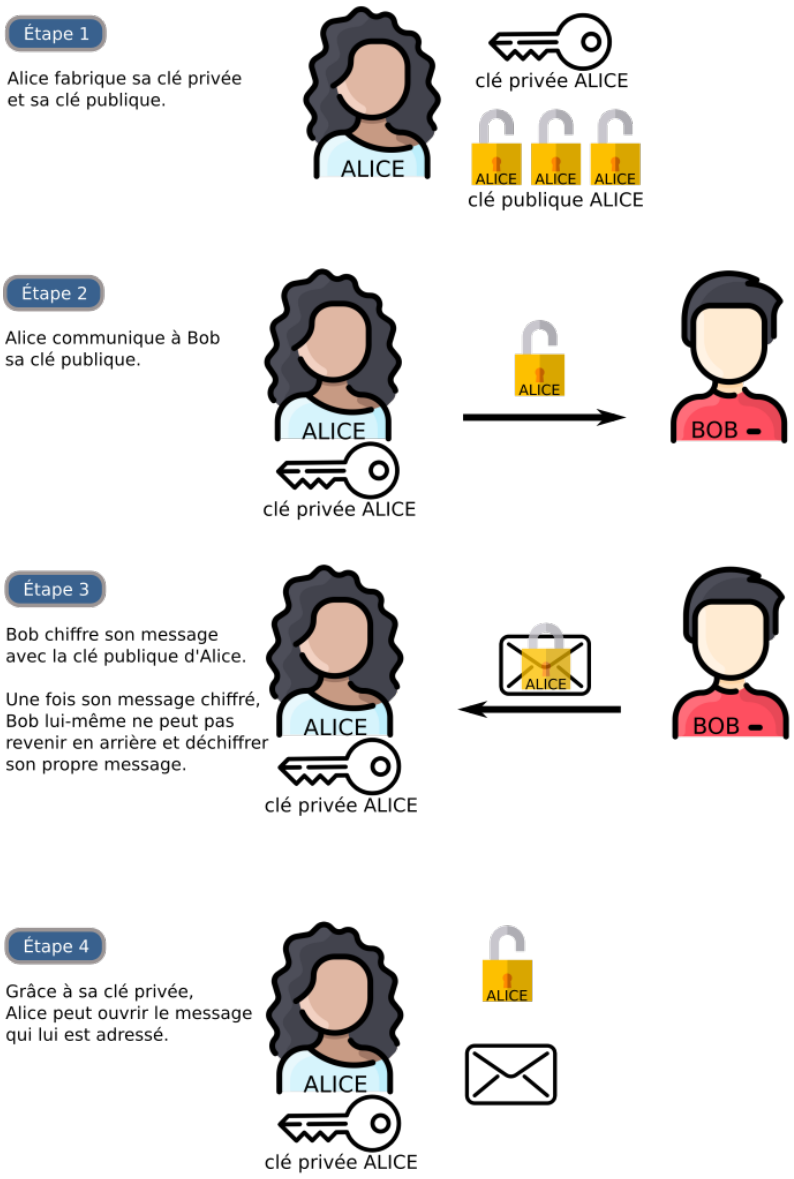
$$K^{pub}_{\text{Alice}}(K^{priv}_{\text{Alice}}(m)) = m = K^{priv}_{\text{Alice}}(K^{pub}_{\text{Alice}}(m)) \quad (1)$$

☞ De plus si on connaît $K^{pub}_{\text{Alice}}(m)$, il doit être impossible de calculer m dans un temps raisonnable sans connaître la clef privée . On dit que K^{pub}_{Alice} est une **fonction à sens unique** (impossible de retrouver l'antécédent à partir de l'image) avec **trappe** (on peut retrouver l'antécédent si on connaît la clef privée). Cette propriété garantit la sûreté du chiffrement.

Analogie 2 *Chiffrement asymétrique à clef publique et boîte aux lettres*

On peut comparer un chiffrement asymétrique à clef publique avec l'utilisation d'une *boîte aux lettres.*

- *Tout le monde peut déposer une lettre dans la boîte aux lettres de Bob.*
- *La boîte aux lettres de Bob est verrouillée par une serrure dont lui seul possède la clef.* (Asymétrie)
- *Pour transmettre un message secret à Bob, il suffit de le déposer dans sa boîte, on est sûr que lui seul pourra le lire.*



Graphique 4 : source : Gilles Lassus https://glassus.github.io/terminale_nsi

Exercice 30 p.217

30 Algorithme de Diffie-Hellman authentifié

On considère l'algorithme de Diffie-Hellman décrit au paragraphe 3.3 du cours (Figure 8).

a. Supposons qu'un attaquant Charlie s'interpose entre Alice et Bob. Il peut intercepter les messages échangés et les remplacer par les siens.

Il a accès à la peinture publique jaune et il possède la couleur secrète rose (dont le mélange avec du jaune donne du rouge). Comment peut-il faire pour créer une clef secrète partagée avec Alice et une clef secrète partagée avec Bob ?

Cette technique s'appelle l'*attaque de l'homme du milieu*.

b. Comment Charlie peut-il exploiter l'attaque précédente pour lire toute la communication entre Alice et Bob ?

c. A-t-il la possibilité de modifier les messages qui circulent entre Alice et Bob ?

d. Pour éviter cela, Alice et Bob ont désormais accès à un algorithme de signature et possèdent chacun leur clef de signature privée et la clef de vérification publique de l'autre. Ils signent tous les messages publics envoyés. Décrire les modifications apportées à l'algorithme et expliquer pourquoi cela empêche Charlie de mener son attaque.

a)

b)

Mais attaque « man in the middle »

Alice — Charlie — Bob

A

$a = g^A \bmod p$ →

← $b = g^D \bmod p$

D

$a = g^D \bmod p$ →

← $b = g^B \bmod p$

B

$kA = b^A \bmod p = g^{DA} \bmod p$

$kA = a^D \bmod p = g^{AD} \bmod p$
$kB = b^D \bmod p = g^{BD} \bmod p$

$kB = a^B \bmod p = g^{DB} \bmod p$

le démon sait tout !

Les attaques *man in the middle* sont nombreuses et souvent le prélude à des attaques plus fortes

c) Oui Charlie peut modifier les messages circulant entre Alice et Bob

d) Alice et Bob peuvent signer chacun de leur envoi avec leur clef privée, le destinataire pourra déchiffrer leur signature avec leur clef publique et ainsi obtenir une preuve d'authentification du message.

Chiffrement asymétrique : 4 cadenas !

- Idée géniale de Diffie et Hellman (et autres), implémentée aussi dans RSA (Rivest-Shamir-Adleman)
- Utiliser deux clefs, l'une Xpub publique et l'autre Xpriv privée,
- telles que $\{\{m\}_{Xpriv}\}_{Xpub} = \{\{m\}_{Xpub}\}_{Xpriv} = m$
- Mais comment être sûr que le message vient d'Alice (ou Bob) ?

Bpub

Apub
Apriv

$ab = \{\{m\}_{Apriv}\}_{Bpub}$ →

← $ba = \{\{r\}_{Bpriv}\}_{Apub}$

Apub

Bpub
Bpriv

$r = \{\{ba\}_{Apriv}\}_{Bpub}$

$m = \{\{ab\}_{Bpriv}\}_{Apub}$

Trop cher en pratique, transmettre $\{m, \{Hash(m)\}_{Apriv}\}_{Bpub}$

L'authenticité des clefs publiques peut être garantie par un tiers de confiance (autorité de certification)

Exercice 6 *Analogie du cadenas et échange de secret avec un chiffrement asymétrique*

En reprenant l'analogie du chiffrement symétrique avec un coffre fort verrouillé par une serrure à clef secrète partagée, on peut construire une analogie pour un chiffrement asymétrique à clef publique. On remplace la serrure par un cadenas que tout le monde peut verrouiller mais que seul Bob peut déverrouiller avec sa clef.

Comment Alice peut-elle envoyer à Bob de façon confidentielle une clef secrète de chiffrement symétrique ?

Correction sur la page suivante.

Méthode

☞ **Une meilleure gestion des clefs avec le chiffrement asymétrique :**

Dans un **chiffrement symétrique** tout possesseur de la clef secrète peut écrire et recevoir des messages confidentiels de tout autre possesseur de la même clef. Pour communiquer avec plusieurs personnes sur des canaux séparés, Alice doit donc posséder d'autant de clefs que d'interlocuteurs.

Dans un **chiffrement asymétrique** à clef publique, tout le monde peut écrire à Alice avec sa clef publique mais elle seule peut lire le message (*asymétrie*). Un seule paire de clefs par utilisateur suffit.

☞ **Comment Alice peut-elle envoyer un message confidentiel à Bob ?**

Alice chiffre son message en clair clair_A avec la clef publique de Bob, puis elle envoie le chiffré $K^{pub}_{\text{Bob}}(\text{clair}_A)$ à Bob. Bob est le seul à pouvoir déchiffrer le chiffré avec sa clef privée :

$$\text{clair}_A \xrightarrow{K^{pub}_{\text{Bob}}} \text{chiffré}_A \xrightarrow{K^{priv}_{\text{Bob}}} \text{clair}_A$$

Alice peut ainsi envoyer à Bob de façon confidentielle une clef de chiffrement symétrique

☞ **Comment Alice peut-elle signer numériquement un message pour Bob ?**

Voir exercice ci-dessous.

Exercice 7 *Signature numérique et problème de l'authentification*

Alice et Bob disposent chacun d'une paire de clefs (clef publique, clef privée) pour un système de chiffrement asymétrique robuste.

Alice souhaite envoyer un message confidentiel m à Bob en lui joignant une signature numérique s construite à partir du message m, attestant qu'elle est bien l'auteur du message.

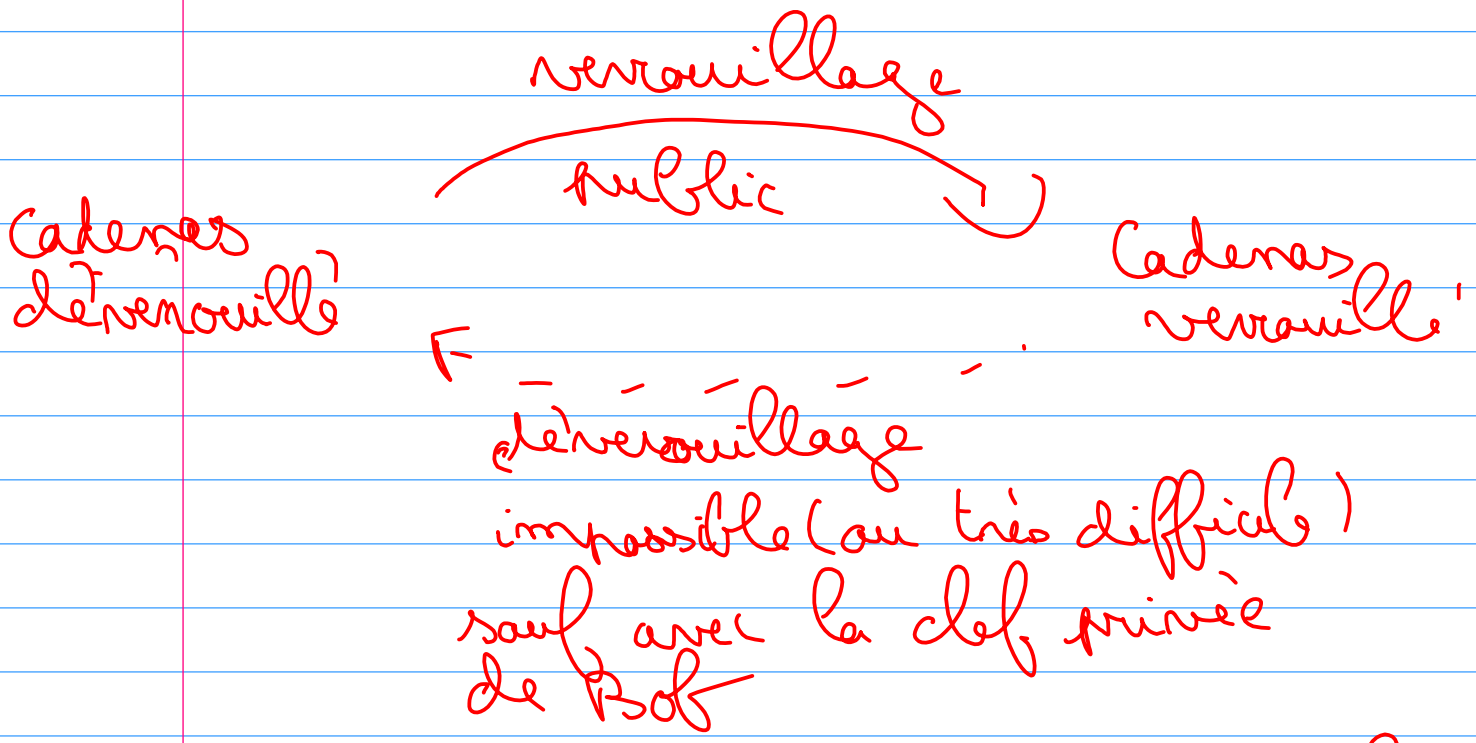
1. On suppose que Bob a pu vérifier que la clef publique d'Alice est bien la sienne (*authentification de la clef publique*). Décrire un protocole permettant la signature numérique du message d'Alice

Exercice 6

Alice place la clef secrète dans un coffre qu'elle verrouille avec le cadenas à clef publique de Bob.

Elle envoie le coffre à Bob.

Seul Bob peut déverrouiller le cadenas avec sa clef privée



On dit que la fonction de verrouillage est une fonction de chiffrement à sens unique (déverrouillage impossible) à trappe (sauf avec la clef privée).

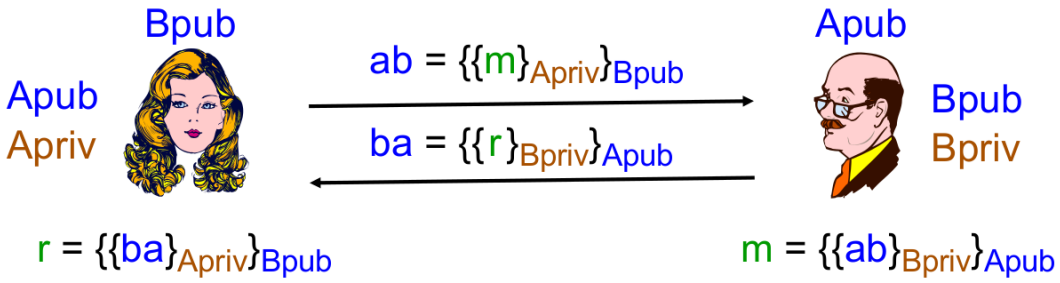
Exercice 7 ***Signature numérique et problème de l'authentification***

Alice et Bob disposent chacun d'une paire de clefs (clef publique, clef privée) pour un système de chiffrement asymétrique robuste.
Alice souhaite envoyer un message confidentiel m à Bob en lui joignant une signature numérique s construite à partir du message m, attestant qu'elle est bien l'auteur du message.

1. On suppose que Bob a pu vérifier que la clef publique d'Alice est bien la sienne (*authentification de la clef publique*). Décrire un protocole permettant la signature numérique du message d'Alice

Chiffrement asymétrique : 4 cadenas !

- Idée géniale de Diffie et Hellman (et autres), implémentée aussi dans RSA (Rivest-Shamir-Adleman)
- Utiliser deux clefs, l'une Xpub publique et l'autre Xpriv privée,
- telles que $\{\{m\}_{Xpriv}\}_{Xpub} = \{\{m\}_{Xpub}\}_{Xpriv} = m$
- Mais comment être sûr que le message vient d'Alice (ou Bob) ?



Trop cher en pratique, transmettre $\{m, \{Hash(m)\}_{Apriv}\}_{Bpub}$

en utilisant les deux paires de clefs d'Alice et Bob.

Alice va chiffrer sa signature avec sa clef privée puis chiffrer le tout avec la clef publique de Bob et lui envoyer. Celui-ci déchiffre le message reçu avec sa clef privée (confidentialité) puis il déchiffre la signature d'Alice avec sa clef publique (authentification). Voir schéma de Gérard Berry en page précédente.

2. Comment peut-on attaquer ce protocole si Bob ne peut pas vérifier l'authenticité de la clef publique d'Alice ?

Spoiler : https://interstices.info/verifier-la-securite-de-nos-communications/

Attaque de type "Man of the Midle" quelqu'un se fait passer pour Alice auprès de Bob et lui transmet sa propre clef publique en la faisant passer pour celle d'Alice.

3.5 Chiffrement asymétrique RSA

Méthode *L'arithmétique au service de la sécurité informatique (2/2)*

La connaissance des fondements mathématiques de RSA est hors-programme.

Le **chiffrement asymétrique à clef publique RSA** a été proposé en 1978 par **Ronald Rivest**, **Adi Shamir** et **Leonard Adleman**.
Ses applications pratiques sont :

- la signature numérique de certificat pour la phase d'authentification dans le protocole `HTTPS` ;
- la signature numérique de condensats de fichiers (obtenus avec une fonction de hachage) ;
- la signature numérique d'une carte bancaire : elle est insérée dans sa puce et vérifiée avec la clef publique de la banque lors de chaque utilisation de la carte sur un terminal etc …

RSA repose sur l'arithmétique des entiers naturels.

- **Étape 1 :** Pour fabriquer sa paire de clefs RSA, Bob, choisit d'abord deux grands entiers premiers p et q et il considère leur produit $n = p \times q$.
- **Étape 2 :** Bob choisit alors un entier $1 \leqslant e < n$, tel que le reste pgcd$(p-1)(q-1), e) = 1$.
  D'après l'*algorithme d'Euclide étendu* connu depuis l'Antiquité, il existe un unique entier $1 \leqslant d < n$ tel que le reste de ed dans la division euclidienne par $(p-1)(q-1)$ est 1 ce qu'on note :

$$ed \bmod (p-1)(q-1) = 1$$

- **Étape 3 :** D'après le *théorème d'Euler*, pour tout entier a tel que que $a \bmod (p-1)(q-1) = 1$, on a pour tout entier $1 \leqslant m < n$, $m^a \bmod n = m$.

  On déduit alors de l'étape 2 que pour tout entier $1 \leqslant m < n$, on a :

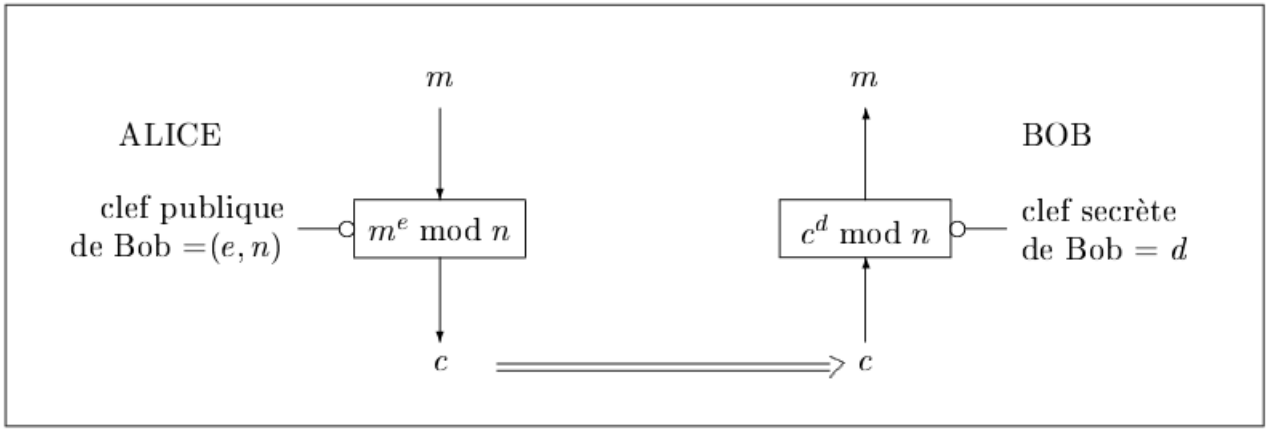
$$(m^e)^d \bmod n = m^{ed} \bmod n = m$$

Dans le raisonnement précédent, le couple (e, n) est la **clef publique** de Bob, l'entier d sa **clef privée** et m représente le message en clair.
On peut vérifier que $(m^e)^d \bmod n = m = (m^d)^e \bmod n$ ce qui équivaut à la propriété (1) caractéristique des chiffrements asymétriques à clef publique.

RSA est sûr parce que la fonction $m \mapsto m^e$ est à **sens unique**. Il est très difficile de retrouver m à partir de m^e sans appliquer la clef privée d avec $(m^e)^d \bmod m = m^{ed} \bmod m = m$.
Ironiquement, si on connaît la factorisation de n en $p \times q$, le calcul de d tel que $ed \bmod (p-1)(q-1) = 1$ est très facile avec le vieil algorithme d'Euclide.

La sûreté de RSA repose donc sur la difficulté de la factorisation d'un entier en produit de deux facteurs premiers. Ce problème est impossible à résoudre en temps raisonnable si n assez grand (actuellement il est recommandé de travailler avec une clef publique n de taille au moins 1024 *bits). Néanmoins l'algorithme de Shor résolverait facilement ce problème sur un* ***ordinateur quantique****.*

*Les chiffrements asymétriques utilisent des opérations plus coûteuses et des clefs beaucoup plus longues (*2048 *bits) que les chiffrements symétriques (*256 *bits). Sur machine, un chiffrement RSA est ainsi environ* 1000 *fois plus lent qu'un chiffrement AES.*



3.6 Résolution du problème d'authentification

Point de cours 5 *Certificats et tiers de confiance*

Un chiffrement symétrique à clef publique ou le protocole de Diffie-Hellman permettent d'échanger un secret, comme une clef de chiffrement symétrique, uniquement à partir de données publiques.
De plus on a vu dans l'exercice 7 que pour envoyer un secret à Bob, Alice peut joindre à son message une signature s en le chiffrant avec sa clef privée. À réception, Bob va déchiffrer la signature avec la clef publique d'Alice et s'il obtient le même contenu que le message, ce sera une preuve que la signature a bien été générée par la clef privée d'Alice.
Néanmoins, quel degré de confiance Bob peut-il accorder au fait que la clef publique dont il dispose est bien celle d'Alice ? Un *homme du milieu* pourrait s'interposer et fournir à Bob une clef publique en la présentant comme celle d'Alice. C'est le **problème de l'authentification**.

Cette confiance dans l'authenticité d'une clef publique peut être assurée par un **tiers de confiance**.
Comme l'état garantit l'identité d'un individu en lui fournissant une carte d'identité signée, une **autorité de certification**, va délivrer un certificat avec plusieurs champs dont :

- la clef publique d'Alice ;
- une description de l'autorité et de l'identité d'Alice ;
- une signature des champs précédents avec la clef privée de l'autorité, la clef publique de l'autorité étant par exemple contenue dans le navigateur pour le Web.

Le certificat fourni par Alice à Bob lui permet alors de prouver l'authenticité de sa clef publique.
Notons $s = K^{priv}_{\text{Autorité}}(K^{pub}_{\text{Alice}})$ la signature du certificat d'Alice. Bob peut la vérifier avec la clef publique de **l'autorité de certification**, institution en laquelle il a confiance (le mécanisme peut être plus compliqué avec une chaîne de confiance) :

$$K^{pub}_{\text{Autorité}}(s) = \underbrace{K^{pub}_{\text{Autorité}}(K^{priv}_{\text{Autorité}}}_{\text{s'annulent}}(K^{pub}_{\text{Alice}})) = K^{pub}_{\text{Alice}}$$

Exercice 8 *Certificat SSL*

Sur le Web, l'accès à un serveur avec le protocole `HTTPS` commence par l'authentification de la clef publique de chiffrement du serveur avec un certificat d'authentification dit SSL au format X.509.

1. Dans un navigateur Web ouvrir la page `https://eduscol.education.fr/`, puis afficher le certificat en cliquant sur le cadenas et télécharger le certificat (fichier `eduscol-education-fr.pem`).
2. Ouvrir `eduscol-education-fr.pem` avec un éditeur de texte. Le fichier est-il lisible ? En fait il est encodé en base 64 (qu'est-ce donc ?).

   .....

   .....

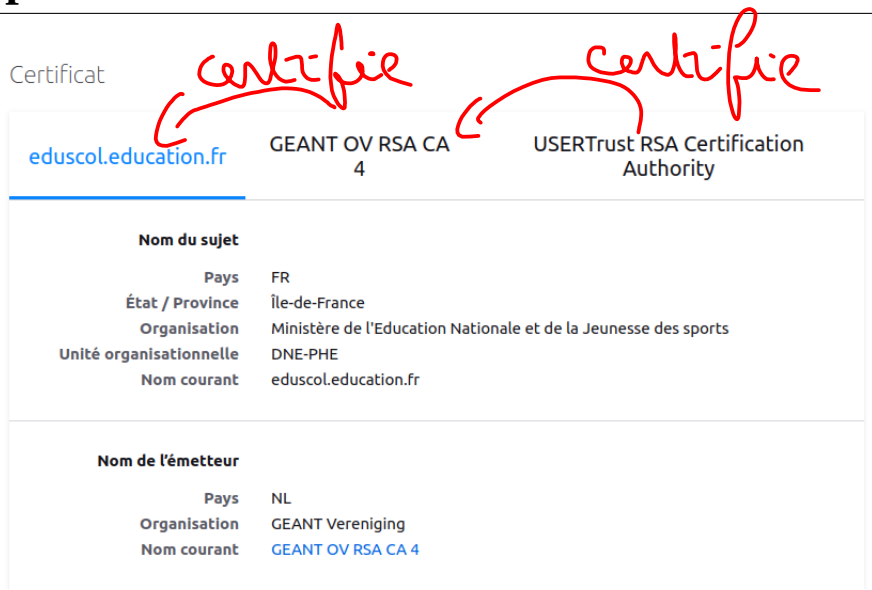
   Pour éditer son contenu, utiliser la commande :

   ```
   openssl x509 -in eduscol-education-fr.pem -text
   ```

   Rechercher : l'émetteur du certificat, son sujet, ses dates de validité, l'algorithme de chiffrement utilisé, les caractéristiques de la clef publique du sujet.

   Émetteur : GEANT Sujet : MEN
   Algorithme de chiffrement de la signature :
   sha384 WithRSA encryption
   clef publique de 4096 bit, Module en base 16
   et exposant 65537
3. Dans le navigateur, on peut afficher la chaîne de confiance du certificat, la reconstituer.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            76:a0:3b:54:5c:6f:9d:7a:e4:30:84:fe:01:7a:87:c5
        Signature Algorithm: sha384WithRSAEncryption
        Issuer: C = NL, O = GEANT Vereniging, CN = GEANT OV RSA CA 4
        Validity
            Not Before: May  2 00:00:00 2022 GMT
            Not After : May  2 23:59:59 2023 GMT
        Subject: C = FR, ST = \C3\8Ele-de-France, O = Minist\C3\A8re de l'Education Nationale et de la Jeunesse des sports, OU = DNE-PHE, CN = eduscol.education.fr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (4096 bit)
                Modulus:
                    00:e1:81:5d:96:e2:de:dd:71:ef:39:07:ca:db:23:
                    33:af:61:40:fc:fa:73:bc:72:45:22:02:25:7d:5f:
                    d7:20:91:ab:06:a0:04:64:68:85:4f:77:64:a2:33:
                    29:ed:8e:bc:f2:bb:f9:41:f5:c7:bd:fd:ef:f0:5e:
                    eb:f8:78:43:c2:10:32:e3:0a:bc:df:0c:2a:d8:90:
                    1d:97:48:c5:4d:51:5d:ef:ed:4e:36:81:4f:f6:79:
                    d5:e6:51:73:3a:5f:e7:ae:69:06:c2:24:af:7b:15:
                    af:27:f2:ea:e8:53:59:3b:e8:29:92:a5:08:85:6d:
                    1f:ca:a3:b3:ee:dc:67:d2:19:75:c1:44:bd:7f:58:
                    f7:55:94:83:7a:2a:5c:60:ad:19:8a:0e:2a:9b:c0:
                    8d:58:d5:75:d6:12:61:ff:70:62:b8:30:c5:92:bd:
                    ad:0e:f3:06:11:53:81:b1:f0:e7:f8:7c:14:4b:ea:
                    49:03:bf:7e:80:b6:5d:18:8c:75:4d:6b:67:35:35:
                    ae:94:84:1b:a9:f2:de:a8:e6:7e:1c:2f:de:90:ce:
                    88:f9:19:76:c1:f9:f0:cb:bf:1e:68:9d:52:9a:22:
                    52:19:9b:58:14:bd:d4:c1:0e:20:d8:06:b9:78:71:
                    9c:c0:f3:b6:55:e1:89:f8:e5:6e:97:d7:94:22:be:
                    fc:80:6c:25:30:96:b6:4e:fa:6f:37:d0:f9:4b:e4:
                    b9:93:3f:1e:44:e4:95:ae:b0:f7:d3:65:be:63:61:
                    e8:ff:33:a5:30:00:46:ab:9a:6b:d1:aa:59:49:10:
                    65:94:73:ef:7c:bb:7d:43:18:3b:72:ed:4c:43:78:
                    ed:97:15:0f:86:1b:af:46:41:b1:04:00:5b:2c:94:
                    7b:d4:7c:2b:26:84:72:7a:e4:00:48:3a:45:71:6a:
                    82:76:97:08:49:17:e9:8b:ec:36:84:14:41:36:61:
                    17:4a:27:ad:a8:be:8e:2a:41:26:81:58:e6:47:92:
                    88:71:9d:ab:81:e3:84:83:b5:1f:7b:d9:83:d9:f7:
                    62:4e:9e:63:ba:11:22:a3:37:d9:41:5b:f7:58:c5:
                    77:15:ef:6d:ac:4f:a2:23:98:b5:93:2d:20:0f:6c:
                    91:2f:93:ac:34:e7:b5:7c:e5:bc:2a:7c:ab:7c:b2:
                    1b:1e:ea:d0:9e:d1:23:bb:e8:bb:e5:e4:33:39:ed:
                    68:b2:8d:47:14:fb:42:aa:53:62:86:fd:4b:5f:8d:
                    cd:5a:15:b8:a9:4f:2f:b1:30:11:f5:a1:75:a2:c3:
                    2b:cf:50:3c:b6:c2:8e:dc:a7:ce:db:56:44:22:e6:
                    49:e7:73:02:81:9d:8a:bd:2a:38:42:08:4f:c2:9f:
                    1e:95:ed
                Exponent: 65537 (0x10001)
```



Graphique 6 : Chaîne de certification

. . . . . .

. . . . . .

. . . . . .

4 Application au protocole HTTPS

4.1 Intégration d'une couche de sécurité dans l'architecture existante

Point de cours 6

La pile de protocoles TCP/IP est la base logicielle d'Internet depuis sa naissance au début des années 1980 : IP pour l'adressage et le routage et TCP pour le transport et la qualité de service. Avec l'apparition du Web au début des années 1990, le protocole HTTP s'est inséré dans la *couche application* de la pile. Ses premières applications étant le partage ouvert de connaissances, les données pouvaient transiter en clair.
Avec le développement du commerce en ligne, des transactions bancaires, la multiplication des espaces privés sur les sites, il est devenu nécessaire d'ajouter une couche de sécurité au protocole HTTP pour garantir la confidentialité des communications et l'authentification des participants. Aujourd'hui 90 % du traffic Web est en HTTPS, la version sécurisée d'HTTP.

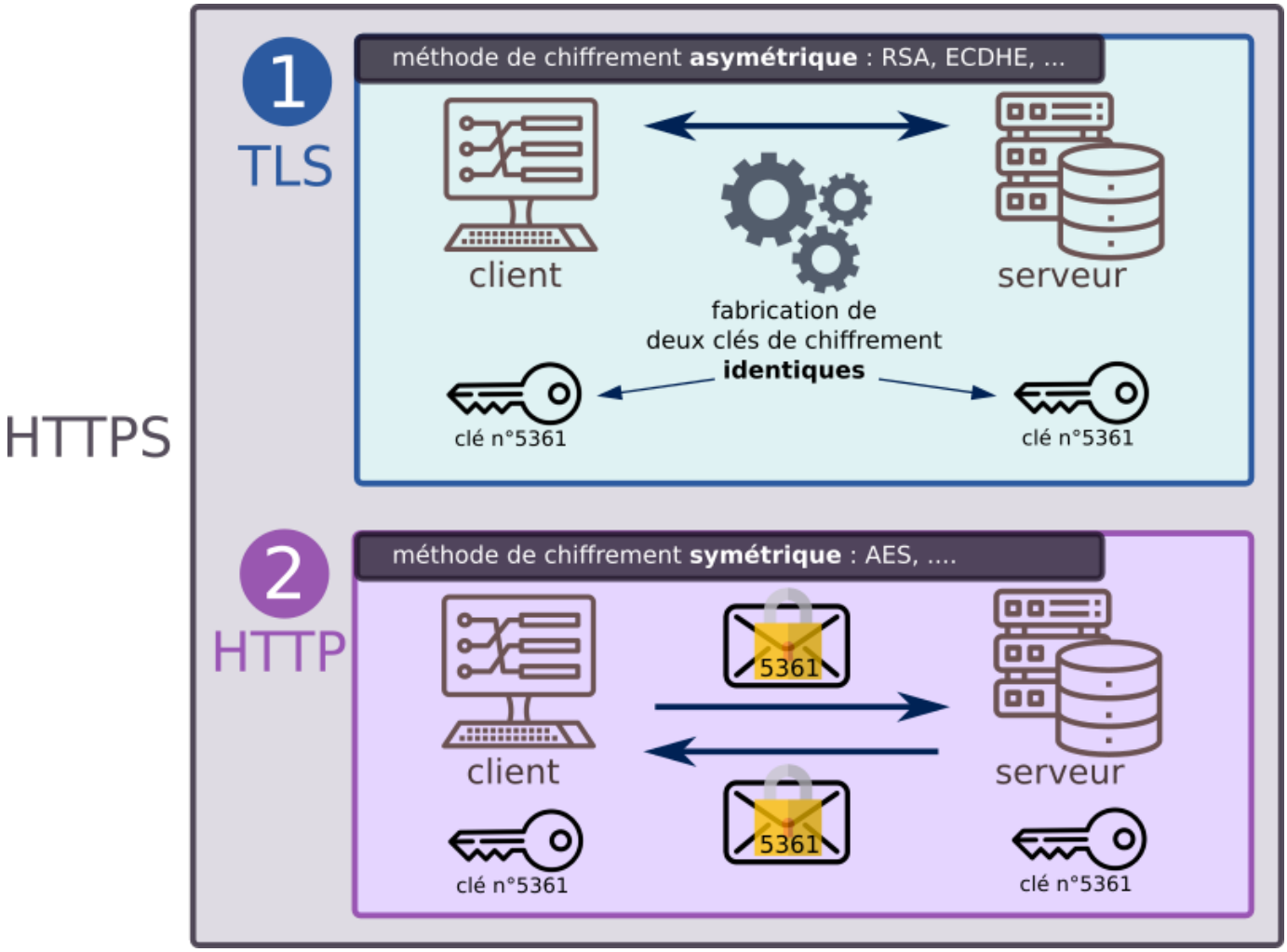
Une connexion en HTTPS sur un serveur Web, s'effectue sur le port TCP 443 (au lieu de 80 pour HTTP) et elle se décompose en deux phases successives :

- **Phase 1 : la poignée de main TLS** Le client vérifie le certificat SSL du serveur. Si l'authentification réussit, le client et le serveur se mettent d'accord sur une clef de **chiffrement symétrique**, à l'aide d'une méthode de **chiffrement asymétrique** (**Diffie-Hellman** ou **RSA**)
- **Phase 2 : échange HTTP** Le client et le serveur peuvent échanger les requêtes et réponses de façon confidentielles en les chiffrant avec la clef de chiffrement symétrique établie lors de la phase 1.

Pour résumer :

$$\text{TLS} + \text{HTTP} = \text{HTTPS}$$

Il n'est pas possible d'utiliser uniquement un chiffrement asymétrique pour chiffrer tous les échanges de données client/serveur car le coût en calcul des chiffrements asymétriques est bien supérieur (environ ×1000) aux chiffrements symétriques. HTTPS utilise donc un système hybride : authentification avec chiffrement asymétrique et chiffrement avec chiffrement symétrique.



Graphique 7 : source : Gilles Lassus https://glassus.github.io/terminale_nsi

4.2 Poignée de main TLS : authentification, échange de clef, chiffrement

Point de cours 7 *Poignée de main TLS*

☞ **Étape 1 : présentation du client**

Le client envoie un message initial « Hello », ainsi que différentes informations : la version de TLS utilisée et les différents suites de chiffrement (asymétrique + symétrique) qu'il peut utiliser.

☞ **Étape 2 : présentation du serveur**

Le serveur répond en renvoyant son certificat contenant sa clé publique et signé par une autorité de certification (tiers de confiance) ainsi que la suite de chiffrement choisie.

☞ **Étape 3 : authentification du serveur par le client**

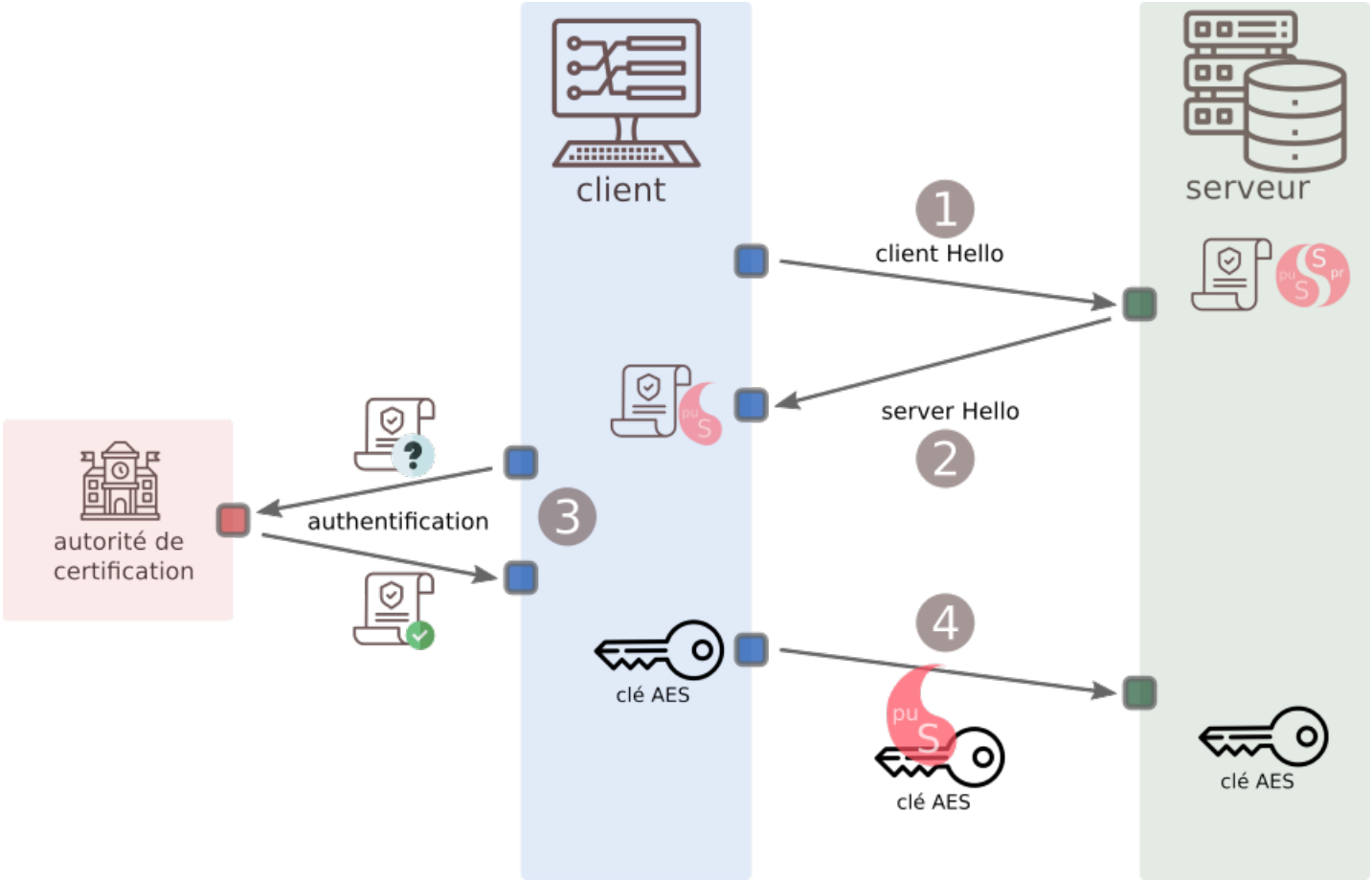
Le client vérifie la validité du certificat avec la clef publique de l'autorité de certification qu'il possède en général dans son navigateur Web ou son système d'exploitation.

☞ **Étape 4 : choix de la clef de chiffrement**

Le client et le serveur conviennent d'une clef de chiffrement symétrique. Deux alternatives :

- **Chiffrement asymétrique à clef publique (RSA)** le client choisit une clef, la chiffre avec la clef publique du serveur pour assurer la confidentialité du transfert et l'envoie au serveur qui la déchiffre avec sa clef privée.
- **Protoclé d'échange de clef de Diffie-Hellman** le client et le serveur génèrent une clef commune à l'aide du protocole d'échange de Diffie-Hellman .

À la fin d'une « poignée de main TLS », le serveur est authentifié auprès du client, les deux ont échangé une clef de chiffrement symétrique, la transmission par protocole `HTTP` de données chiffrées avec cette clef (en général avec AES) peut commencer.



Graphique 8 : source : Gilles Lassus https://glassus.github.io/terminale_nsi