Compression, Correction et Chiffrement

Vendredi 12 février

Option Informatique

Ecole Alsacienne

Plan

1. Compression
2. Détection et correction des erreurs
3. Chiffrement de l'information
4. Premiers exemples de chiffrement
5. Clé publique – clé privée

Compression

Compression

- Version textuelle

```
Hello world
```

- Version ASCII

```
01001000 01100101 01101100 01101100 01101111
00100000 01110111 01101111 01110010 01101100
01100100
```

- Version Huffman

```
1101011110101111100001111100010010
```

symbole		H	d	e	l	o	r	w
référence	1100	1101	010	011	10	111	000	001

- Question : *Comment obtenir un tel code ?*

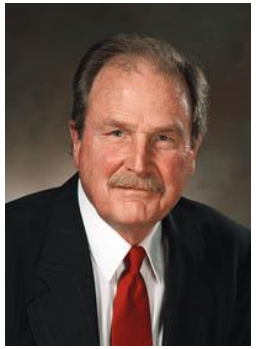
Code de Huffman

- L'idée est d'utiliser des références plus courtes pour représenter les symboles qui apparaissent le plus fréquemment

- Un code qui utiliser des références de longueurs différentes est appelé un code de Huffman

- David Albert Huffman, pendant sa thèse au MIT

A Method for the Construction of Minimum-Redundancy Codes



Code de Huffman

- **Exercice** : Imaginez un code pour représenter le message `je pense donc je suis` en un minimum de caractères

- Approche naïve : affecter toutes les références dans l'ordre

symbole		e	s	j	n	p	d	o	c	u	i
occurences	4	4	3	2	2	1	1	1	1	1	1
référence	0	1	00	01	10	11	000	001	010	011	100

- Le message devient donc :

```
01 1 0 11 1 10 00 1 0 000 001 10 010 0 01 1 0 00 011 100 00
```

- Problème :
  - Il n'y a pas d'espace en binaire :

```
01101111000100000011001000110000111000 0
```

  - Ce code est ambigu : il y a plusieurs façons de le décoder !

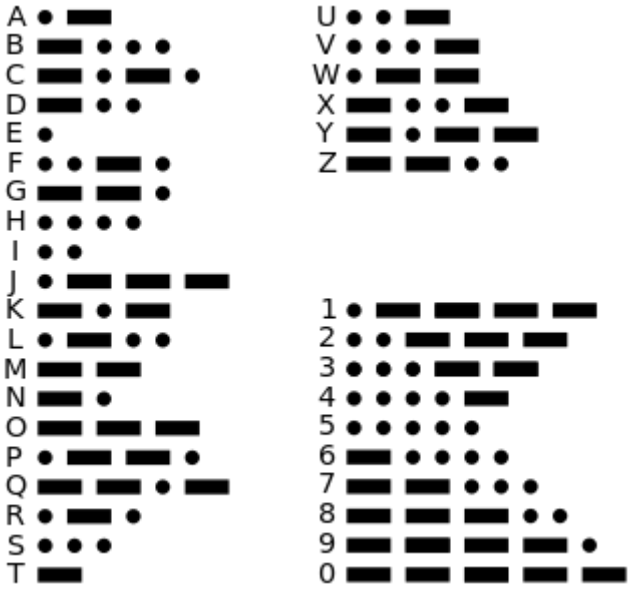
Code préfixe

- On dit qu'une référence est préfixe d'une autre référence si la première constitue exactement le début de la seconde.

- Un code préfixe est un code dans lequel aucune référence n'est préfixe d'une autre.

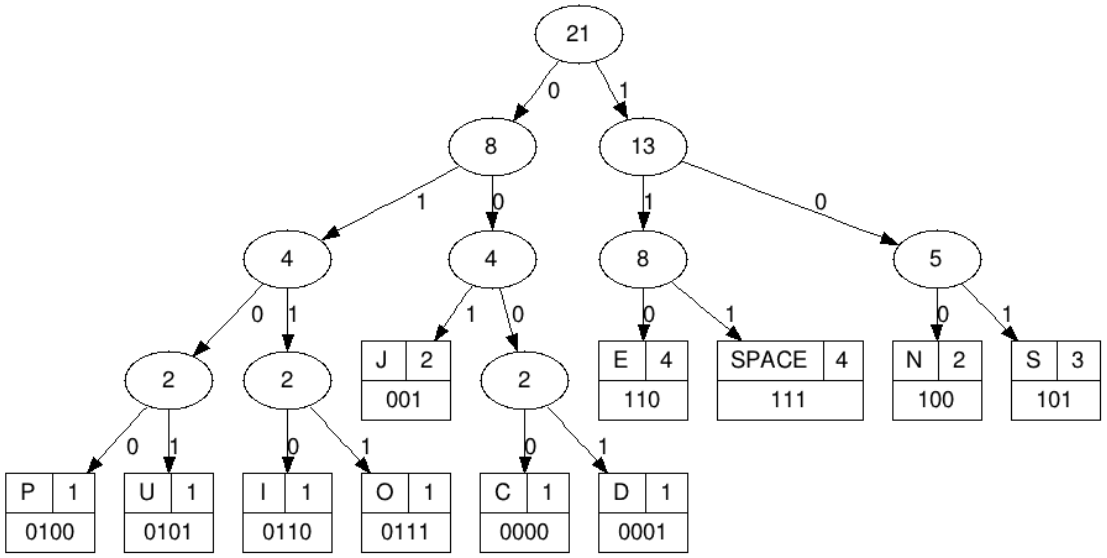
- Pour la compression de données, on utilisera toujours des codes préfixes.

- Exemple : le Morse n'est pas préfixe
  - ●●● est un code ambigu
  - Un séparateur est nécessaire (un « blanc »)



RETOUR À L'EXERCICE

- **Exercice** : Imaginez un code pour représenter le message `je pense donc je suis` en un minimum de caractères



symbole		e	s	j	n	p	d	o	c	u	i
occurences	4	4	3	2	2	1	1	1	1	1	1
référence	111	110	101	001	100	0100	0001	0111	0000	0101	0110

001110111010011010010111011110001011110000011100111010101010110101

Compression : Remarques

- Il existe des méthodes encore plus poussées pour atteindre des taux de compression encore plus importants

- En particuliers, certaines méthodes reposent sur la compression par dictionnaire : les références peuvent alors porter sur des mots entiers (et pas seulement des symboles).

- Il est fréquent de combiner plusieurs algorithmes de compression

- Remarques
  - Ces méthodes requièrent d'envoyer un dictionnaire en plus du message (avec la liste des références pour chaque symbole)
  - Cet ajout est négligeable dès que le message est un peu long

Compression avec perte

- Une autre façon d'augmenter le taux de compression : accepter une certaine perte d'information

- Des différences parfois invisibles à l'œil nu

36 Ko

13 Ko

3 Ko

- Les seuils varient en fonction des utilisations
  - Streaming vidéo : tolérance élevée
  - Enregistrement d'une image : tolérance faible
  - Sécurité d'un avion : tolérance zéro

Détection et correction des erreurs

UN OCÉAN DE DONNÉES

Un océan de données

- Avec un tel volume de données, il est inévitable que des erreurs se produisent

- Plusieurs sources d'erreurs possibles :
  - Usure
  - Distance
  - Perte d'information
  - Problèmes de réseau

- Dans l'idéal :
  - Détection automatique des erreurs
  - Correction automatique des erreurs

Un exemple d'erreur

- En ASCII, le texte `Hello` s'écrit :

```
01001000 01100101 01101100 01101100 01101111
```

- Imaginons qu'une erreur survienne lors de la transmission de ce message :

```
01011000 01100101 01101100 01101100 01101111
```

- On obtient **`X`**`ello`

UNE MÉTHODE SIMPLE : LA REDONDANCE

- Principe : chaque bit est envoyé 3 fois

- Exemple :
  - Texte :

    `Hello`
  - ASCII :

    `01001000 01100101 01101100 01101100 01101111`
  - ASCII redondant :

    `000111000000111000000000`
    `000111111000000111000111`
    `000111111000111111000000`
    `000111111000111111000000`
    `000111111000111111111111`

UNE MÉTHODE SIMPLE : LA REDONDANCE

- Si une erreur survient, on obtient un triplet dont tous les éléments ne sont pas identiques

  000111000**1**00111000000000

- On peut donc en déduire qu'il y a une erreur dans ce triplet

- On peut même corriger cette erreur automatiquement :
  - Si les 0 sont majoritaires, on peut supposer que le triplet est 000
  - Si les 1 sont majoritaires, on peut supposer que le triplet est 111

- Cette méthode fonctionne relativement bien, mais
  - Elle est couteuse : la longueur du message est triplée
  - Si deux erreurs ont lieu dans le même triplet, l'erreur persistera

Les bits de contrôle

- Une méthode moins couteuse consiste à ajouter un bit de contrôle tous les 10 bits transmis.

- Ce bit indique la parité du nombre de 1 dans le paquet de 10 correspondant :
  - S'il y avait un nombre pair de 1, le bit de contrôle vaut 0
  - S'il y avait un nombre impair de 1, le bit de contrôle vaut 1

`01001000 01`**`1`**`100101 0110`**`1`**`1100 011011`**`0`**`00 0110111`**`0`**

- En vérifiant ces bits de contrôle lors de la réception du message, on peut détecter les éventuelles erreurs :

`010`**`1`**`1000 01`**`1`**`100101 0110`**`1`**`1100 011011`**`0`**`00 0110111`**`1`**

Les bits de contrôle

- La longueur du message est allongée de 10%.
- Si plusieurs erreurs sont présentes dans un paquet de 11 bits, elles ne sont pas forcément détectées
- On peut placer les bits de contrôle tous les 100 bits ou tous les 1000 bits
  - La méthode est alors plus économique (le message est plus court)
  - Mais le risque que plusieurs erreurs soient présentes augmente.
- Cette méthode permet de détecter les erreurs, mais pas de les corriger
  - Pratique si on peut redemander le message (ex : échanges réseaux)
  - Insuffisant sinon (ex : lecture d'un DVD)

Les bits de contrôle : version avancée

- Il existe une version plus complexe et plus efficace de ces bits de contrôle, avec 20 bits de contrôle pour chaque paquet de 100 bits.

- Chaque paquet de 100 bits est vu comme un tableau avec 10 lignes et 10 colonnes.

- Un bit de contrôle est ajouté
  - pour chaque ligne
  - pour chaque colonne

0	1	0	0	1	0	0	0	0	1	**1**
1	0	0	1	0	1	0	1	1	0	**1**
1	1	0	0	0	1	1	0	1	1	**0**
0	0	0	1	1	0	1	1	1	1	**0**
0	0	1	0	0	0	0	0	0	1	**0**
1	1	0	1	1	1	0	1	1	0	**1**
1	1	1	1	0	1	1	1	0	0	**1**
1	0	0	1	1	0	1	1	0	0	**1**
0	1	1	0	0	1	0	0	0	0	**1**
1	0	0	0	0	0	0	0	1	0	**0**
0	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	

Les bits de contrôle : version avancée

- Un tel tableau permet de détecter et de corriger une éventuelle erreur

0	1	0	0	1	0	0	0	0	1	1
1	0	0	1	0	1	0	1	1	0	1
1	1	0	0	0	0	1	0	1	1	0
0	0	0	1	1	0	1	1	1	1	0
0	0	1	0	0	0	0	0	0	1	0
1	1	0	1	1	1	0	1	1	0	1
1	1	1	1	0	1	1	1	0	0	1
1	0	0	1	1	0	1	1	0	0	1
0	1	1	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0
0	1	1	1	0	1	1	1	0	0	

- Remarques
  - La longueur du message est allongée de 20%
  - Si plusieurs erreurs surviennent dans un paquet de 120 bits, elles ne seront pas forcément détectée

Exercices

- **Exercice 1.** Ajoutez des bits de contrôle à la séquence suivante : `01101011101111111001`.

- **Exercice 2.**
  - Ecrivez un texte de 13 caractères (avec des caractères ASCII)
  - Convertissez ce texte en codes ASCII (décimaux)
  - Transformez ces codes décimaux en binaires
  - Placez les 100 premiers bits dans un tableau
  - Ajoutez les bits de contrôles
  - Recopiez les 120 bits correspondants sur un bout de papier en y ajoutant une ou plusieurs erreurs
  - Donnez ce bout de papier à votre voisin
  - Essayez de retrouver le message codé sur le papier de votre voisin

Chiffrement de l'information

Chiffrement de l'information

- On appelle chiffrement de l'information le fait de transformer une information pour la rendre aussi incompréhensible que possible, sauf pour son destinataire

- On parle ainsi de
  - Chiffrer un message
  - Déchiffrer un message

- Sûreté et sécurité
  - Sûreté des informations : protection contre les erreurs involontaires (comme les erreurs de transmissions)
  - Sécurité des informations : protection contre les personnes malveillantes

Un peu de vocabulaire

- κρυπτός (*kruptos*, caché)
  - Les mots **crypter** et **cryptage** sont des anglicismes (*encryption*)
  - On peut toutefois parler de **décrypter** un message
  - La cryptologie (science du secret) regroupe deux domaines importants de la recherche en informatique
    - La cryptographie (écriture secrète)
    - La cryptanalyse (l'analyse de cette écriture secrète)

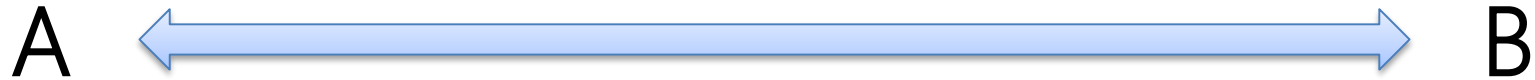
- Coder et décoder
  - Coder un message consiste à transformer ce message grâce à un code déterminé (par exemple pour le représenter en binaire)
  - Il n'y a pas de volonté de rendre ce message incompréhensible sauf pour son destinataire

Alice et Bob

- Deux "personnages" célèbres : Alice et Bob

- Schéma classique :

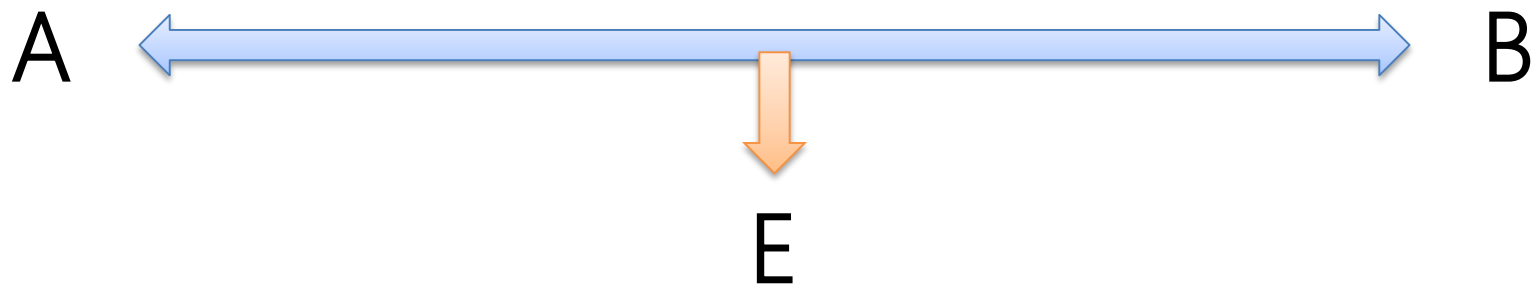
  « Alice et Bob souhaitent échanger des messages »



- Mais Alice et Bob ne sont pas seuls…

Alice et Bob (et Eve)

- Eve, l'écouteuse externe
  - De l'anglais *eavedropper* (personne qui écoute aux portes)



- Attaque passive
  - Eve peut écouter les échanges entre Alice et Bob
  - Mais elle ne peut pas les modifier

Alice et Bob (et Mallory)

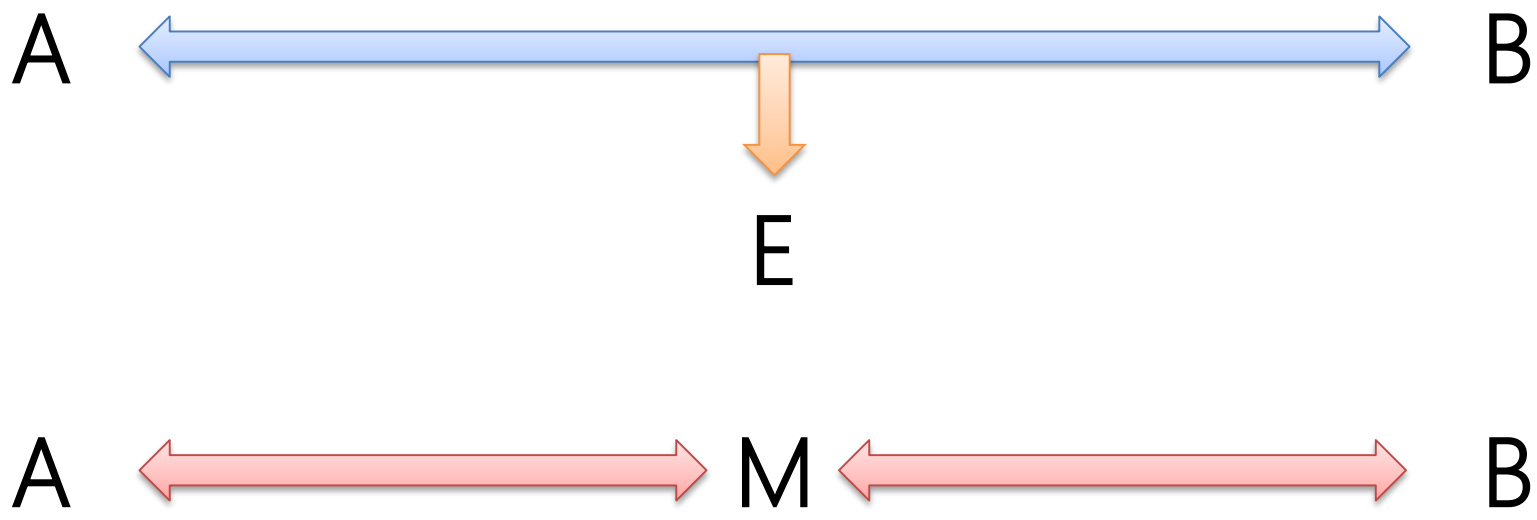
- Mallory, l'attaquant malveillant
  - De l'anglais *malicious*

- Attaque active
  - Mallory peut écouter les échanges entre Alice et Bob
  - Elle peut aussi modifier ces messages, envoyer les siens, remettre en jeu d'anciens messages, etc.

- Autres noms
  - Mallet
  - Oscar, l'adversaire (de l'anglais *opponent*)
  - Trudy, l'intrus (de l'anglais *intruder*)

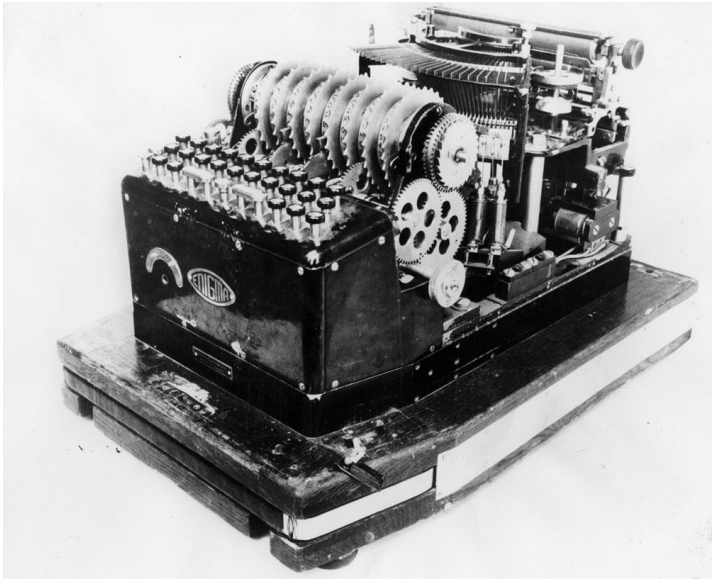
Alice et Bob

- Quelques objectifs de la cryptographie :
  - Comment empêcher Eve de comprendre les messages en transit ?
  - Comment s'assurer que les messages n'ont pas été modifiés par Mallory ?
  - Comment Alice peut-elle être sure qu'elle s'adresse bien à Bob, et pas à Mallory ?



Un exemple historique

- *Quelle est cette machine ?*



- Enigma, machine utilisée par les nazis pendant la seconde guerre mondiale pour chiffrer et déchiffrer leurs messages

- Les travaux de cryptanalystes polonais et britanniques (dont Alan Turing) ont permis aux Alliés de déchiffrer les messages

Premiers exemples de chiffrement

Chiffrement par décalage

- *« Zhqm, zmgm, zmfm. »*

  Jules César, 47 av. JC

- Le chiffre de César

  Chaque lettre est remplacée par celle située trois lettres plus loin dans l'alphabet

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	Y	Z
D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	Y	Z	A	B	C

- Nom officiel : chiffrement par décalage

Chiffrement par décalage

- Nombre de variantes : nombre de lettres dans l'alphabet

- Faiblesses
  - Si on sait qu'un chiffrement par décalage est utilisé, il suffit de tester quelques dizaines de possibilités (méthode *brute force*)
  - Analyse fréquentielle
    - Si on dispose d'assez de données et qu'on connait la langue utilisée
    - On peut étudier la fréquence de chaque symbole dans la version chiffrée
    - Ex : si c'est la lettre *h* qui revient le plus souvent dans la version chiffrée, c'est sans doute qu'elle chiffre la lettre *e* (la plus fréquente en français)
  - Analyse des mots probables
    - Pour Enigma : « Répétez », « Munition », « Météo », « RAS »

ALPHABET DÉSORDONNÉ

- Principe

  Chaque lettre est remplacée par une autre lettre de l'alphabet

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
O	E	S	G	J	Y	C	M	U	N	V	P	T	Z	I	Q	R	X	W	K	L	F	H	B	A	D

- Aussi appelé substitution mono-alphabétique

- Nombre de variantes

  $26!$ possibilités (soit plus de 4×10^{26})

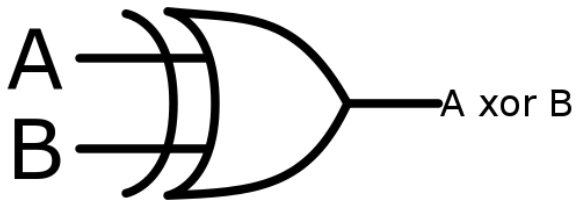
- Faiblesses :
  - Analyse fréquentielle
  - Analyse des mots probable

Méthode du masque jetable

- Méthode du masque jetable
  - Avant d'échanger le message, les deux interlocuteurs se mettent d'accord sur une clé, appelée masque
  - Ce masque définit les bits du message à inverser
  - Ce masque n'est utilisé que pour ce message (il est jeté après utilisation)
- Exemple

Message à chiffrer	0	1	0	0	1	0	0	0
Masque	0	0	1	0	1	0	1	0
Message chiffré	0	1	1	0	0	0	1	0
Masque	0	0	1	0	1	0	1	0
Message déchiffré	0	1	0	0	1	0	0	0

- "Ou exclusif" bit à bit
  - *L'un ou l'autre, mais pas les deux*
  - Porte logique XOR



Méthode du masque jetable

- Plus robuste :
  - Résiste à l'analyse fréquentielle
  - Résiste à l'analyse des mots probables
- Méthode utilisée par les diplomates et les services secrets au début du XXe siècle
- Faiblesses : pour chaque message, il faut
  - Générer un nouveau masque, aussi long que le message à transmettre et totalement aléatoire
  - Transmettre ce masque au destinataire (sans être intercepté)

Clé publique – clé privée

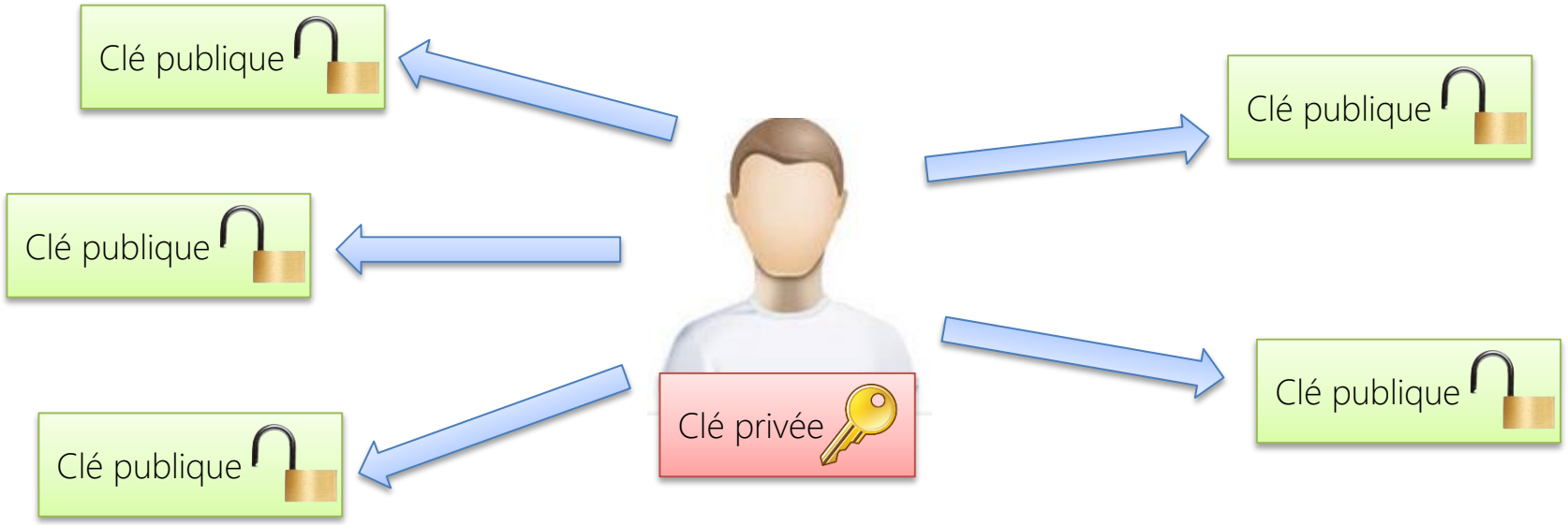
Clé publique – Clé privée

- **Problème** : Alice veut écrire à Bob, mais il ne leur est pas possible de se mettre d'accord sur une clé secrète

- Principe :
  1. Bob envoie un cadenas ouvert à Alice, mais il garde la clé
  2. Alice met son message dans une boite, et ferme la boite avec le cadenas
  3. Alice envoie la boite cadenassée à Bob
  4. Comme Bob est le seul à avoir la clé du cadenas, il est le seul qui peut ouvrir la boite

Clé publique – Clé privée

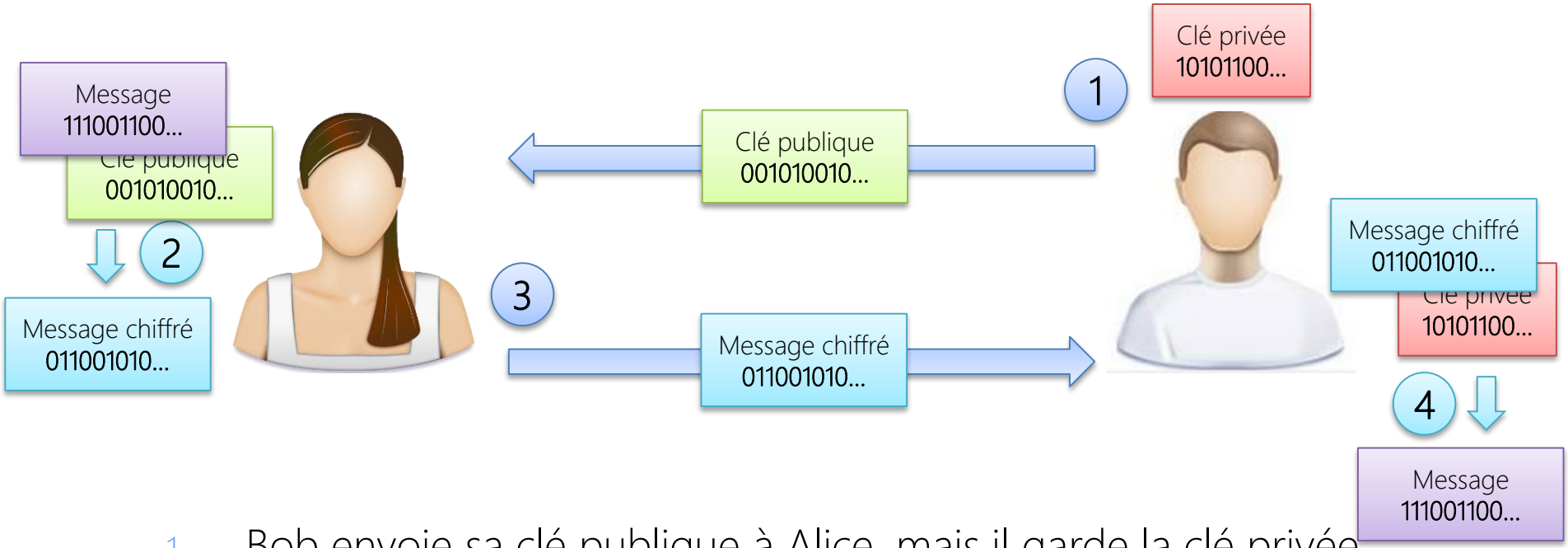
- Deux clés
  - Clé publique : des cadenas à la disposition de toute personne souhaitant contacter Bob
  - Clé privée : la clé qui ouvre ces cadenas



- On parle de méthodes à clé publique – clé privée.

Clé publique – Clé privée

- Avec des 0 et des 1



1. Bob envoie sa clé publique à Alice, mais il garde la clé privée
2. Alice chiffre son message grâce à la clé publique de Bob
3. Alice envoie le message chiffré à Bob
4. Comme Bob est le seul à avoir la clé privée, il est le seul à pouvoir déchiffrer le message

CHIFFREMENT SYMÉTRIQUE ET ASYMÉTRIQUE

- On peut distinguer deux grandes catégories de chiffrements
  - Les chiffrements symétriques
  - Les chiffrements asymétriques

- Dans une méthode de chiffrement symétrique, les deux interlocuteurs utilisent la même clé secrète

- Dans une méthode de chiffrement asymétrique, les deux interlocuteurs ne possèdent pas les mêmes clés.

- Le chiffrement symétrique étant plus rapide que le chiffrement asymétrique, on peut se servir du second pour échanger la clé secrète utilisée pour le premier.

RSA

- La méthode à clé publique – clé privée la plus utilisée est la méthode RSA

- Son nom vient de ses trois inventeurs
  - Ronald Rivest
  - Adi Shamir
  - Leonard Adleman

- Inventé en 1977, récompensé par le prix Turing en 2002

RSA – Création des clés

- Création des clés
  - On choisit deux nombres premiers distincts p et q
  - On calcule leur produit $n = pq$, appelé module de chiffrement
  - On calcule $\varphi(n) = (p-1)(q-1)$ (indicatrice d'Euler en n)
  - On choisit un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle exposant de chiffrement
  - On calcule l'entier d, inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle exposant de déchiffrement (pour ce calcul, on peut utiliser l'algorithme d'Euclide étendu)

- La **clé publique** est le couple (n, e)

- La **clé privée** est le couple (n, d)

RSA – Chiffrement et Déchiffrement

- Message à chiffrer : un entier M strictement inférieur à n.

- Chiffrement du message M grâce à la clé publique (n, e)

$$C = M^e \mod n$$

- Déchiffrement du message chiffré C grâce à la clé privée (n, d)

$$R = C^d \mod n$$

- Et bien entendu, $R = M$

RSA – Exemple

- Création des clés
  - $p = 7$ et $q = 11$
  - $n = pq = 7 \times 11 = 77$
  - $\varphi(n) = (p-1)(q-1) = 6 \times 10 = 60$
  - On choisit un entier $e = 53$ premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$
  - On calcule l'entier $d = 17$, inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$ (on a $e \times d \equiv 53 \times 17 \equiv 901 \equiv 1 \mod 60$)
  - La clé publique est $(77,53)$
  - La clé privée est $(77,17)$

- Chiffrement d'un message
  - Message : $M = 42$
  - Message chiffré : $C = 42^{53} \mod 77 = 14$
  - Message chiffré : $R = 14^{17} \mod 77 = 42$

Clé publique – clé privée

RSA ET NOMBRES PREMIERS

- Question : *Comment pourrait-on casser RSA ?*

- Solution :
  - Trouver les nombres premiers p et q tels que $n = pq$
  - En déduire $\varphi(n) = (p-1)(q-1)$
  - Connaissant e et $\varphi(n)$, en déduire d

- Difficultés :
  - Factoriser n comme un produit de deux nombres premiers
  - Tester la primalité de très grands nombres

RSA ET NOMBRES PREMIERS

- Plus un nombre n est grand, plus il est long d'en chercher une factorisation comme produit de nombres premiers.

- Aujourd'hui, les seuls algorithmes de factorisation que l'on connait ont une complexité exponentielle : le temps nécessaire pour factoriser n est proportionnel à k^n

- La sécurité d'un système RSA dépend donc grandement de la taille des clés utilisés :
  - Une clé de 256 bits peut être factorisée en quelques heures
  - Une clé de 1024 bits pourrait être cassée dans un avenir proche
  - On utilise des clés de 2048 ou 4096 bits

RSA ET NOMBRES PREMIERS

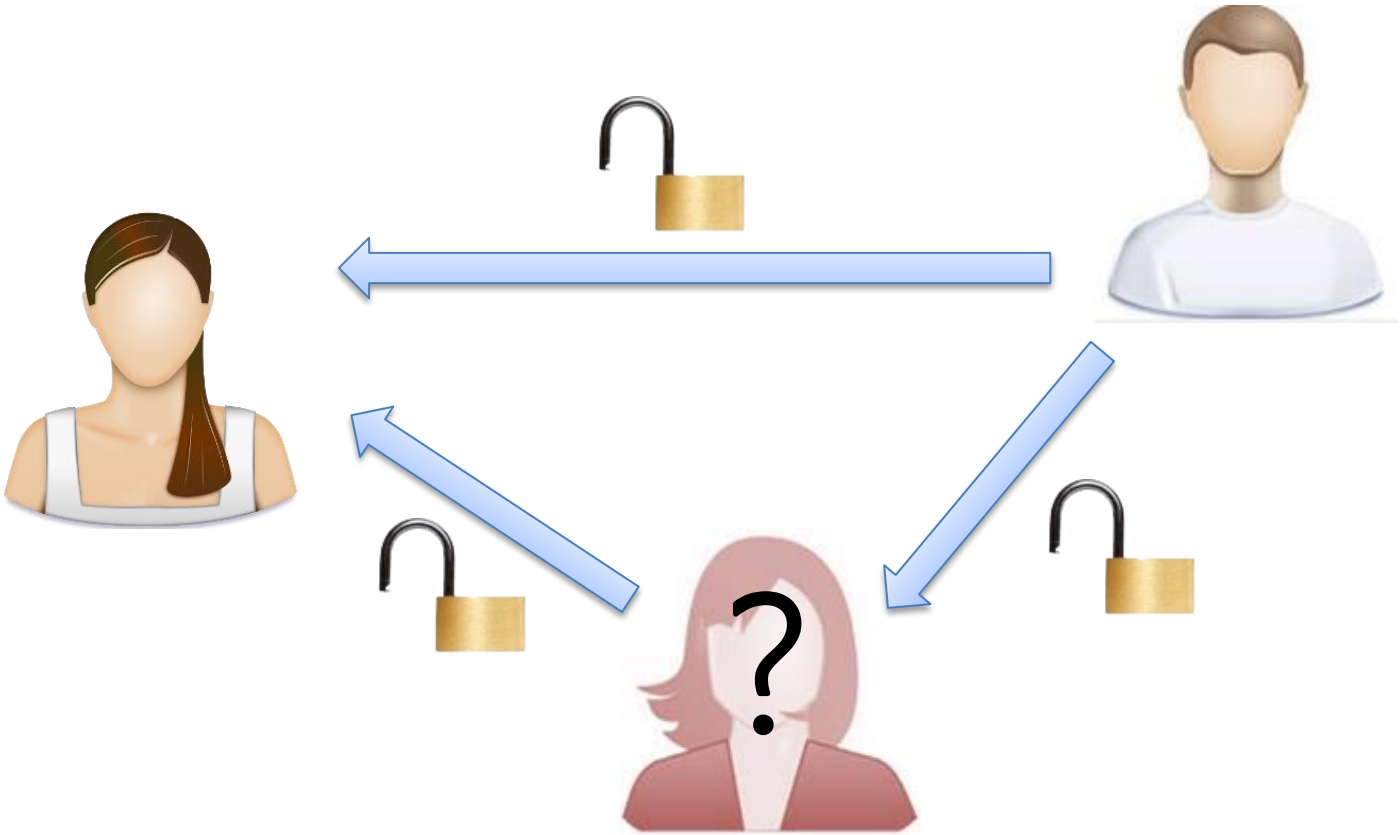
- Si l'on découvre un algorithme "rapide" de factorisation
  - Les systèmes basés sur ce chiffrements seront remis en cause
  - Ainsi que les données chiffrées jusque-là avec ces méthodes

- Une piste : l'algorithme de Shor
  - Ecrit en 1996
  - Capable de factoriser un nombre en temps non exponentiel
  - Utilisable uniquement sur un ordinateur quantique

- Ordinateurs quantiques
  - Calculateur basé sur les propriétés quantiques de la matière
  - Premiers modèles dans les années 1990
  - Difficulté majeure : réalisation physique de l'élément de base, le qubit

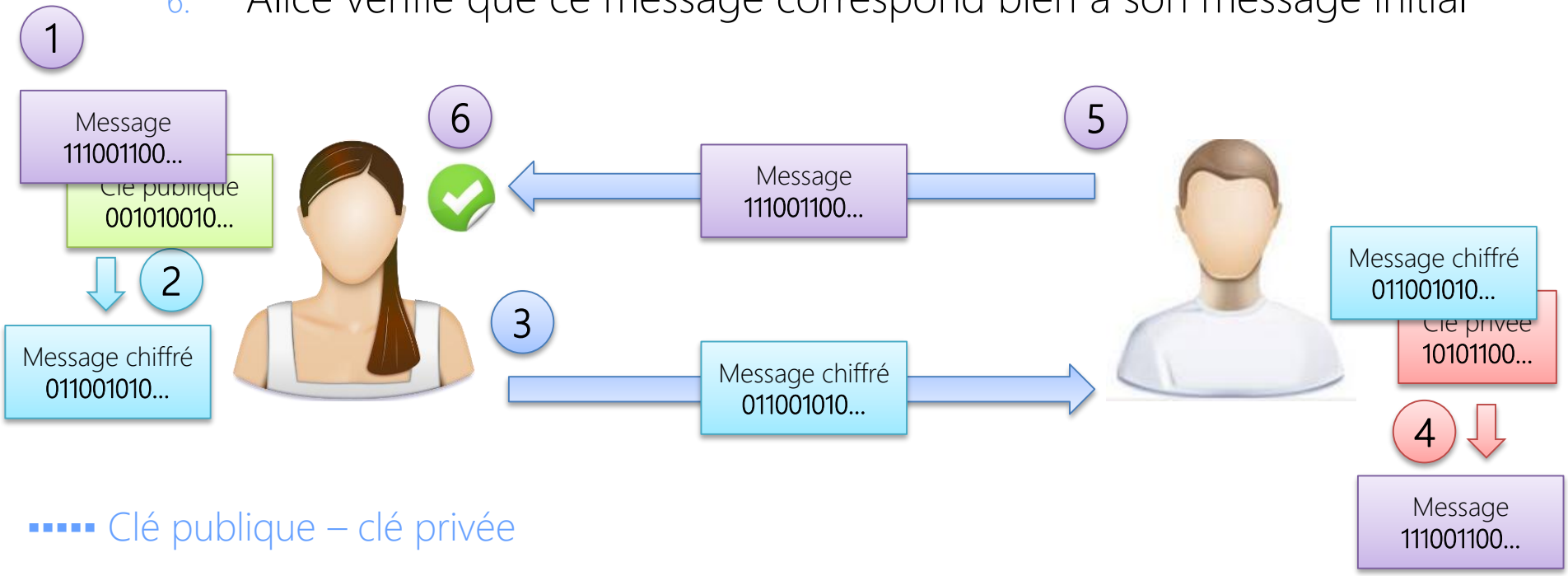
AUTHENTIFICATION

- Question : *Comment Alice peut-elle vérifier que la personne qui lui envoie la clé publique de Bob est bien Bob ?*



AUTHENTIFICATION

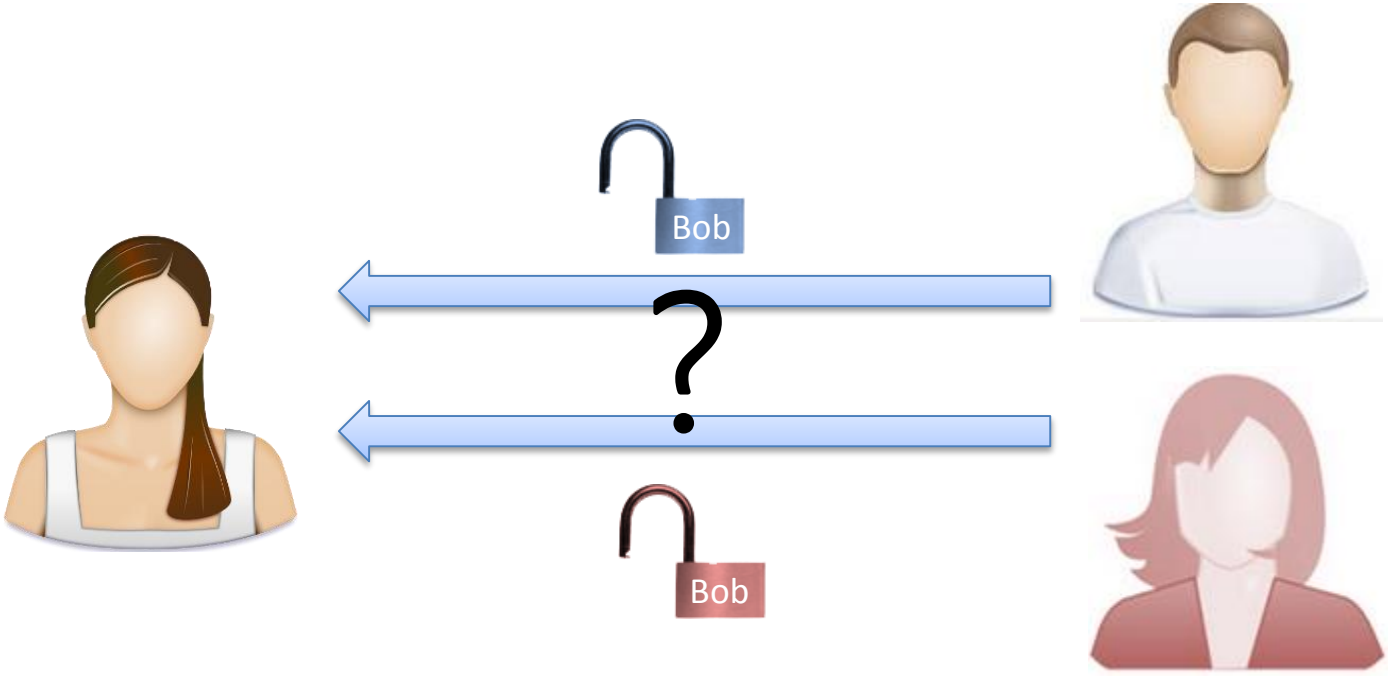
- Solution :
  1. Alice prépare un message
  2. Elle chiffre ce message en utilisant la clé publique de Bob
  3. Alice envoie ce message chiffré à Bob
  4. Bob déchiffre ce message avec sa clé privée
  5. Bob envoie ce message décrypté à Alice
  6. Alice vérifie que ce message correspond bien à son message initial



▪▪▪▪▪ Clé publique – clé privée

Authentification et Tiers de confiance

- **Question** : *Comment Alice peut-elle vérifier que la personne qui lui envoie une clé publique est bien Bob ?*



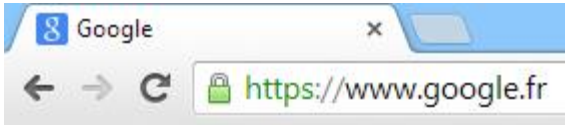
Authentification et Tiers de confiance

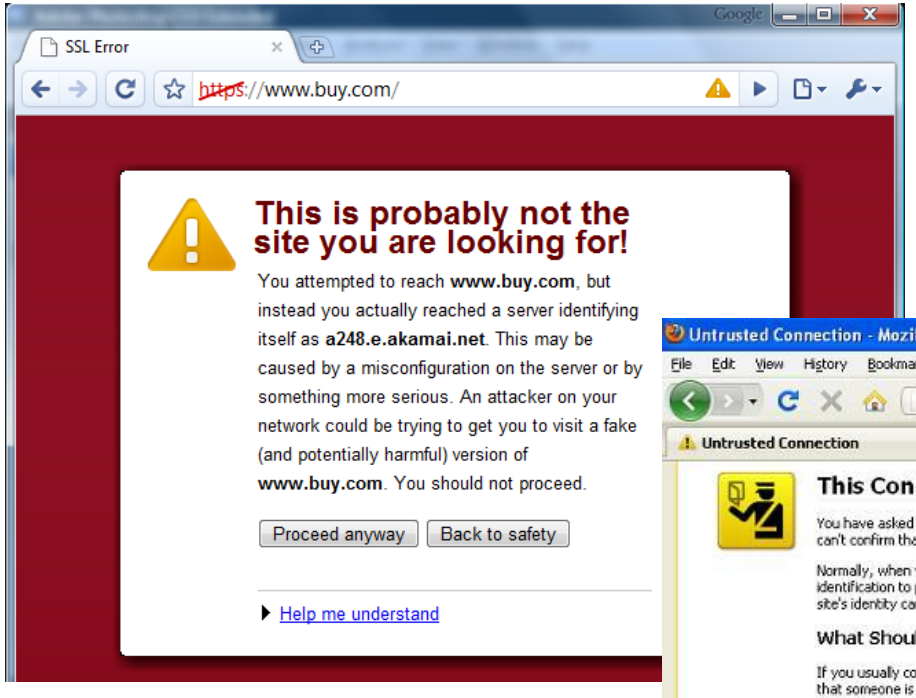
- Un tiers de confiance est un organisme connu à l'avance des deux interlocuteurs, et qui fait autorité pour authentifier les clés publiques

- Ces autorités utilisent des certificats, qui sont envoyés à la place des clés publiques.

AUTHENTIFICATION ET TIERS DE CONFIANCE

- **Exemple** : Un certificat HTTPS permet de vérifier l'identité d'un site internet

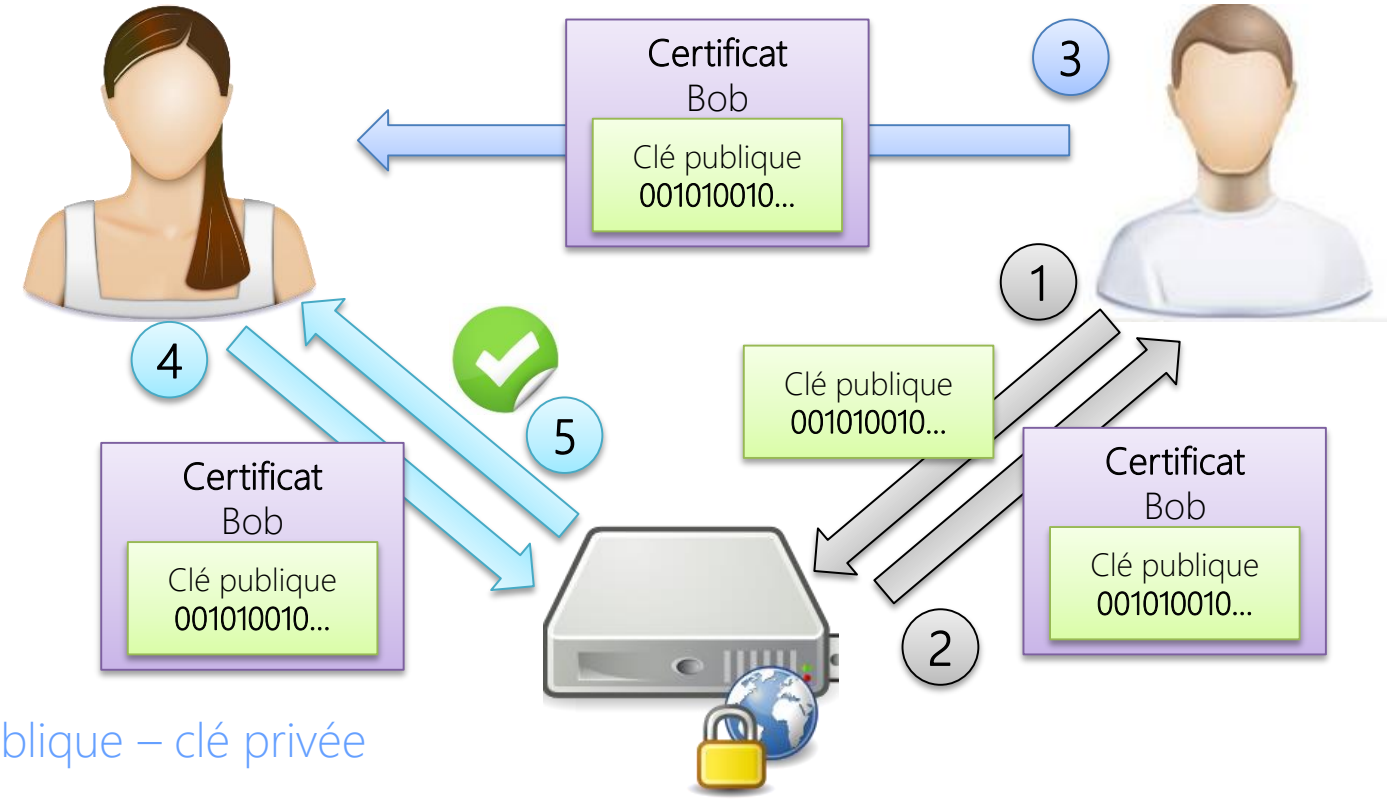




- Clé publique – clé privée

Authentification et Tiers de confiance

- Principe
  1. Bob envoie sa clé publique et des informations personnelles au tiers de confiance
  2. Le tiers de confiance vérifie qu'il s'agit bien de Bob, et délivre un certificat qu'il transmet à Bob (ce certificat inclut sa clé publique)
  3. Bob transmet ce certificat à Alice
  4. Lorsqu'Alice reçoit le certificat venant de Bob, elle le transmet au tiers de confiance
  5. Le tiers de confiance vérifie les informations du certificat et informe Alice du résultat



▪▪▪▪▪ Clé publique – clé privée

Exercices

Exercices

- **Exercice 1.** Chiffrez un message grâce au chiffre de César et demandez à un de vos voisins de le déchiffrer

- **Exercice 2.** Utilisez la méthode du masque jetable pour échanger un message sur 16 bits avec l'un de vos voisins

- **Exercice 3.** Pour retirer de l'argent à un distributeur avec une carte bancaire, il est nécessaire d'entrer un code à 4 chiffres
  - Combien existe-t-il de combinaisons possibles ?
  - Quel est le nombre moyen d'essais nécessaire pour trouver la bonne combinaison ?
  - En cas d'erreur sur la première saisie, un délai d'une seconde est imposé entre avant la saisie suivante. En supposant que ce délai soit multiplié par 10 à chaque échec, quel est le temps moyen nécessaire pour trouver la bonne combinaison ?

- **Exercice 4.** Testez l'algorithme RSA avec de petits entiers (par exemple en prenant $p = 3$ et $q = 5$)

PROCHAINE SÉANCE

Vendredi 6 février

[TD] MÉTHODES DE CHIFFREMENT