Sécurisation des communications

1. Introduction

La **sécurisation des échanges est un enjeu majeur** de l'économie moderne. Sans elle il serait impossible de mettre en œuvre une économie globalisée. Elle repose sur la cryptographie (écriture secrète) et la cryptanalyse (analyse de cette dernière).

La cryptographie est pourtant une science très ancienne, on en trouve des traces 2 000 ans avant notre ère en Égypte ancienne.

Pour protéger les communications, on chiffre les messages à l'aide d'une méthode de cryptographie, comportant une ou plusieurs clés secrètes. Les deux principes de chiffrement sont le **chiffrement symétrique**, où la même clé est utilisée pour le chiffrement et le déchiffrement du message, et le **chiffrement asymétrique**, où deux clés différentes sont utilisées. Reste à assurer que les clés ne soient pas piratées, ce qui est le problème de l'authenticité des utilisateurs. La sécurité des communications dépend à la fois des méthodes de chiffrement et des protocoles d'échange des clés.

2. Vocabulaire

- **Cryptographie** : c'est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés.
- **Clé de chiffrement** : c'est un paramètre utilisé en entrée d'une opération cryptographique.
- **Coder** : représenter de l'information par un ensemble de signes prédéfinis. On utilise parfois le verbe encoder.
- **Décoder** : interpréter un ensemble de signes pour extraire l'information qu'ils représentent.
- **Chiffrer** : rendre une suite de symboles incompréhensible au moyen d'une clé de chiffrement.
- **Déchiffrer/décrypter** : retrouver la suite de symboles originale à partir du message chiffré. On utilise le terme déchiffrer lorsque l'on utilise une clé de déchiffrement pour récupérer le texte initial et le terme décrypter lorsque l'on arrive à déterminer le message original sans utiliser la clé.
  Coder et décoder s'utilisent lorsqu'il n'y a pas de secret. Par exemple, on parle de "codage en complément à deux" des entiers. C'est juste une façon de représenter les entiers positifs ou négatifs par une suite de bits. N'importe qui peut décoder la suite de bits pour déterminer l'entier. Le terme "coder" (qui est un anglicisme, to code) est aussi utilisé de façon informelle comme synonyme de « programmer » comme dans la phrase "j'ai codé cette application". L'anglicisme "crypter" est à proscrire ; on utilisera "chiffrer".
- **Décrypter par force brute** : cela consiste à essayer toutes les clés possibles pour décrypter le message. Un des objectifs de la cryptographie moderne est de rendre cette méthode inefficace par un nombre toujours plus grand de clés possibles

3. Chiffrement symétrique

3.1. Définition

Un chiffrement est dit symétrique si les clés de chiffrement et de déchiffrement sont identiques.

Il existe de nombreux algorithmes de chiffrement symétrique parmi lesquels on peut citer :

- DES (Data Encryption Standard) avec clés de 56 bits, soit 2^{56} clés possibles. Il est obsolète de nos jours.
- AES (Advanced Encryption Standard) avec clés de 128 bits minimum, soit$2^{128} \approx 3{,}4 \cdot 10^{38}$ clés possibles au minimum. C'est l'un des standards actuels.

Avec l'accroissement de la puissance de calcul des ordinateurs et l'amélioration des techniques de décryptage, certains de ces algorithmes sont devenus obsolètes (DES). Plus les ordinateurs se perfectionnent, et plus les algorithmes s'affaiblissent, obligeant à étudier de nouvelles solutions. C'est ainsi que la cryptographie évolue pour vaincre la cybermenace. Une clé est représentée par un nombre de bits fixés. Pendant longtemps une clé d'une

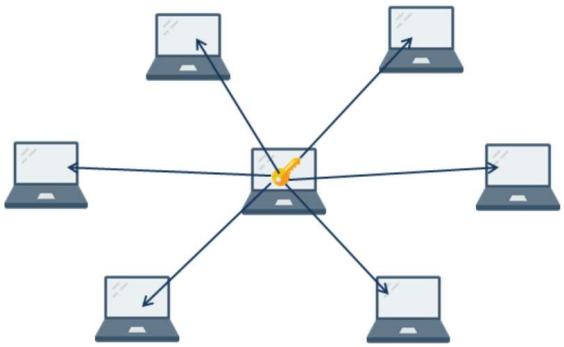
quarantaine de bits a été suffisante mais de nos jours, le standard est maintenant de 128 bits mais peut monter jusqu'à 4 096 bits.

3.2. Fonctionnement

Dans le chiffrement symétrique, la clé dite partagée, est commune à l'ensemble des interlocuteurs. Elle est communiquée à l'ensemble de ceux-ci, par un moyen "sûr". Par exemple en rencontrant physiquement les interlocuteurs et en leurs donnant la clé sur un support amovible. Cette méthode reste peu pratique c'est pourquoi on utilise plutôt un système de cryptographie asymétrique pour diffuser cette clé.

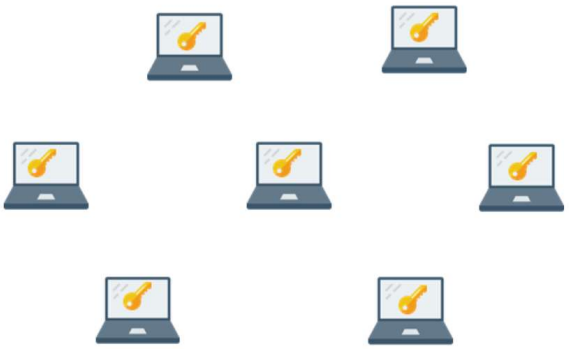
Étape 1

Un ordinateur du réseau créé une clé et la diffuse à tous les ordinateurs qui doivent communiquer de manière sécurisée entre eux.



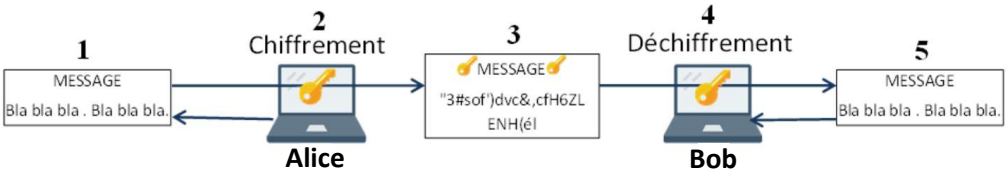
Étape 2

Chaque ordinateur du réseau stocke la clé partagée qu'il a reçu.



Étape 3

Communication chiffrée entre deux ordinateurs.

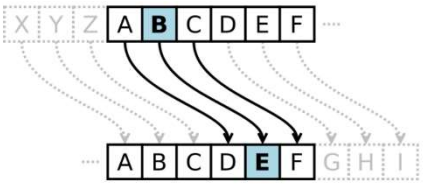


1. Alice rédige son message en clair.
2. Alice utilise la clé partagée et chiffre son message.
3. Alice envoie son message chiffré à Bob.
4. Bob utilise la clé partagée pour déchiffrer le message.
5. Bob peut lire le message en clair.

3.3. Chiffre de César

Un des exemples le plus connu et le plus ancien de chiffrement symétrique est le chiffre de Jules César. Il consiste en un simple décalage des lettres dans l'alphabet, d'un nombre de lettres convenu à l'avance, la clé.

Le chiffrement de César du message : HELLO WORLD avec une clé de 3 lettres (A est chiffré D, B est chiffré E…) donne le message chiffré : KHOOR ZRUOG



Le code de César n'est pas très fiable car comme il ne comporte que 25 clés (le décalage peut aller de 1 à 25), il est facile de le décrypter par force brute.

3.4. Chiffrement XOR

Cette méthode repose sur l'opérateur logique XOR (ou exclusif), noté ⊕ et sur la répétition d'une clé alphanumérique.

Table de vérité de l'opérateur XOR entre deux bits E1 et E2 (ci-contre).

E1	E2	S
0	0	0
0	1	1
1	0	1
1	1	0

Propriété de XOR : soit trois nombres entiers x, y et z (codés sur 8 bits par exemple) tels que **x ⊕ y = z** alors on a aussi : **z ⊕ x = y** et **z ⊕ y = x**.

Soit le message **`Hello World!`** qui donne en binaire :
`01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100 00100001`
Chaque octet correspond au code ASCII de chaque caractère que l'on peut directement obtenir avec le site : https://www.rapidtables.com/convert/number/ascii-to-binary.html.

Soit le mot **`octet`** qui va servir de clé de chiffrement. Il donne en binaire :
`01101111 01100011 01110100 01100101 01110100`

Pour chiffrer le message, il faut effectuer un XOR bit à bit. Comme la clé est plus courte que le message, il faut "reproduire" la clé vers la droite autant de fois que nécessaire (si la taille du message n'est pas un multiple de la taille de la clé, on peut reproduire seulement quelques bits de la clé pour la fin du message).

```
  01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100 00100001
⊕ 01101111 01100011 01110100 01100101 01110100 01101111 01100011 01110100 01100101 01110100 01101111 01100011
  ----------------------------------------------------------------------------------------------------------
  00100111 00000110 00011000 00001001 00011011 01001111 00110100 00011011 00010111 00011000 00001011 01000010
```

Le code obtenu, en binaire, est :
`00100111 00000110 00011000 00001001 00011011 01001111 00110100 00011011 00010111 00011000 00001011 01000010`
Le message obtenu est alors : 'O4B

Le déchiffrement se fait à l'aide de la clé (**`octet`**) en réalisant à nouveau un XOR entre le message codé et la clé.

```
  00100111 00000110 00011000 00001001 00011011 01001111 00110100 00011011 00010111 00011000 00001011 01000010
⊕ 01101111 01100011 01110100 01100101 01110100 01101111 01100011 01110100 01100101 01110100 01101111 01100011
  ----------------------------------------------------------------------------------------------------------
  01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100 00100001
```

Après vérification le message décodé est bien : **`Hello World!`**

4. Chiffrement asymétrique

4.1. Définition

Un chiffrement est dit asymétrique si les clés de chiffrement et de déchiffrement sont différentes.

Pour un chiffrement asymétrique l'une des clés est appelée **clé publique**, l'autre **clé privée**. La clé publique est diffusée sans chiffrage sur le réseau et peux être interceptée par un tiers (malveillant le plus souvent) sans que cela ne soit préjudiciable. La clé privée, quant à elle, est générée de différentes manières en fonction du système utilisé et ne sera jamais diffusée en clair sur le réseau.

Là encore, il existe de nombreux algorithmes de chiffrement asymétrique parmi lesquels on peut citer :

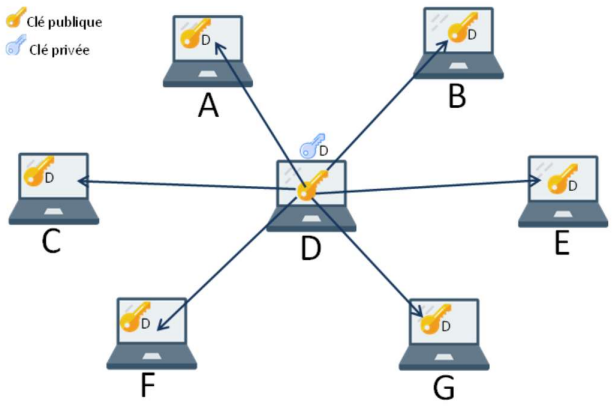
- Puzzles de Merkle (1974).
- Méthode de Diffie-Helman (1976).
- Le système RSA (Ronald Rivest, Adi Shamir et Leonard Adelman) (1977).

4.2. Principe de fonctionnement du système RSA

La clé publique dans ce système, est diffusée à tous les ordinateurs du réseau souhaitant communiquer avec celui qui l'a créé. La clé privée, quant à elle, reste sur la machine de son créateur.

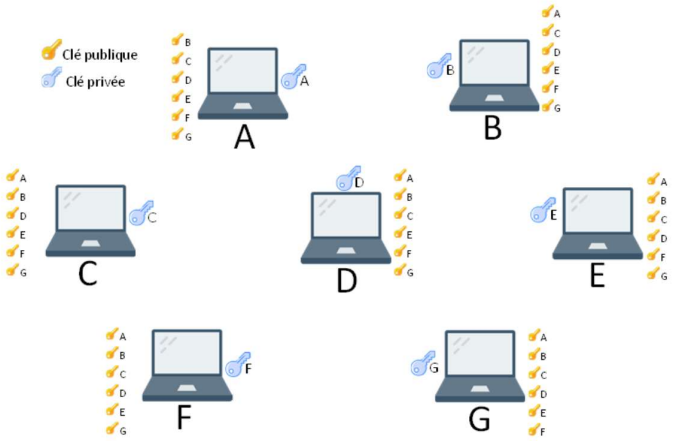
Étape 1

Un ordinateur du réseau créé une clé publique et la diffuse à tous les ordinateurs qui doivent communiquer de manière sécurisée avec lui. Il créé aussi une clé privée qu'il conserve dans sa mémoire.



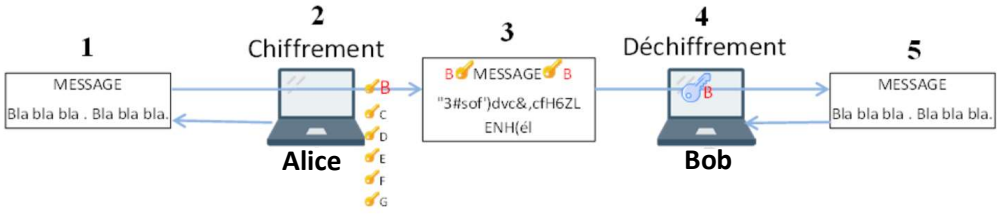
Étape 2

Lorsque chaque ordinateur du réseau a fait de même, toutes les machines sont en possession de nombreuses clés publiques et d'une seule clé privée (celle qu'il a généré lui-même).



Étape 3

Envoi d'un message d'Alice à Bob. La clé publique sert à chiffrer un message et la clé privée à le déchiffrer.



1. Alice rédige son message en clair.
2. Alice utilise la clé publique de Bob pour chiffrer son message.
3. Alice envoie son message chiffré à Bob.
4. Bob utilise sa clé privée pour déchiffrer le message.
5. Bob peut lire le message en clair.

4.3. Limite du chiffrement asymétrique

Le chiffrement asymétrique étant beaucoup plus robuste que le chiffrement symétrique on peut être tenter de se passer de ce dernier. Cependant le chiffrement asymétrique reposant sur des calculs beaucoup plus complexes que ceux du chiffrement symétrique, il est beaucoup trop long de chiffrer l'intégralité d'un message par chiffrement asymétrique. On préfère donc, comme pour le protocole HTTPS décrit par la suite, utiliser un chiffrement asymétrique pour diffuser de manière sécurisé une clé de chiffrement symétrique. Celle-ci est ensuite utilisée pour chiffrer les communications sans risque qu'elle soit utilisée par un tiers.

5. Le protocole sécurisé HTTPS

5.1. HTTP + TLS = HTTPS

La cybersécurité est un enjeu mondial !

En effet de plus en plus d'achats, de démarches administratives ou bancaires, de communications sensibles se font désormais par le web. Il est donc nécessaire d'avoir plusieurs garanties :

- le chiffrement des données de bout en bout ;
- l'authentification de l'interlocuteur ;
- l'intégrité des données (que les données demandées ou reçues soient bien les bonnes).

Il a donc été nécessaire de sécuriser le **protocole HTTP** (**HyperText Transport Protocol**) en lui adjoignant un protocole de sécurisation, le **protocole SSL** (**Secure Sockets Layer**) dans un premier temps, puis à partir des années 2000, le **protocole TLS** (**Transport Layer Security**). La juxtaposition de ces deux protocoles donne le fameux **HTTPS** (**HyperText Transport Protocol Secure**) qui permet une communication sécurisée sur le web. Dans un navigateur web, le protocole HTTPS est indiqué au début de la barre d'adresse et précédé par un cadenas.

Ce protocole de sécurisation doit garantir la prise en compte de plusieurs caractéristiques :

- Une parfaite compatibilité avec le protocole http.
- Une conception évolutive permettant de prendre en compte l'augmentation des performances des ordinateur. On pourra par exemple agir facilement sur la taille de la clé de chiffrement.
- Des performances permettant au site web l'utilisation fréquente de ce protocole de sécurisation.

5.2. Certificats et tiers de confiance

Pour sécuriser totalement une communication entre deux interlocuteurs, il ne suffit pas de chiffrer l'information. Il est aussi nécessaire de s'assurer que l'information parvienne bien au correspondant en s'assurant de son identité. Les Autorités de Certification (AC) sont des entités habilitées à délivrer des certificats (numériques) attestant de l'identité d'un correspondant (on peut comparer ces AC à l'état français qui délivre à chaque citoyen une carte d'identité, le certificat). Ces AC sont des entreprises spécialisées, des associations à but non lucratifs ou des états. Les certificats reposent sur le principe du chiffrement asymétrique vu précédemment. Généralement ils contiennent les informations suivantes :

- l'identifiant de l'AC qui signe le certificat ;
- l'identifiant de l'entité certifiée ;
- la date de validité du certificat ;
- la clé publique de l'entité certifiée ;
- l'algorithme utilisé pour la signature du certificat ;
- la signature du certificat.

Les systèmes d'exploitation et les navigateurs web possèdent une copie de chacune des clés publiques des AC

5.3. Détail du protocole HTTPS

Exemple

1. Un client d'une banque souhaite effectuer un virement via le site de celle-ci. Voici le fonctionnement de la couche TLS du protocole HTTPS permettant de s'assurer de l'identité de son correspondant :on la qualifie souvent de "**poignée de main TLS**".
2. Le client envoie un message initial (nommé "Hello") au serveur (supposé) de la banque.
3. Le serveur envoie sa réponse contenant le certificat que lui a délivré l'AC.
4. Le client vérifie le certificat au moyen de la clé publique de l'AC contenue dans son navigateur ainsi que la date de validité de celui-ci.
5. Le client choisit une clé de chiffrement symétrique qui servira de clé de session pour chiffrer les communications lors du virement. Il chiffre cette clé de session à l'aide de la clé publique de sa banque puis la transmet au serveur de cette dernière.
6. Le serveur déchiffre la clé de session à l'aide de sa clé privé (qu'il est le seul à posséder) Le client et le serveur possèdent donc maintenant une clé de chiffrement-déchiffrement symétrique, transmise de façon sécurisée et qu'ils vont pouvoir utiliser tous deux pendant toute la durée de la communication.