	Architecture matérielle	**NSI T^{ale}**
	Sécurisation des communications	Cours/TD

*

1. Introduction et historique

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).

Dans la période actuelle on y trouve aussi d'autres aspects :

- l'authentification,
- la non-répudiation,
- l'intégrité,
- l'anonymat,

Toutes ces composantes reposent sur la la **cryptographie asymétrique** (ou à clé publique) dont l'origine est récente (*Whitfield Diffie* et *Martin Hellman*, 1976).

La cryptographie est pourtant une science très ancienne, on en trouve des traces 2000 ans avant notre ère en Égypte ancienne.

2. Bref historique

2.1. Période ancienne : avant les calculateurs

L'objectif était déjà le même : assurer qu'un message ne puisse être lu que par son destinataire.
Les techniques mises en œuvre étaient très variées : substitutions de lettres, langues secrètes, stéganographie (faire passer un message inaperçu dans un autre support) etc.

Toutes reposaient sur le choix d'un procédé avant l'échange de messages
La connaissance du procédé était parfois suffisante pour décrypter (=casser) le message

Un peu de vocabulaire :
Message clair : message que tout le monde peut lire :
"Les navires arrivent à minuit"

Chiffrer ou coder : appliquer un procédé de chiffrement à un message :
"KZQ BPCUEZ PEEUCZBR P LUBUR"

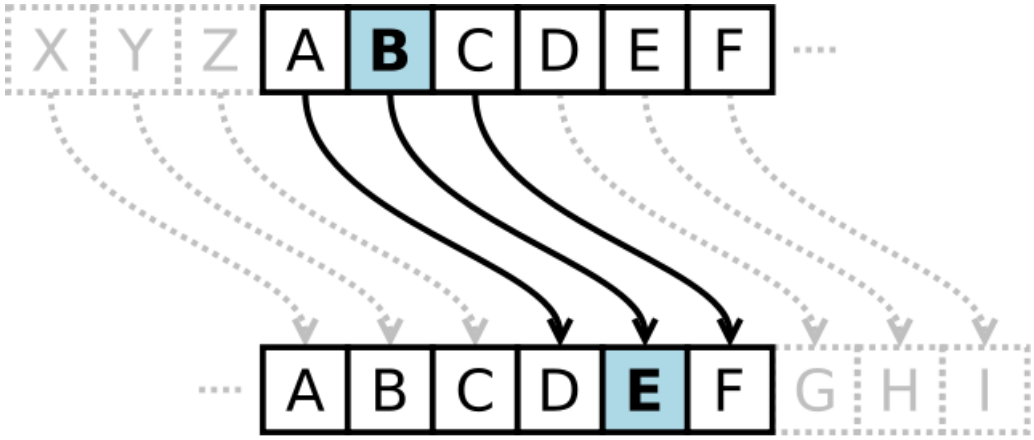
Décoder ou déchiffrer : lire le message à l'aide l'algorithme et de la clé. C'est l'objectif.

Décrypter ou casser : lire le message sans connaissance de la clé. C'est ce qu'on souhaite éviter.

Quelques méthodes restées célèbres

Durant la période ancienne on rivalise d'ingéniosité.

- Le code César (60 av. J-C.) : simple décalage de lettres.



- le chiffrement affine dans lequel on applique des transformations mathématiques simples mais où toutes les lettres sont transformées de la même manière.
- l'analyse des fréquences qui s'appuie sur la fréquence des lettres dans une langue et brise sans peine les deux chiffres précédents (~1300).
- Le chiffre de Vigenère, qui a résisté 300 ans aux assauts des analystes.

Au XIX^{ème} siècle les premières machines réalisant des calculs compliquent la tâche. Il faut trouver des procédés qui leur résistent.

On s'intéresse alors aux grands principes. L'un d'eux émerge et reste d'actualité :

- la sécurité d'un système ne doit pas reposer sur le secret de la méthode de chiffrement (*Kerckhoffs*, 1883).

2.2.Au XX^{ème} siècle

La cryptanalyse prend d'abord le dessus : le succès des alliés durant la première guerre mondiale est indissociable de leur aptitude à décrypter rapidement les messages allemands.

Durant la seconde guerre mondiale, les Allemands ont longtemps eu le dessus, leurs messages, chiffrés à l'aide d'une machine appelée ***Enigma*** semblaient impénétrables. C'est l'équipe d'**Alan Turing** qui parviendra à les rompre.
On réalise alors qu'il est nécessaire d'inventer de nouvelles méthodes.

L'apparition des calculateurs rend de nouvelles méthodes possibles, reposant sur la possibilité de donner à un message l'allure de l'aléatoire.

- Bonjour Robert → 3855087441179157013228464900911563060250023l6

Toutes les méthodes sont encore symétriques, l'émetteur et le receveur partagent une même clé qui sert au chiffrement et au déchiffrement.

Cette clé doit bien avoir été communiquée avant le premier message…
Mais comment ?
Comment établir de nouvelles communications lointaines sans qu'une clé ne transite en clair ?

La révolution vient de **Diffie** et **Hellman** qui proposent **le premier échange sécurisé de clé**. Il repose sur l'existence de fonction à sens unique ou à brèche secrète.

3. Fonction à sens unique : exemple simpliste

L'exemple le plus connu repose sur **le produit et la factorisation** :

- Multiplier deux entiers : facile. Toutes les machines savent faire.
- Retrouver les facteurs à partir du produit : très difficile. Les machines savent faire mais sont extrêmement lentes.

Sens unique :

- Aisément calculée, difficile à inverser.
- `n=263467`. `n` est le produit de deux entiers `p` et `q`
- Lesquels ?

Brèche secrète :

- Si vous connaissez l'un des facteurs… alors c'est facile.
- `p=487`.
- Donc `q=n/p=541`.

4. La révolution : Diffie et Hellman

En **1976**, **Diffie** et **Hellman** proposent un protocole de communication qui permet à deux interlocuteurs d'établir une communication sécurisée alors qu'ils sont distants et que tout le monde peut intercepter leurs messages.
Ce protocole est encore employé. Régulièrement amélioré on en a conservé le principe.

Il permet d'établir une communication secrète SANS qu'une clé ne doive transiter en clair sur un réseau.

C'est l'invention de la **cryptographie asymétrique**.

5. Différents types de clé

La **cryptographie symétrique** repose sur une seule clé :

- Une clé secrète permet d'encoder et décoder un message. Elle ne doit être connue que de l'émetteur et du récepteur.

La **cryptographie asymétrique** utilise deux clés :

- Une **clé publique** permet d'encoder un message. **Tout le monde la connaît**.
- Une **clé privée** permet de déchiffrer un message. **Seul vous la connaissez**.

Un exemple de communication très simplifié avec un chiffrement asymétrique

Afin qu'on puisse lui écrire, Robert a généré deux clés :

- une publique qu'il rend accessible,
- une privée qu'il conserve.

- J'écris à Robert : j'encode avec sa clé publique. Qu'est-ce que j'encode ?
  - ✗ "J'ai fait du cassoulet"
- Robert décode le message à l'aide de sa clé privée.
- Robert encode sa réponse avec **MA** clé publique et me l'envoie.
- Je décode ma clé privée etc.

Plusieurs défauts à cette méthode

Le chiffrement asymétrique est lent,

- Nous ne sommes pas à l'abri d'une amélioration des sciences qui rendraient obsolètes certaines méthodes.
- L'informatique quantique promet de factoriser très rapidement les entiers. Le chiffrement **RSA** (le plus couramment employé) serait alors inutile.

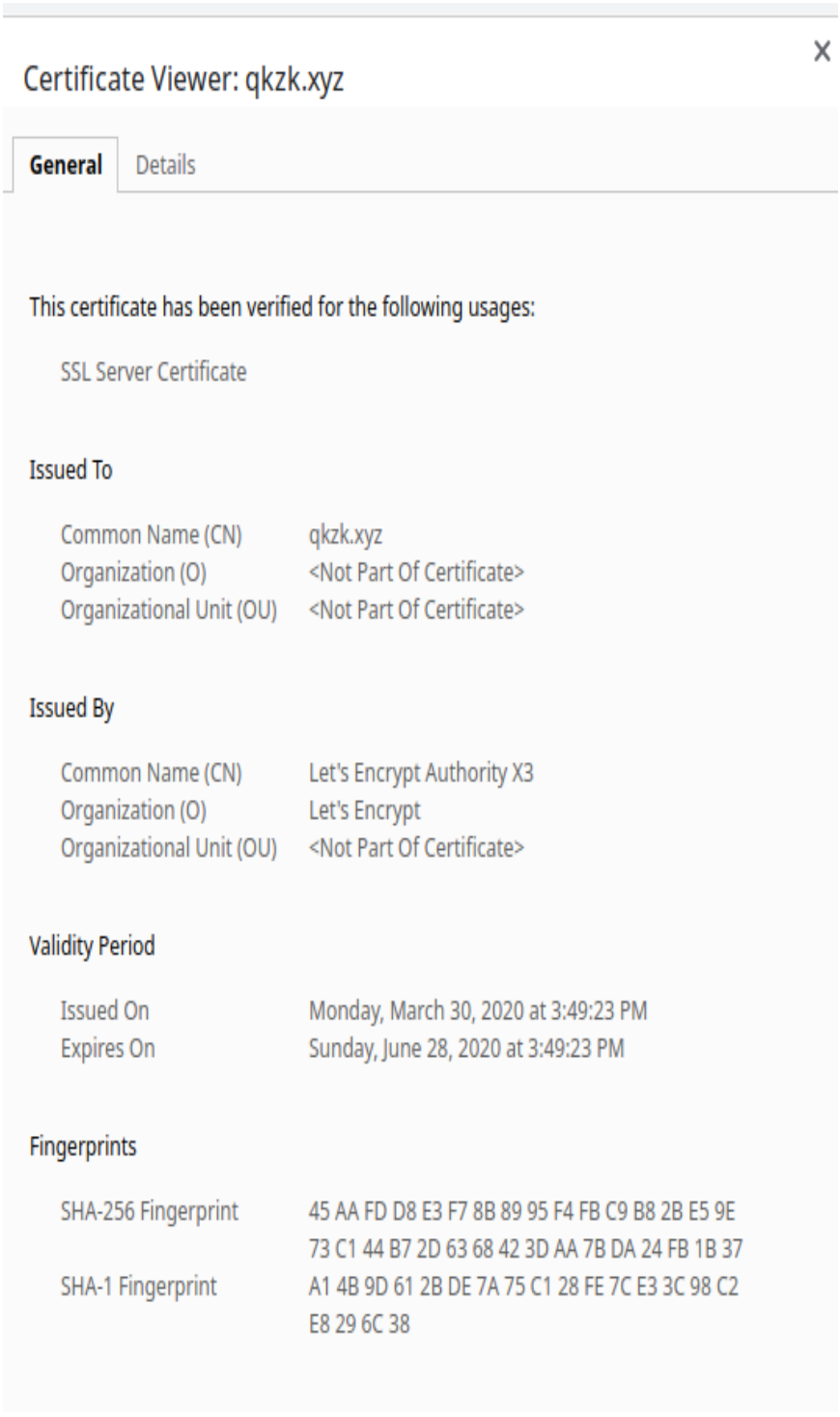
Amélioration considérable de la méthode

- J'écris à Robert : j'encode avec sa clé publique. Qu'est-ce que j'encode ?
  - ✗ UNE CLÉ SECRÈTE
- Robert décode le message à l'aide de sa clé privée.
- Il lit la clé secrète. Nous sommes les seuls à la connaître.
- Robert encode sa réponse avec un algorithme symétrique rapide et fiable à clé secrète. Il peut envoyer ce message car nous seuls disposons de la clé.
- Je reçois le message et le décode avec la clé secrète.

On comprend bien qu'il est à la fois possible et préférable de **combiner chiffrement asymétrique** (pour établir une communication) **et chiffrement symétrique** (une fois qu'elle est établie).

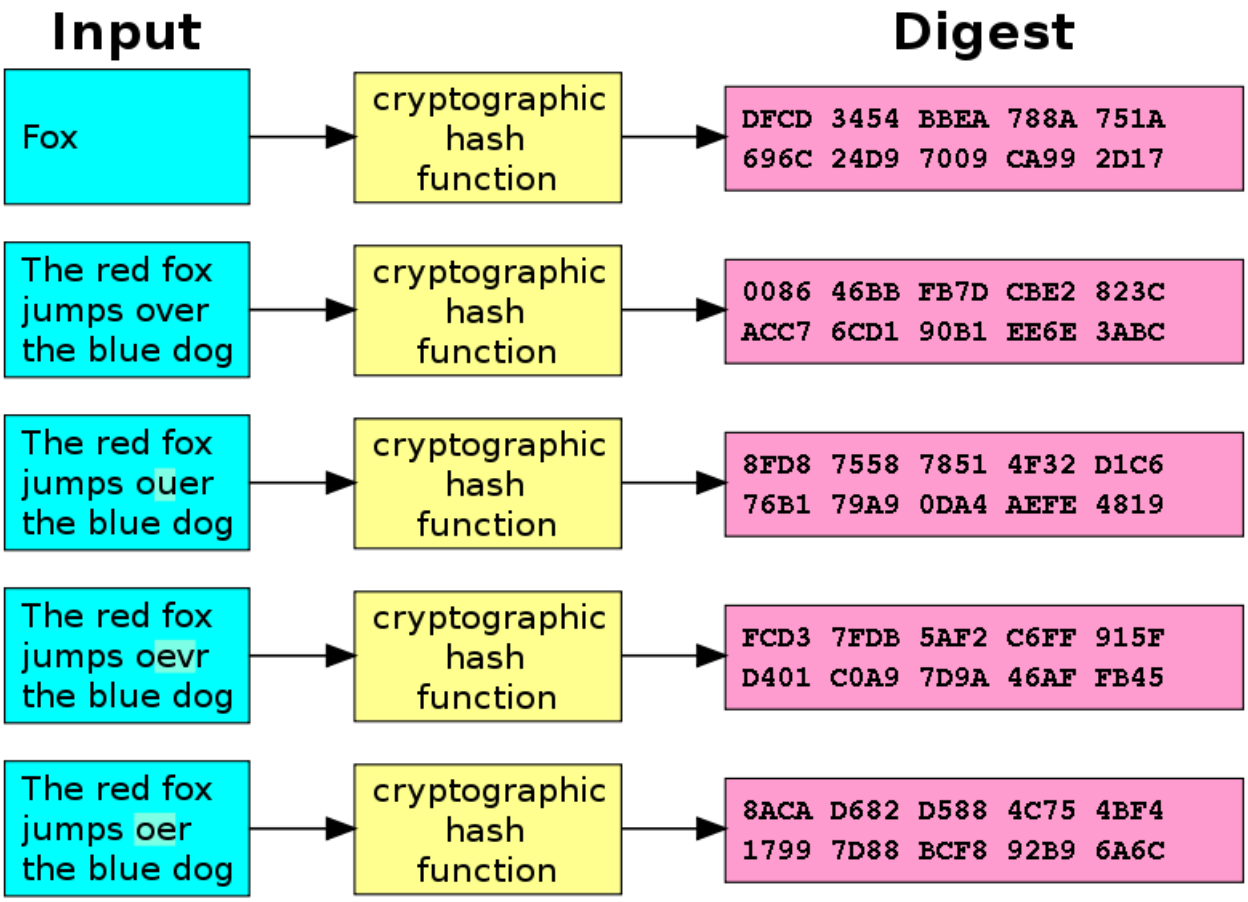
6. Autres enjeux de la sécurisation

Mon interlocuteur est-il celui qu'il prétend être ?
C'est l'**authenticité** (certificats pour SSL et TLS employés par HTTPS).



Ce message que je reçois a-t-il été altéré par un tiers ?
C'est l'**intégrité** (fonctions de hachages cryptographiques : SHA, MD5)

- Un contenu (fichier, message…) est transformé (très rapidement) en un mot de taille fixe.
- Changer un symbole du mot suffit à produire un résultat très différent.
- Cette transformation est impossible à inverser.
- On envoie alors le fichier initial,
- Le hash est rendu public.
- Quiconque modifie l'un doit changer l'autre.



7. Quelques algorithmes modernes

- **RSA** : **asymétrique**. principalement utilisé pour établir une communication symétrique (**HTTPS** etc.). Repose sur l'arithmétique.
- **AES** (ou **Rijndael**): **symétrique**. Repose sur des transformations bit par bit réversibles qui donnent au message une apparence aléatoire. Ses prédécesseurs connus sont **TKIP** (**wifi WPA**) et **DES** (qui ne résiste plus aux attaques brute force).
- **SERPENT**, **Blowfish**, **Twofish** **symétriques** : concurrents moins employés d'**AES**.
- **E0** (**bluetooth**) et **ChaCha20** (**web**) sont des algorithmes **symétriques** de chiffrement par flot. Ce principe évite d'avoir à découper les messages.
- **SHA** et **MD5** sont des fonctions de **hashage** cryptographiques. Permettent surtout de s'assurer qu'un message ou fichier n'a pas été altéré.

- **TLS** et **SSL** : protocole réseau “englobant” **HTTP** pour former **HTTPS**. Permet la communication sécurisée entre un client et un serveur web. Utilise tous les précédents.
- **End to End Encryption** principe combinant chiffrement **asymétrique** et **symétrique** pour assurer des échanges. Ce terme signifie que le serveur entre les usagers Alice et Bob (qui se parlent) ne peut décoder le message. Utilisé par **Signal**, **Wire**, **Whatsapp** et **Telegram** (optionnel).

Source : https://qkzk.xyz/docs/nsi/cours_terminale/architecture/securisation/introduction/