NSI Terminale - Architecture

Sécurisation : résumé

qkzk

2020/06/28

Sécurisation des communications

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,

Chiffrement

L'objectif : assurer qu'un message ne puisse être lu que par son destinataire.

```
    message clair                    coder              message codé
Les navires arrivent à minuit   ---------------->   KZQ BPCUEZ PEEUCZBR P LUBUR

    message codé                    décoder             message clair
KZQ BPCUEZ PEEUCZBR P LUBUR     ---------------->   Les navires arrivent à minuit
```

Symétrique vs asymétrique

symétrique:

> l'émetteur et le receveur partagent une même clé qui sert au chiffrement et au déchiffrement.
>
> Cette clé **doit avoir été communiquée avant le premier message**...

asymétrique :

> l'émetteur encode avec la clé publique du réceveur.
>
> le receveur décode avec sa clé privée. Lui seul peut décoder le message.
>
> Lorsque le receveur répond, il fait le contraire.

Le chiffrement symétrique repose sur l'existence de fonctions à sens unique ou à brèche secrète (exemple : factorisation des entiers.)

Certains protocoles (comme Diffie-Helmman - 1976) permettent un échange de clé sécurisé.

Fonction à sens unique : la factorisation des entiers

- 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 $q\ldots$ Lesquels ?

Brèche secrète :

Si vous connaissez l'un des facteurs… alors c'est facile. $p = 487$.

Donc $q = n/p = 541$.

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.

1. J'écris à Robert : *j'encode avec sa clé publique.* Qu'est-ce que j'encode ?

   "J'ai fait du cassoulet"

2. *Robert décode le message à l'aide de sa clé privée.*

Robert encode sa réponse *avec MA clé publique* et me l'envoie.

3. Je décode *ma clé privée* etc.

Plusieurs défauts à cette méthode

1. Le chiffrement asymétrique est lent,
2. nous ne somme 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

1. J'écris à Robert : j'encode avec sa clé publique. Qu'est-ce que j'encode ?

   UNE CLÉ SECRÈTE

2. 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.

3. 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).

HTTPS

HTTPS (littéralement « protocole de transfert hypertextuel sécurisé ») est la combinaison du protocole HTTP et d'une couche de chiffrement, généralement TLS (sécurité de la couche transport).

Trois objectifs sont visés par ce protocole :

- **Authenticité** : un certificat assure que vous visitez bien le site voulu.
- **Confidentialité** : les échanges sont chiffrés et ne peuvent être lus par un tiers.
- **Intégrité** : HTTPS rend la technique "man in the middle" quasi impossible.

HTTPS procède en deux temps :

1. Négociation (poignée de main) : échange de clés, validées par un certificat (cryptographie **asymétrique**)
2. Communication : échange de données chiffrées (cryptographie **symétrique**)

1. Négociation : la poignée de main

La phase de négociation assure **l'authenticité** de l'interlocuteur.

0. Lorsque vous vous connectez, vous recevez un certificat transmis par le site. Ce certificat a été délivré par une autorité, une entreprise qui généralement ne fait que ça et en laquelle tout le monde a confiance.

   Ainsi, vous êtes rassuré : le site visité n'est pas celui d'un faussaire.

Chiffrement asymétrique : pour initialiser la connexion.

1. Ce certificat étant transmis, il contient donc une **clé publique** qui permet de chiffrer un message.
2. De votre côté le navigateur chiffre sa clé publique avec la clé publique du certificat et la retourne au serveur.

   Tout le monde peut intercepter ce message, mais **seul le serveur** peut le déchiffrer.
3. Il déchiffre avec sa clé privée,

   Il **calcule une clé secrète** et la chiffre avec votre clé publique.

   Il vous la renvoie.
4. Vous recevez le message, le déchiffrez avec votre clé privée (seul vous pouvez le faire).

BOOM. Client et serveur sont seuls détenteurs d'une clé secrète commune.

2. Chiffrement symétrique : durant la communication

Une fois l'authenticité avérée et qu'une clé secrète a pu être échangée, la communication réelle commence.

Toutes les données sont maintenant chiffrées de manière symétrique avec la clé secrète.

Commence alors l'échange HTTP habituel... mais tous les messages - y compris les adresses (pages internes visitées, mots clés) sont chiffrées.

Résumé des étapes d'une communication TLS

A chaque envoi de données le serveur :

- découpe les données en paquets,
- compresse les données,
- chiffre les paquets avec votre clé secrète commune,
- signe les données avec sa clé privée,
- les envoie

A chaque réception de données le client :

- déchiffre avec la clé secrète,
- vérifie la signature avec la clé publique du serveur,
- décompresse les données,
- les assemble