Sécurisation des communocations

L'objectif fondamental de la sécurité informatique est de permettre à deux personnes, appelées traditionnelement *Alice* et *Bob* de communiquer à travers un canal peu sûr de telle sorte qu'un opposant, *Oscar*, qui a accès aux informations qui circulent sur le canal de communication, ne puisse ni comprendre et/ou modifier ce qui est échangé, ni se faire passer pour Alice ou Bob.

La **confidentialité** des informations est garantie par *les algorithmes de chiffrement* qui doivent en empêcher l'accès à ceux qui n'en sont pas destinataire. Ces derniers sont composés d'une fonction de chiffrement, qui transforme le message original pour le rendre incompréhensible, et d'une fonction de déchiffrement qui permet de retrouver le message original.

Deux techniques sont disponibles : le *chiffrement symétrique* et le *chiffrement asymétrique*.

1 Le chiffrement symétrique

Les *chiffrements symétriques* utilisent la *même clef* pour chiffrer et déchiffrer un message. La protection de cette clef est cruciale pour la confidentialité des informations échangées.

Un exemple simple est le chiffrement par décalage ou *chiffrement de César*, car déjà utilisé du temps des Romains. Le chiffrement du message est réalisé en décalant de n lettres dans l'alphabet chaque lettre du message initial (en recommençant à «A» si le l'on dépasse «Z»). Ici l'entier n constitue la clef de chiffrement. Par exemple, en utilisant un décalage de 5 lettres, le message

```
L INFORMATIQUE C EST SUPER
```

devient

```
Q NSKTWRFYNVZJ H JXY XZUJW
```

Une méthode de chiffrement un peu moins naïve est le *chiffrement par XOR* (*ou exclusif*). Celle-ci repose sur l'utilisation de l'opérateur binaire *ou exclusif* noté $\oplus$.
Etant donné un message m et une clef de chiffrement k, on recopie la clef de façon à obtenir une chaîne de la même longueur que le message :

```
L'INFORMATIQUE C'EST SUPER
NSINSINSINSINSINSINSINS
```

Chaque caractère est ensuite converti en nombre (par exemple en son Unicode) :

```
76 39 73 78 70 79 82 77 65 84 73 81 85 69 32 67 39 69 83 84 32 83 85 80 69 82
78 83 73 78 83 73 78 83 73 78 83 73 78 83 73 78 83 73 78 83 73 78 83 73 78 83
```

On effectue ensuite l'opération $\oplus$ sur chaque nombre du message et de la clef. Par exemple, pour la lettre L et N, on obtient :

	1	0	0	1	1	0	0	76
$\oplus$	1	0	0	1	1	1	0	78
	0	0	0	0	0	1	0	2

Le messsage chiffré sera alors :

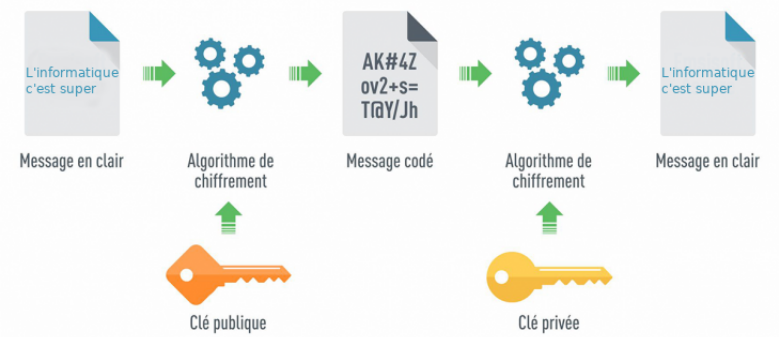
```
2 116 0 0 21 6 28 30 8 26 26 24 27 22 105 13 116 12 29 7 105 29 6 25 11 1
```

Pour déchiffrer ce message, il suffit de remarquer que $A \oplus A = 0$. Pour un message M, une clef K et le message chiffré $C = M \oplus K$, on a alors $C \oplus K = M \oplus K \oplus K = M$. On peut donc déchiffrer le message avec la même opération que pour le chiffrer.

Le standard de chiffrement symétrique depuis 2000 est l'AES (*Advanced Encryption Standard*) qui est basé sur des mathématiques plus complexes. En plus d'être sûr, il est très efficace et permet de chiffrer de longs messages en temps réel.

2 Le chiffrement asymétrique

Un *chiffrement asymétrique* utilise une clef de chiffrement différente de la clef de déchiffrement. La clef de chiffrement est connue de tous, elle est appelée *clef publique*. La clef de déchiffrement est uniquement connue de l'expéditeur, elle est appelée *clef privée*.
Si Alice veut encoyer un message à Bob, elle utilise la clef public de Bob pour chiffrer ce message. Bob pourra ensuite déchiffrer le message avec sa clef préviée.



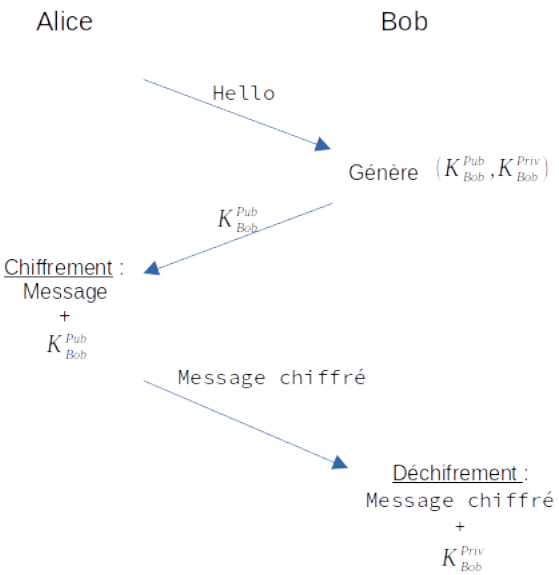
Les chiffrements asymétriques reposent sur l'existence de fonctions mathématiques dites à *sens unique*, c'est-à-dire de fonctions qui sont facilement calculable mais dont la réciproque est en pratique impossible à calculer.

L'un des chiffrements le plus utilisés est le RSA (Rivest-Shamir-Adleman) inventé en 1978, il repose sur *le problème de la factorisation* :
Soit n un entier produit de deux grand nombres premiers p et q. Si l'on connait p et q, il est très facile de calculer $n = p \times q$. Mais réciproquement, s'il on connaît n, il est très difficile de retrouver p et q.

Un autre exemple de chiffrement asymétrique est le chiffrement d'Elgamal, inventé en 1985, qui lui repose sur *le problème du logarithme discret* :
Etant connu p, g et u, il est facile de calculer le reste de la division enclidienne de g^u par p. Mais si l'on connaît p, g et g^u, il est difficile de retrouver u.

Le chiffrement asymétrique permet à Alice et Bob de s'échanger des messages sans que Oscar, qui observe uniquement les échanges (adversaire *passif*), ne puissent en déduire leur contenues.
Si Alice souhaite envoyer un message à Bob, les deux procèdent comme suit :

1. Bob génère une paire $(K^{\text{pub}}_{\text{Bob}}, K^{\text{priv}}_{\text{Bob}})$ et envoie la clef publique à Alice.
2. Alice chiffre son message m avec la clef publique de Bob, $K^{\text{pub}}_{\text{Bob}}(m)$, et l'envoie à Bob.
3. Bob utilise alors sa clef privée pour retrouver le message d'Alice, $K^{\text{priv}}_{\text{Bob}}(K^{\text{pub}}_{\text{Bob}}(m)) = m$.

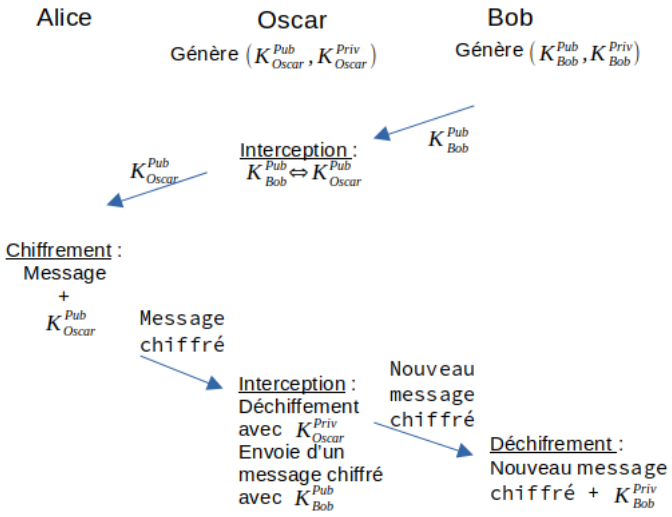


Le chiffrement asymétrique utilise des clefs de grandes tailles et nécessite un temps de calcul long et plus de ressources que lors d'un chiffrement symétrique. On utilise alors le chiffrement asymétrique pour l'envoie de message de petite taille, tel l'envoie d'une clef symétrique.

3 Authentification des participants

3.1 Attaque de «l'homme du milieu»

L'inconvénient majeur de l'échange précédent est qu'il est senseible à une attaque dite de «l'homme du milieu» (*Man in the middle*). Un adversaire *actif*, Oscar, placé entre Alice et Bob peut intercepter les communications, et ainsi se faire passer pour Bob auprès d'Alice (et réciproquement).



Oscar peut alors non seuleument obtenir tous les échanges entre Alice et Bob, mais est aussi capable de les modifier.
La faille permettant cette attaque est l'absence d'**authentification**. Oscar qui imite le comportement de Bob, n'a pas eu à prouver qu'il était Bob.

3.2 Certificats et tiers de confiance

Une solution pour se prémunir de l'attaque de «l'homme du milieu», est d'introduire une tierce personne de *confiance*. De la même manière qu'un citoyen français s'authentifie grâce à sa carte d'identité délivrée par l'Etat Français en qui l'on fait confiance, les entités sur internet sont authentifiées grâce à un *certificat* délivré par un *tiers de confiance*. Ces certificats numériques sont crées à partir des clefs publics et privées des participants.
Si Théo est le tiers de confiance d'Alice et Bob, il procède de la manière suivante :

1. Bob et Théo se rencontre. Théo vérifie l'identité de Bob et utilise ensuite sa clef **privée** $K^{\text{priv}}_{\text{Theo}}$ pour chiffrer la clef de Bob :
$$s = K^{\text{priv}}_{\text{Theo}}(K^{\text{pub}}_{\text{Bob}})$$
Le fichier s sera alors le *certificat*. On dit que Théo *signe* la clef publique de Bob.
2. Alice, qui veut envoyer un message à Bob, reçoit la clef publique de Bob $K^{\text{pub}}_{\text{Bob}}$ et le certificat s.
3. Alice récupère la clef publique de Théo $K^{\text{pub}}_{\text{Theo}}$ en qui elle a confiance et calcul :
$$K^{\text{pub}}_{\text{Theo}}(s) = K^{\text{pub}}_{\text{Theo}}(K^{\text{priv}}_{\text{Theo}}(K^{\text{pub}}_{\text{Bob}})) = K^{\text{pub}}_{\text{Bob}}$$
Si la clef envoyé par Bob correspond au résultat du calcul, Alice peut authentifier Bob comme étant son interlocuteur.
4. Alice peut alors utiliser la clef publique de Bob pour lui envoyer une clef symétrique et ainsi pouvoir initier une communication sécurisée.

4 Le protocle HTTPS

4.1 Autorités de certifications et format X.509

Un tiers de confiance sur internet est appelé une *authorité de certification* (CA) qui délivre des certificats. Les navigateurs web possèdent les clefs publiques des CA. Elles sont en nombres relativement restreint,de l'ordre de la centaine.

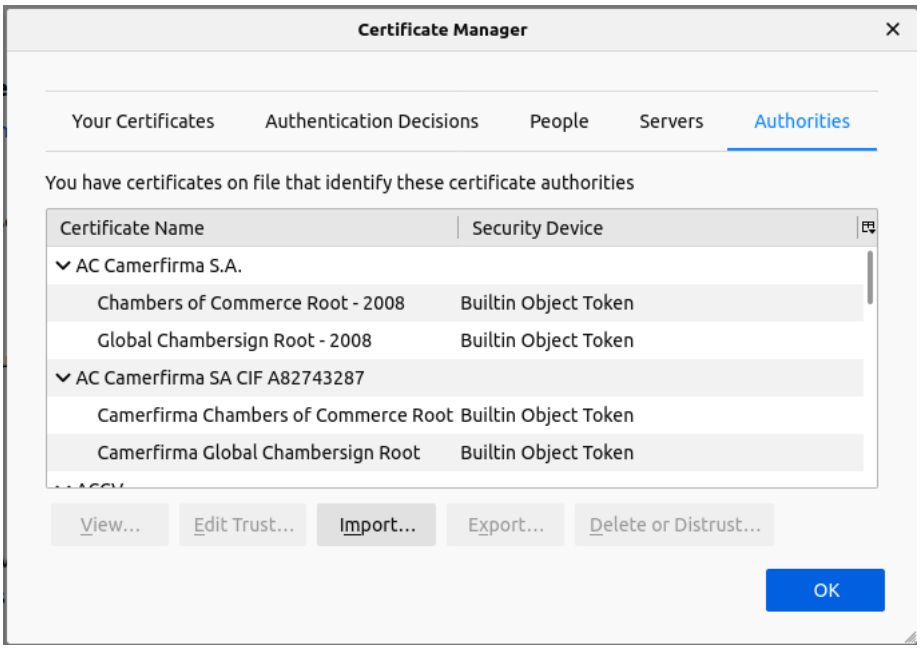


FIGURE 1 – Liste des autorités de certifications de Firefox

Le format standard de certificat est le format X.509 qui contient entre autre :

- le nom du CA emetteur,
- la période de validité,
- le nom du sujet certifié,
- la clef publique du sujet certifié.

Le tout est signé grâce à la clef privée du CA.
Sous Unix, la commande `openssl` permet d'obtenir des informations sur un certificat. Par exemple, le certificat de `https://www.cnrs.fr/` :

```
nsi@linux:$ openssl x509 -in www-cnrs-fr.pem -text

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            24:76:86:f3:98:0f:de:ac:95:6a:40:24:a5:6e:f4:90
        Signature Algorithm: sha384WithRSAEncryption
        Issuer: C = NL, O = GEANT Vereniging, CN = GEANT OV RSA CA 4
        Validity
            Not Before: Feb 10 00:00:00 2021 GMT
            Not After : Feb 10 23:59:59 2022 GMT
```

```
Subject: C = FR, postalCode = 75016, ST = \C3\8Ele-de-France, L = Paris,
street = "3, rue Michel-Ange", O = Centre national de la recherche scientifique,
OU = MOY1678, CN = www.cnrs.fr
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        RSA Public-Key: (4096 bit)
        Modulus:
            00:c4:33:04:56:d7:ef:5a:cb:0b:50:e9:c0:2c:6f: ...
        Exponent: 65537 (0x10001)
        ...
```

Le CA ayant produit ce certificat est «*GEANT Vereniging*». Le certificat est valide du 10 février 2020 au 10 février 2021. Le site certifié est `www.cnrs.fr`. La clef publique est une clef RSA de 4096 bits et est donnée par deux champs `Modulus` et `Exponent`.

4.2 Détails du protocole HTTPS

Le protocole HTTPS (*HTTP Secure*) est le protocole HTTP sécrisé par le protocole TLS (*Transport Layer Secure*) qui permet la confidentialité et l'intégrité des messages échangés.
Le protocole TLS ajoute simplement une phase permettant l'authentification du serveur et la mise en place sécurisée d'une clef de chiffrement symétrique appelé *clef de session*. La figure 2 décrit la phase de mise en place, appelée «poigné de main TLS» (*TLS handshake*).

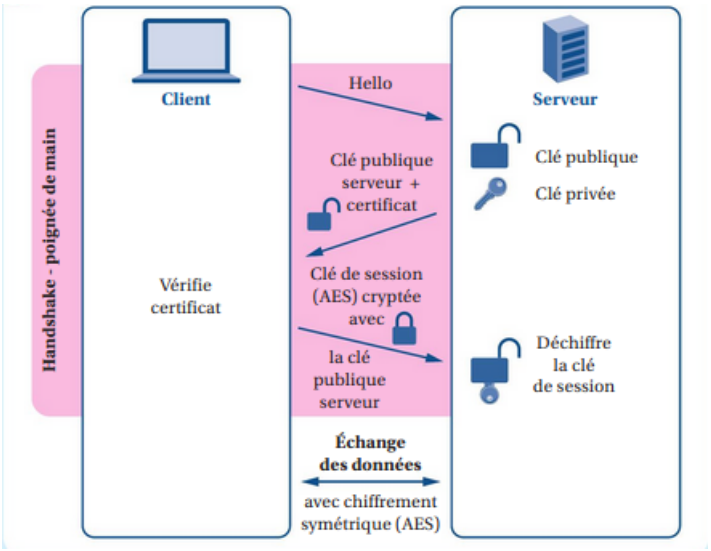


FIGURE 2 – Le Transport Layer Security