

Sécurisation des communications

- Les données transmises sur Internet sont découpées et encapsulées en paquets qui circulent à travers des réseaux de routeurs de la source à la destination. Lors de ce transfert, il est possible pour chaque routeur d'inspecter le contenu des paquets. Cette situation n'est clairement pas idéale, en particulier lors du transit d'informations sensibles (informations personnelles, données de santé, numéros de carte bancaire,...), ou de conversations privées (expression d'une opinion,...).
- La démocratisation d'Internet, la diversification des usages et l'explosion de la quantité de données transmises ont révélé la nécessité de prendre des précautions et de sécuriser les échanges en particulier autour de trois questions :
 - Comment chiffrer le contenu des communications afin qu'elles ne soient lisibles que par la source et la destination ?
 - Comment garantir que l'identité du serveur auquel on se connecte est bien celle à laquelle on pense se connecter ?
 - Comment garantir les deux points précédents en utilisant l'infrastructure d'Internet existante ?



- Depuis l'Antiquité, les êtres humains ont cherché à sécuriser leurs communications, en particulier dans le domaine militaire. Les premières méthodes de chiffrement se sont améliorées, avec la fabrication de différentes machines de chiffrement, pour obtenir un rôle majeur lors de la Première Guerre mondiale et de la Seconde Guerre mondiale (ci-contre un exemplaire de machine *enigma* utilisée par les Allemands pour chiffrer leurs communications).
- La *cryptologie*, étymologiquement la « science du secret », est un art ancien et une science nouvelle. Elle englobe la *cryptographie* (l'écriture secrète) et la *cryptanalyse* (l'analyse de cette dernière).

1. Un peu de vocabulaire.

- **Coder** : Représenter de l'information par un ensemble de signes prédéfinis.
- **Décoder** : Interpréter un ensemble de signes pour en extraire l'information qu'il représente.

Coder et décoder s'utilisent lorsqu'il n'y pas de secret (par exemple le codage des entiers). On utilise aussi coder comme synonyme de programmer.

- **Chiffrer** : Rendre une suite de symboles incompréhensibles à l'aide 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 message initial.
 - On utilise le terme *décrypter* lorsque l'on détermine le message initial sans utiliser la clé.

2. Chiffrement symétrique

Le terme *symétrique* vient du fait que la même clé est utilisée pour chiffrer et déchiffrer le message.

Un premier exemple

- Le chiffrement par décalage (appelé aussi codage de César, en référence à Jules César qui utilisait cette technique pour ses correspondances militaires) consiste à choisir un entier n et à décaler chaque lettre du message initial de n lettres dans l'alphabet (en recommençant à "A" si le décalage fait dépasser "Z"). C'est l'entier n qui constitue la clé de chiffrement.
- La méthode de déchiffrement revient à décaler les lettres du message chiffré de n positions dans l'autre sens.

Exemple 1 :

En utilisant un décalage de 3 lettres, le message "ils sont fous ces romains" devient "lov vrqw irxv fhv urpdlqv".

XOR

Une méthode de chiffrement un peu moins naïve consiste à utiliser l'opérateur binaire "xor" (ou exclusif) noté \oplus et une clé de chiffrement selon les principes énoncés dans l'exemple suivant :

Exemple 2 : chiffrement

- Considérons le message "L'INFORMATIQUE C'EST SUPER" et la clé "NSI". On recopie la clé plusieurs fois de façon à obtenir une chaîne de même longueur que le message initial :

```
L'INFORMATIQUE C'EST SUPER
NSINSINSINSINSINSINSINSINS
```

- Chaque caractère du message et de la clé étendue est ensuite converti en nombre (par exemple par son code Unicode, ou ce qui revient au même ici à son code ASCII)

```
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 entre chaque nombre du message et de la clé étendue. Cet opérateur entre deux bits renvoie 0 si les deux bits sont égaux et 1 s'ils sont différents : $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$
- Voici ce que cela donne pour la lettre "T" (84) et la lettre "N" (78) :

```
    01010100    84
⊕  01001110    78
-----
    00011010    26
```

- Ainsi la lettre "T" sera chiffrée 26. En effectuant cette opération pour tous les caractères du message initial, le message chiffré sera donc :
 - 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
- On remarque que grâce à la clé , une même lettre n'est pas nécessairement chiffrée par le même nombre et que deux nombres égaux ne chiffrent pas nécessairement la même lettre.

Exemple 3 : déchiffrement

- L'opérateur xor possède une propriété intéressante : Il est réversible. Cela signifie que si $A \oplus B = C$, alors $A \oplus C = B$ et $B \oplus C = A$. on peut aisément le vérifier sur l'exemple précédent : $84 \oplus 78 = 26$, $84 \oplus 26 = 78$, $26 \oplus 78 = 84$.

```
    01010100    84
⊕  01001110    78
-----
    00011010    26
```

```
    01010100    84
⊕  00011010    26
-----
    01001110    78
```

```
    00011010    26
⊕  01001110    78
-----
    01010100    84
```

- On peut donc déchiffrer le message de l'exemple réexécutant l'opérateur \oplus avec la clé "NSI" au message chiffré :

```
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
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
```

Résultat :

```
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
```

L'opérateur \oplus est une des opérations de base implémentées dans les circuits des processeurs, plus précisément dans l'UAL (Unité arithmétique et Logique) qui effectue les calculs sur les opérations de base. Ces caractéristiques font qu'il est couramment utilisé dans les algorithmes de chiffrement modernes (de façon plus sûre et plus complexe que dans l'exemple...)

Algorithmes

Voici deux algorithmes de chiffrement symétrique parmi les plus utilisés aujourd'hui :

- AES(Advanced Encryption Standard)
- ChaCha20

Bien que très complexes, ils reposent sur des principes similaires au chiffrement xor décrit précédemment.

- Une clé initiale est étendue(mais pas aussi simplement qu'en la répétant)
- La clé et le message sont mélangés(en utilisant entre autres des opérations \oplus), de façon réversible.

En plus d'être sûrs (il est quasiment impossible de décrypter le message sans posséder la clé), ils sont très efficaces et permettent de chiffrer de longs messages.

Exemples d'utilisation d'AES

- Nombre de logiciels de gestion de mots de passe utilisent le cryptage AES. Cela permet de sécuriser les données contenues dans les coffres-forts numériques.
- Les VPN(Virtual Private Network) utilisent également des clés de chiffrement. En effet, leur fonctionnement nécessite le cryptage des données circulant entre les ordinateurs privés et leurs serveurs. Ainsi sont garantis l'anonymat en ligne et la protection contre les attaques en force brutes
- Avec la transformation numérique, la dématérialisation et les nouveaux usages qui vont avec (télétravail par exemple), de plus en plus de professionnels utilisent ce type de logiciels pour échanger des fichiers. Et certains contiennent parfois des données sensibles, tels des dossiers médicaux. C'est pourquoi ces solutions recourent souvent à la norme AES.

3. Chiffrement asymétrique

- On peut considérer que le chiffrement symétrique est sûr, pour peu que la clé soit suffisamment longue et judicieusement choisie. Il est alors difficile de décrypter les informations par attaque de force brute (c'est à dire en testant toutes les combinaisons possibles jusqu'à trouver la clé). Par exemple , si l'on disposait de 3 milliards de machines, et que chacune puisse tester un milliard de clé par secondes, cela prendrait un peu plus de deux milliards d'années pour récupérer une clé de chiffrement AES de 128 bits...
- Il y a néanmoins un défaut important. En effet si deux personnes(deux machines) souhaitent communiquer de façon sécurisée, elles doivent d'abord se mettre d'accord sur la clé à utiliser(et sur l'algorithme de chiffrement). Or elles sont justement dans la situation où elles ne peuvent pas encore communiquer de façon sûre. Il faut donc en plus un moyen sûr de communiquer la clé.
- Pour résoudre ce problème, diverses techniques ont été développées dès les années 1970 d'abord par les services secrets britanniques et américains. Ces techniques reposent sur un *chiffrement asymétrique* appelée aussi *chiffrement à clé publique*.
- Il s'agira dans cette partie de décrire les principes de ces techniques, sans entrer dans les détails mathématiques qui dépassent largement le programme de terminale.

Méthode de Diffie-Hellman

Diffie-Hellman est un protocole qui permet à des participants de se mettre d'accord sur une clé de chiffrement symétrique en utilisant un canal de communication non sûr.

Ce protocole proposé en 1976, doit son nom aux mathématiciens Whitfield Diffie(1944-) et Martin Hellman (1945-).

Il repose sur les concepts de *clé publique* et *clé privée* ainsi que sur des fonctions mathématiques qui ne seront pas décrites en détail ici.



Exemple 4 : Analogie du cadenas

Imaginons que Bob souhaite envoyer un message à Alice sans qu' Oscar, qui possède le réseau de communications, puisse le lire :

- Alice fabrique un cadenas(sa clé publique) et une clé pour l'ouvrir (sa clé privée).
- Alice envoie le cadenas ouvert(la clé publique) à Bob. Tout le monde peut potentiellement le voir, y compris Oscar.
- Bob utilise la clé publique pour chiffrer son message : il le glisse dans une boîte verrouillée à l'aide du cadenas. Mais connaître la clé publique ne suffira pas à déchiffrer le message sans connaître la clé privée. Ainsi, quand Bob envoie le tout à ALice, Oscar voit la boîte verrouillée par le cadenas mais ne peut pas l'ouvrir.
- Alice ouvre le cadenas avec sa clé privée et consulte le message.

Exemple 5 : Descriptif mathématique

- Alice et Bob souhaitent se communiquer une clé secrète dans un réseau de communication possédé par Oscar. Ils vont pour cela utiliser des clés publiques que tout le monde pourra voir.
- La clef publique est constituée de deux nombres : un nombre premier P et un nombre G plus petit que P . Tous les calculs se feront modulo P (c'est à dire en considérant le reste de la division euclidienne par P).
- On utilise une fonction f qui à tout entier n associe le nombre $f(n) = G^n \% P$

Voici un exemple simple lorsque $P = 23$ et $G = 5$:

1. Alice choisit sa clé privée a , par exemple $a = 6$. Elle calcule sa clé publique $A = f(a)$, par le calcul $A = 5^6 \% 23 = 8$ et l'envoi à Bob. Oscar peut la voir.
2. Bob fait de même. Il choisit sa clé privée b , par exemple $b = 15$. Il calcule sa clé publique $B = f(b) = 5^{15} \% 23 = 19$ et l'envoi à Alice. Oscar peut la voir.
3. Alice, calcule une clé de chiffrement en effectuant $B^a \% P$, c'est à dire $19^6 \% 23 = 2$
4. Bob, par un raisonnement analogue calcule $A^b \% P$, ce qui donne $8^{15} \% 23 = 2$
5. Alice et Bob ont obtenu la même clé sans avoir communiqué leur clé privée à quiconque.

Remarques :

- La clé commune obtenue par Alice et Bob est en fait $f(ab) = (G^a)^b \% P = (G^b)^a \% P = G^{ab} \% P$
- Bien qu' Oscar possède A et B il lui est difficile d'effectuer les opérations inverses pour trouver $f(ab)$ (cela dépasse largement le cours de terminale) sauf à utiliser la *force brute* , c'est à dire tester toutes les possibilités...Ce qui est faisable si les nombres P et G sont petits , comme ici.
- En pratique , on utilise généralement des nombres premiers P de taille 2048 bits (de l'ordre de 600 chiffres en écriture décimale !), ce qui rend impossible les attaques par force brute en l'état actuel des technologies.
- Pour en savoir plus :
 - Descriptif du protocole sur Wikipédia : https://fr.wikipedia.org/wiki/%C3%89change_de_cl%C3%A9s_Diffie-Hellman (https://fr.wikipedia.org/wiki/%C3%89change_de_cl%C3%A9s_Diffie-Hellman)
 - Des exemples de nombres premiers utilisables en production : <https://tools.ietf.org/html/rfc3526#section-3> (<https://tools.ietf.org/html/rfc3526#section-3>)

RSA (Rivest, Shamir, Adleman)

- Même si le protocole Diffie-Hellman permet de résoudre le problème de l'échange de clés, il reste un point faible, celle de l'authenticité des participants : Quand Alice envoie sa clé publique, est-elle sûre de l'envoyer à Bob ? N'est-ce pas Oscar qui se fait passer par Bob ? Est-elle sûre que le message qu'elle a reçu est bien de Bob ?
- La méthode de chiffrement asymétrique RSA permet de résoudre, en partie, ce problème. Elle doit son nom à ses trois concepteurs cryptologues ci-contre : Ron Rivest(1947 -), Adi Shamir (1952 -), Len Adleman(1945 -).
- Ce système repose sur des théorèmes mathématiques complexes qui ne seront pas abordés ici.
- RSA consiste en la mise en place d'une paire de clés par participant : une clé publique et une clé privée.



Exemple 6 :

On note :

- K_{Alice}^{Pub} : La clé publique d'Alice
- K_{Alice}^{Priv} : La clé privée d'Alice
- La notation $K_{Alice}^{Pub}(m)$ signifie "chiffrer un message m avec la clé publique d'Alice".

La manière exacte de générer ces clés fait appel aux nombres premiers et à l'arithmétique. Ce qu'il faut retenir, c'est que ces deux clés sont liées :

- Alice peut chiffrer un message m avec sa clé privée et le déchiffrer avec sa clé publique. Mais elle peut aussi le chiffrer avec sa clé publique et le déchiffrer avec sa clé privée.
- Ainsi on a $K_{Alice}^{Pub}(K_{Alice}^{Priv}(m)) = K_{Alice}^{Priv}(K_{Alice}^{Pub}(m)) = m$

Les propriétés mathématiques en jeu font qu'il est impossible de:

- Deviner K_{Alice}^{Priv} en connaissant K_{Alice}^{Pub}
- Deviner m en connaissant $K_{Alice}^{Pub}(m)$ ou $K_{Alice}^{Priv}(m)$

A l'aide de ces propriétés, voici comment Bob peut envoyer un message secret à Alice :

1. Alice envoie à Bob (et à tout le monde) sa clé publique K_{Alice}^{Pub}
2. Bob chiffre son message m avec la clé publique d'Alice, il obtient $K_{Alice}^{Pub}(m)$ et l'envoie à Alice.
3. Alice déchiffre le message de Bob avec sa clé privée, elle obtient $K_{Alice}^{Priv}(K_{Alice}^{Pub}(m)) = m$

Si Bob dispose d'une paire de clés privée et publiques, Alice pourra lui envoyer des messages.

Remarques :

- RSA permet de chiffrer des messages sans s'être mis d'accord sur une clé de chiffrement symétrique.
- L'inconvénient de RSA, c'est qu'il nécessite des calculs coûteux, de par la taille des nombres en jeu.
- En pratique, il est utilisé comme le protocole Diffie-Hellman. Un des participants choisit une clé et un algorithme de chiffrement symétrique, puis il communique cette clé en la chiffrant avec le protocole RSA. Une fois la clé reçue, un algorithme de chiffrement symétrique efficace peut être utilisé.
- Le protocole RSA est impossible à casser en un temps raisonnable.
- Un autre avantage de RSA, c'est qu'il est utilisé comme système d'authentification.
- En savoir plus :
 - Fonctionnement détaillé : https://fr.wikipedia.org/wiki/Chiffrement_RSA (https://fr.wikipedia.org/wiki/Chiffrement_RSA)
 - Vidéo explicative des concepts mathématiques sous-jacents : <https://www.youtube.com/watch?v=Y2bsLRdVBP8> (<https://www.youtube.com/watch?v=Y2bsLRdVBP8>)

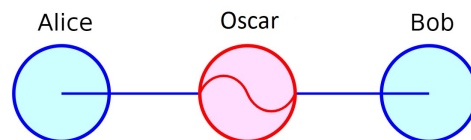
4. Authentification des participants

- Jusqu'à présent, Alice et Bob disposent d'outils qui leurs permettent de communiquer de façon sûre en empêchant quiconque qui réussirait à obtenir les messages chiffrés, de les déchiffrer.
- Mais il reste un point important : Lorsque qu'Alice envoie sa clé publique, est-elle sûre qu'elle l'envoie à Bob ? Lorsqu'elle reçoit la clé de Bob, est-elle sûre que c'est Bob qui l'a envoyée ? Et si c'était Oscar ?

Attaque de l'homme du milieu

Exemple 7 :

- Oscar qui possède le réseau, se fait passer pour Bob et demande à Alice de communiquer.
- Alice envoie sa clé publique à Oscar, pensant que c'est Bob.
- Oscar envoie sa clé publique à Alice.
- Oscar et Alice créent un canal sûr de communication entre eux : Oscar peut déchiffrer les messages d'Alice.
- Oscar envoie sa clé publique à Bob en se faisant passer pour Alice.
- Bob envoie sa clé publique à Oscar en pensant que c'est Alice.
- Oscar et Bob créent un canal sûr de communication entre eux : Oscar peut déchiffrer les messages de Bob.
- Oscar peut déchiffrer les messages d'Alice et Bob en se faisant passer l'un ou l'autre.



Ainsi, le chiffrement, aussi sûr soit-il, n'est ici d'aucune utilité.

L'attaque de l'homme du milieu (*Man In The Middle*, ou *MITM*) est une manoeuvre classique que l'on rencontre souvent lorsque l'on reçoit par exemple des messages nous incitant à cliquer sur un lien vers notre banque pour diverses raisons. Ce type de message contient la plus part du temps de faux liens qui nous redirigent vers des sites usurpant l'identité du serveur auquel on pense se connecter afin de récolter de précieuses informations (mots de passe, clés de chiffrement...)

Certificats et tiers de confiance

En plus de chiffrer les communications, il faut s'assurer de l'identité de son correspondant. Sur Internet, il existe des *certificats* permettent à une entité de prouver son identité. Ces certificats sont délivrés par une autre entité, en qui tous les participants peuvent avoir confiance, que l'on appelle un *tiers de confiance*. Ces certificats numériques sont générés à l'aide des clés RSA privées et publiques des participants.



Exemple 8 : Reprenons la situation précédente.

- Bob va voir Neo qui vérifie l'identité de Bob. Une fois cela fait, Neo fournit un *certificat* avec sa signature en chiffrant la clé publique de Bob avec sa clé privée. On pourrait écrire que le certificat s fourni par Neo est $s = K_{Neo}^{Priv}(K_{Bob}^{Pub})$
- Lorsqu'Alice demande à communiquer avec Bob, celui-ci fournit sa clé publique K_{Bob}^{Pub} et son certificat s .
- Alice récupère auprès de Neo en qui elle a confiance sa clé publique K_{Neo}^{Pub} et effectue le calcul $K_{Neo}^{Pub}(s) = K_{Neo}^{Pub}(K_{Neo}^{Priv}(K_{Bob}^{Pub})) = K_{Bob}^{Pub}$
- Alice compare le résultat avec la clé publique que Bob lui a fourni et vérifie ainsi que cette clé est bien celle de Bob.
- Une fois sûre, elle peut initier une communication sécurisée en utilisant RSA ou Diffie-Hellman pour se mettre d'accord sur une clé.

5. Protocole HTTPS

Le protocole HTTPS (HyperText Transfert Protocol Secure) est le protocole qui sécurise les échanges sur le web. Il repose non seulement sur les différents principes décrits précédemment, mais il satisfait aussi d'autres conditions :

- Il est compatible avec HTTP, il le rend plus sûr.
- Il assure une compatibilité avec de futures modifications. En effet, il est pour l'instant impossible pour un attaquant de casser les clés de chiffrement dans un temps de calcul raisonnable. Mais il faut pouvoir se prémunir des attaques futures, réalisées avec du matériel utilisant des technologies nouvelles et plus puissantes. Il faut pouvoir augmenter la difficulté des problèmes à résoudre (augmenter la taille des clés, changer un algorithme de chiffrement,...) sans remettre en cause tout le protocole.
- Il est performant. Le chiffrement asymétrique, plus sûr mais aussi plus coûteux en temps de calcul, est utilisé au strict minimum, lors de l'authentification et de l'échange de clé. En complément, le chiffrement symétrique permet ensuite d'échanger de façon sûre de grandes quantités de données, en temps réel.

Autorités de certifications

- Une autorité de certification (CA pour *Certificate Authority*) est une entité habilitée à délivrer des certificats numériques. Cela peut-être une entreprise spécialisée, une association à but non lucratif, un état.
- Elles sont peu nombreuses, vérifient des critères stricts et sont contrôlées régulièrement.
- Les systèmes d'exploitation et les navigateurs Internet possèdent les clés publiques des autorités de certifications sous forme de fichiers.
- C'est avec ces clés publiques que l'on peut par exemple vérifier la validité du certificat d'un site selon les principes décrits précédemment.

Exemple 9 :

Dans la plupart des navigateurs, il est aisé de consulter le certificat d'un site en cliquant sur l'icône du cadenas située en général à gauche de l'URL. Ces fichiers sont au format X.509. Ci-dessous des extraits du certificat délivré à Twitter, (captures d'écran), récupérés depuis le navigateur Firefox :

Nom du sujet		Informations sur la clé publique	
Pays	US	Algorithme	RSA
État / Province	California	Taille de la clé	2048
Localité	San Francisco	Exposant	65537
Organisation	Twitter, Inc.	Module	91:7F:2F:E2:71:54:59:47:27:B8:07:69:3E:26:9E:AB:4B:18:61:A9:D7:EB:82:79:63:C2:...
Nom courant	twitter.com		
Nom de l'émetteur		Divers	
Pays	US	Numéro de série	0F:1E:51:1A:5F:A8:41:65:32:47:E1:A4:F1:E9:D4:76
Organisation	DigiCert Inc	Algorithme de signature	SHA-256 with RSA Encryption
Nom courant	DigiCert TLS RSA SHA256 2020 CA1	Version	3
		Télécharger	PEM (cert) PEM (chain)
Validité		Empreintes numériques	
Pas avant	Wed, 24 Feb 2021 00:00:00 GMT	SHA-256	FE:7C:62:83:D9:F0:14:87:F3:63:7A:9D:3F:51:DB:E8:9C:12:AE:6A:8B:A0:72:BB:39:3...
Pas après	Tue, 22 Feb 2022 23:59:59 GMT	SHA-1	C2:38:5F:66:19:47:6E:52:41:4D:4D:F3:23:C1:5A:8A:77:FB:D3:19
Noms alternatifs du sujet		Contraintes de base	
Nom DNS	twitter.com	Autorité de certification	Non
Nom DNS	www.twitter.com		

On y trouve, entre autres informations :

- Le ou les noms de site certifiés : twitter.com , www.twitter.com
- L'identité de l'autorité qui signe le certificat : "DigiCertInc"
- La date de validité du certificat : jusqu'au 22 février 2022
- La clé publique de l'entité certifiée, ici Twitter, celle qui sera utilisée par le navigateur pour chiffrer le début de la communication : Sur 2048 bits, elle est composée de deux valeurs (module et exposant)
- L'algorithme de génération de la clé publique : RSA
- L'algorithme utilisé pour la signature du certificat : "SHA-256 with RSA Encryption" (une variation de RSA)
- La signature du certificat : Ce sont les valeurs du champ "Empreintes numériques". Cela correspond à l'étape décrite à l'exemple 8, le chiffrement du certificat avec la clé privée de l'autorité.

Ainsi, pour vérifier que le site est bien celui de Twitter, le navigateur utilise la clé publique de l'autorité de certification pour vérifier la signature. Les navigateurs possèdent dès leur installation les clés publiques des autorités de certifications.

Détails du protocole HTTPS

Alice et Bob nous ont aidé à comprendre la nécessité de sécuriser une communication. Maintenant, allons vraiment sur le web:

Le protocole HTTPS permet d'établir des communications sécurisées entre un client et un serveur web. Il utilise le protocole HTTP plus ancien auquel une couche de cryptographie est ajoutée qui va permettre l'authentification du serveur, c'est le protocole TLS (*Transport Layer Security*, il succède à un ancien protocole nommée SSL).

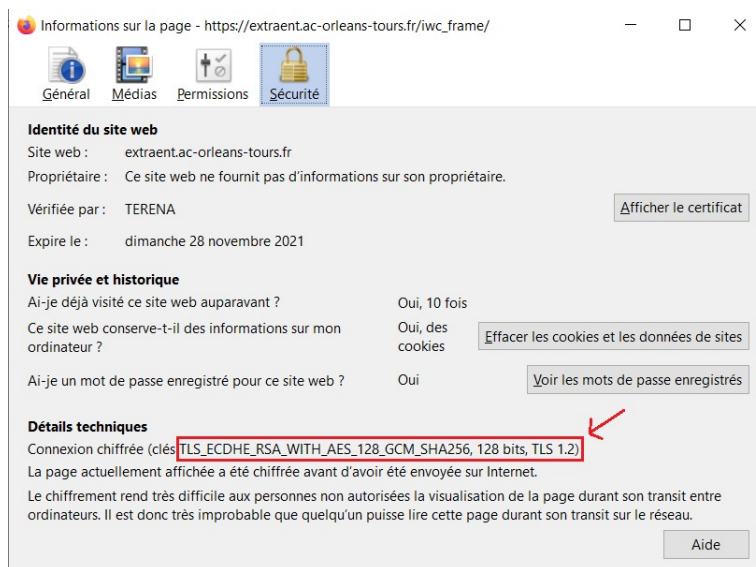
La phase d'authentification avec TLS est communément appelée poignée de main (*handshake* en anglais). Voici les étapes de cette poignée de main :

1. Le navigateur du client envoie un message initial au serveur (un "Hello") contenant entre autre la version de TLS utilisée, les algorithmes de chiffrements qu'il supporte,...
2. Le serveur envoie sa réponse qui contient entre autres le certificat X.509 contenant sa clé publique, signée par une autorité de certification.
3. Le navigateur vérifie le certificat à l'aide de la clé publique de l'autorité de certification.
4. Le client et le serveur conviennent d'une *clé de session* (par exemple avec RSA ou avec Diffie-Hellman) qui va leur permettre ensuite d'initier un chiffrement symétrique.
5. Le serveur est authentifié par le client et les deux ont convenu d'une clé de session, ils peuvent désormais échanger des messages chiffrés tout en utilisant le protocole HTTP.

Remarques :

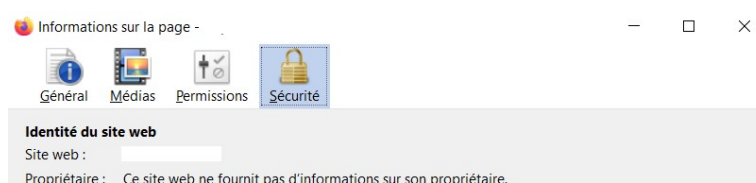
- L'authentification empêche les attaques de type homme du milieu.
- Le chiffrement symétrique des messages suffit ensuite à empêcher toute machine intermédiaire entre le serveur et le client (routeur, autre machine du réseau, hacker,...) de décrypter les messages, même s'ils sont interceptés.
- Certaines de ces informations sont accessibles partir de l'icône du cadenas située en général à gauche du champ d'URL dans les navigateurs.
- La plupart des navigateurs récents mettent en gardent et refusent souvent l'accès à une connexion non sécurisée.
- Même lorsque tout semble authentique, il convient de rester prudent lors du transfert d'informations sensibles (mots de passe, numéro de carte bancaire, ...)

Exemple 10 : Une connexion sécurisée



- La chaîne `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bits, TLS 1.2` nous indique que :
 - Le protocole TLS version 1.2 est utilisé.
 - Le protocole d'échange de clé se fait avec une variante de Diffie-Hellman (ECDHE : Elliptic Curve Diffie Hellman Ephemeral)
 - L'authentification se fait avec RSA
 - Le chiffrement symétrique se fait avec AES, sur une clé de 128 bits.

Exemple 11 : Une connexion non chiffrée



Vérifiée par : Non spécifié

Vie privée et historique

Ai-je déjà visité ce site web auparavant ? Non

Ce site web conserve-t-il des informations sur mon ordinateur ? Non [Effacer les cookies et les données de sites](#)

Ai-je un mot de passe enregistré pour ce site web ? Non [Voir les mots de passe enregistrés](#)

Détails techniques

Connexion non chiffrée

Le site web ne prend pas en charge le chiffrement pour la page actuellement affichée.

Les informations envoyées sur Internet sans chiffrement peuvent être vues par d'autres personnes durant leur transit.

[Aide](#)

Dans cet exemple, les données transmises vers le site ne sont pas chiffrées. Il est alors aisé de récupérer ces données en clair, par exemple avec un logiciel de capture de trames (type Wireshark ou autre). On remarque aussi que cette connexion n'est pas authentifiée, autant de raisons de se méfier. La plupart des navigateurs signalent de façon encore plus explicite qu'accéder à ce type de site est potentiellement

6. Pour résumer...

- La sécurisation des communications sur Internet repose sur la cryptographie.
- Les algorithmes de chiffrement symétrique utilisent la même clé pour chiffrer et déchiffrer les messages. Certains, comme AES ou ChaCha20 sont sûrs et efficaces. Ils supposent néanmoins que les participants puissent échanger la clé symétrique de façon sûre.
- Cela peut être fait grâce à la cryptographie asymétrique avec un protocole d'échange de clés comme Diffie-Hellman ou une paire de clés publique et privée fournies par le protocole RSA.
- RSA permet également d'authentifier un participant avec l'aide de tiers de confiance que sont les autorités de certifications.
- Le protocole HTTPS ajoute au protocole HTTP une phase d'authentification et d'échange de clé avec le protocole TLS.
- Une fois le serveur authentifié par le client, ce dernier utilise RSA ou Diffie-Hellman pour convenir d'une clé de chiffrement symétrique. Les deux parties peuvent alors commencer une communication chiffrée.

Une vidéo de 10 mn qui reprend l'essentiel de ce cours :

<https://www.youtube.com/watch?v=7W7WPMX7arl> (<https://www.youtube.com/watch?v=7W7WPMX7arl>)

