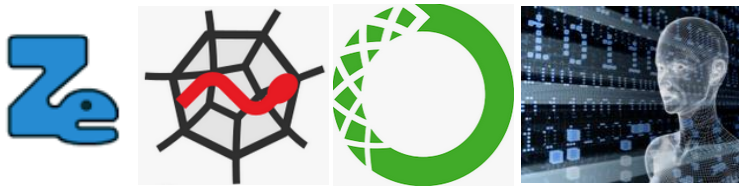


Sécurité des communications

Vocabulaire

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.

Cryptanalyse : Est un ensemble de techniques utilisées pour décrypter un texte, c'est-à-dire retrouver le texte en clair à partir du texte chiffré sans posséder la clé de déchiffrement.

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 » est aussi utilisé de façon informelle comme synonyme de « programmer ». L'anglicisme « crypter » est à proscrire ; on utilisera « chiffrer ».

Introduction

https://www.youtube.com/watch?v=8BM9LPDjOw0

Les codes secrets - Écrit et réalisé par David Louapre © Science étonnante

Cryptographie symétrique

Une première technique lorsque l'on souhaite chiffrer un message est d'utiliser une méthode de chiffrement symétrique. Formellement, une telle méthode est donnée par deux fonctions :

- c(m, k) est la fonction de chiffrement. Elle prend en arguments un message en clair m (une chaîne de caractères par exemple) et une clé de chiffrement k (qui peut être une chaîne de caractères, un nombre, etc.). Elle produit en sortie une chaîne de caractères chiffrée m'.
- d(m', k) est la fonction de déchiffrement. Elle prend en arguments un message chiffré m' et une clé de déchiffrement k et renvoie le message en clair m.

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

XOR (ou exclusif)

Cette méthode de chiffrement symétrique repose sur l'utilisation de l'opérateur binaire ou exclusif noté ⊕. Etant donné un message m (par exemple « L'INFORMATIQUE C'EST SUPER ») et une clé de chiffrement k (par exemple « NSI »), on recopie plusieurs fois la clé de façon à obtenir une chaîne de même longueur que le message :

L	`	I	N	F	O	R	M	A	T	I	Q	U	E		C	`	E	S	T		S	U	P	E	R
N	S	I	N	S	I	N	S	I	N	S	I	N	S	I	N	S	I	N	S	I	N	S	I	N	S	I	N	S

Chaque caractère du message et de la clé augmentée est ensuite converti en nombre (par exemple en son code 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 ⊕ entre chaque nombre du message et de la clé. Nous la détaillons ci-dessous entre les nombres 84 (la lettre « T ») et 78 (la lettre « N ») :

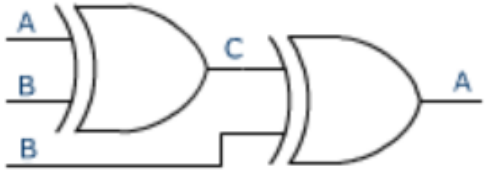
Table de vérité de XOR		
A	**B**	**R = A ⊕ B**
0	0	0
0	1	1
1	0	1
1	1	0

	0	1	0	1	0	1	0	0	84
⊕	0	1	0	0	1	1	1	0	78
	0	0	0	1	1	0	1	0	26

On rappelle que l'opération ⊕ entre deux bits renvoie 0 si les deux bits sont égaux et 1 s'ils sont différents. 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

Il existe une multitude de propriété concernant l'opérateur ⊕. Cependant la propriété qui nous intéresse à ce stade est qu'il est réversible :
si A ⊕ B = C, alors A ⊕ C = B et B ⊕ C = A.



Ainsi, on peut facilement contrôler que 84 ⊕ 78 = 26, mais aussi que 84 ⊕ 26 = 78 et 78 ⊕ 26 = 84. Par conséquent, le message chiffré ci-dessus peut être déchiffré en réexécutant l'opération ⊕ nombre à nombre avec les entiers formant la clé étendue « NSINSINSINSINSINSINSINSINS ».

Attention cependant : si la méthode de chiffrement est connue, alors une clé trop courte (comme celle choisie ici) peut compromettre la sécurité de la communication.

Parmi les algorithmes de chiffrement symétriques les plus utilisés, on peut citer AES (Advanced Encryption Standard) et ChaCha20.

Quelques propriétés de l'opération ou exclusif :

- ✓ Commutatif A ⊕ B = B ⊕ A
- ✓ Associatif (A ⊕ B) ⊕ C = A ⊕ (B ⊕ C)

  <u>Conséquence l'ordre des calculs n'intervient pas.</u>
- ✓ C'est un opérateur d'imparité

  A ⊕ B ⊕ C ⊕ D ⊕ E = ? Et bien il suffit de décompter le nombre de variables en entrée à 1. Si ce nombre est impair alors le résultat global est égal à 1 sinon à zéro.

  Conséquence pour calculer par exemple (A ⊕ B) ⊕ C

  on applique l'associativité donc (A ⊕ B) ⊕ C = A ⊕ B ⊕ C

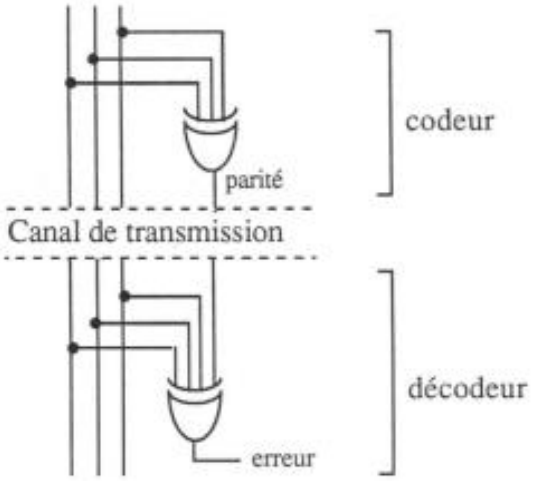
  et pour calculer le résultat il suffit de décompter le nombre d'entrée à 1 et donc :

A	B	C	A ⊕ B ⊕ C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Application le bit de parité[1] :

Prenons un code quelconque, et adjoignons à chaque mot un bit supplémentaire, dit **bit de parité**, formé de la somme OU EXCLUSIF des autres bits du mot.

Chaque mot du nouveau code ainsi obtenu contiendra un nombre pair de "1". Pour détecter une erreur (ou plus exactement un nombre impair d'erreurs), il suffira de vérifier la parité à la réception, c'est-à-dire d'effectuer la somme OU EXCLUSIF de tous les bits du mot reçu (y compris le bit de parité). Bien entendu, telle quelle, cette méthode est incapable de détecter un nombre pair d'erreurs.



Cryptographie asymétrique

Malgré les nombreux avantages du chiffrement symétrique, ce dernier possède un défaut important. En effet, si deux personnes veulent établir un canal de communication sûr en utilisant une méthode de chiffrement symétrique, les deux personnes doivent d'abord se mettre d'accord sur la clé à utiliser. Or, les deux participants sont justement dans la situation où ils ne peuvent pas encore communiquer de façon sûre.

Le système RSA

Le système RSA est un système de chiffrement asymétrique basé sur des paires de clés publiques et privées. Ce système repose sur des théorèmes mathématiques complexes, dont le détail va au-delà du programme de terminale.

Le système RSA consiste en la mise en place d'une paire de clé publique et privée pour chaque participant. Nous notons K^{pub}_k la clé publique de Krabs et K^{priv}_k sa clé privée. La notation $K^{-}_k(m)$ signifie, « chiffrer m avec la clé K (publique ou privée) de Krabs ». Etant donné un message m, on a :

$$K^{pub}_k(K^{priv}_k(m)) = K^{priv}_k(K^{pub}_k(m)) = m$$

Ce qui signifie que si on chiffre le message m d'abord avec la clé privée puis qu'on chiffre le résultat avec la clé publique, ou dans l'autre ordre, on obtient

[1] Technique générale en électronique numérique, la rédaction présentée est Pierre Csillag, Introduction aux codes correcteurs, Ellipses

le message initial. En d'autres termes, l'une des deux clés permet de défaire le chiffrement effectué par l'autre clé. Les propriétés mathématiques mises en jeux font aussi qu'il est :

- Impossible, en connaissant K^{pub}_k, de deviner K^{priv}_k ;
- Impossible, en connaissant uniquement $K^{pub}_k(m)$ ou $K^{priv}_k(m)$, de deviner m.

Une fois ces propriétés établies, si Bob souhaite envoyer un message secret à Krabs, les deux procèdent comme suit :

1. Krabs diffuse sa clé publique K^{pub}_k. Par exemple, il la met sur sa page web, il l'envoie par e-mail, etc. Cette clé peut être connue de tous. La clé privée est gardée secrète par Krabs.
2. Bob chiffre son message m en effectuant $K^{pub}_k(m)$ et l'envoie à Krabs.
3. Krabs applique sa clé privée au message chiffré $K^{priv}_k(K^{pub}_k(m)) = m$ et obtient ainsi le message en clair.

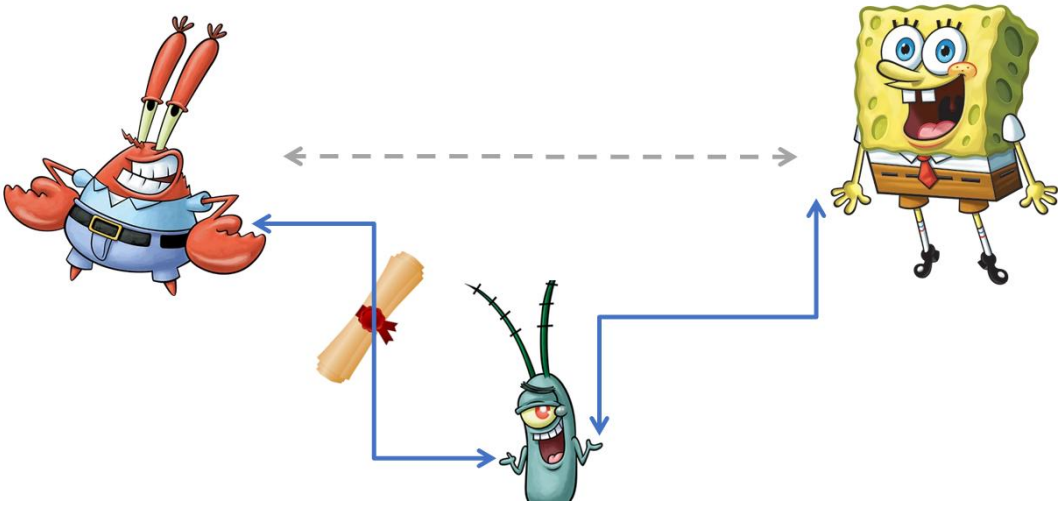
Comme on le voit, le système RSA permet de chiffrer des messages sans s'être mis d'accord sur une clé de chiffrement symétrique commune. Ce système possède cependant un inconvénient de taille : le chiffrement et le déchiffrement nécessitent des calculs très coûteux. Cette complexité fait qu'il n'est pas possible d'utiliser RSA pour chiffrer directement des gros volumes de données ou des flux de communication.

Man In The Middle attack

Comme nous l'avons vu, le chiffrement des communications permet de se protéger contre l'espionnage des communications. Considérons maintenant le scénario suivant. Supposons que Krabs veuille dévoiler la recette du pâté de crabe à Bob. Les communications entre Krabs et Bob étant confidentielles, elles doivent évidement être chiffrées, mais cela n'est pas suffisant.

Plankton (le malveillant) peut procéder à l'attaque suivante. Il envoie un e-mail à Krabs, lui demandant de se connecter d'urgence sur le site de partage pour mettre à jour ces données.

Si Krabs n'est pas méfiant, alors il ne se rendra pas compte qu'il est connecté au mauvais site. Plankton pourra donc recevoir tous les messages envoyés par Krabs. Il pourra les retransmettre au véritable site Bob, en se faisant passer pour Krabs. Plankton sera donc dans une position d'intermédiaire entre Krabs et Bob. Il pourra retransmettre passivement les requêtes de Krabs à Bob, ou les modifier avant de les retransmettre :



Ici, le chiffrement n'est d'aucune aide. En effet, même si Krabs établit un canal sûr entre lui et Plankton, et que Plankton en établit un entre lui-même est Bob, l'attaque peut quand même avoir lieu.

La faille fondamentale permettant cette attaque est l'absence d'authentification. Ici, le serveur de Plankton, qui imite le comportement du serveur de Bob, n'a pas eu à prouver qu'il était bien une machine appartenant à Bob.

Certificat et tiers de confiance

Comme on l'a vu, afin de sécuriser entièrement les communications entre deux participants, il faut non seulement les chiffrer, mais aussi que chaque participant puisse s'assurer de l'identité de son correspondant. L'entité qui veut prouver son identité présente un certificat. Ce dernier doit être délivré par une autre entité, en laquelle les deux participants ont confiance, que l'on appelle un tiers de confiance. Si Bob, qui souhaite correspondre avec leCrabeCroustillant.com, veut s'assurer que c'est bien Krabs qui est le propriétaire du site, il devra lui demander un certificat. Ce dernier sera délivré par un tiers de confiance, Carlo. Evidemment, dans le cadre de communication en réseau, le certificat doit être lui aussi numérique. Ces certificats numériques sont créés à partir des clés RSA publiques et privées des participants.

1. Krabs va voir Carlo. Carlo vérifie que Krabs est bien le propriétaire du site leCrabeCroustillant.com. Il peut par exemple constater que Krabs peut administrer le site, rajouter des pages, ou qu'il peut présenter des documents légaux (preuve d'achat du nom de domaine par exemple). Une fois ces vérifications faites, Carlo utilise sa clé privée K^{priv}_C pour chiffrer la clé publique de Krabs : $s = K^{priv}_C(K^{pub}_k)$

Un tel fichier s s'appelle un certificat. Dans ce cas particulier, on dit que Carlo signe la clé publique de K^{pub}_k avec sa clé privée.

2. Bob veut se connecter à leCrabeCroustillant.com. Lors de la connexion, le site lui fournit la clé publique de Krabs, K^{pub}_k ainsi que le certificat s.
3. Bob récupère alors la clé publique de Carlo, en qui il a confiance. Il calcule :

$$K^{pub}_C(s) = K^{pub}_C(K^{priv}_C(K^{pub}_k)) = K^{pub}_k$$

Bob peut alors comparer la clé publique que Krabs lui a fourni et celle résultant du déchiffrement de la signature. Si les deux sont les mêmes, Bob à la garantie que Krabs est bien passé voir Carlo et que ce dernier a bien fait toutes les vérifications nécessaires.

4. Bob ayant authentifié leCrabeCroustillant.com comme étant bien le site détenu par Krabs, elle peut initier avec lui une communication sécurisée.

Le protocole HTTPS

Nous sommes maintenant équipés pour expliquer le fonctionnement du protocoles HTTPS (HyperText Transfer Protocole Secure).

Autorités de certifications et format X.509

Une autorité de certification (AC) est une entité habilitée à délivrer des certificats. Une AC est un tiers de confiance. Parmi les différentes AC, on retrouve :

- Des entreprises spécialisées ;
- Des associations à but non lucratif :
- Des états.

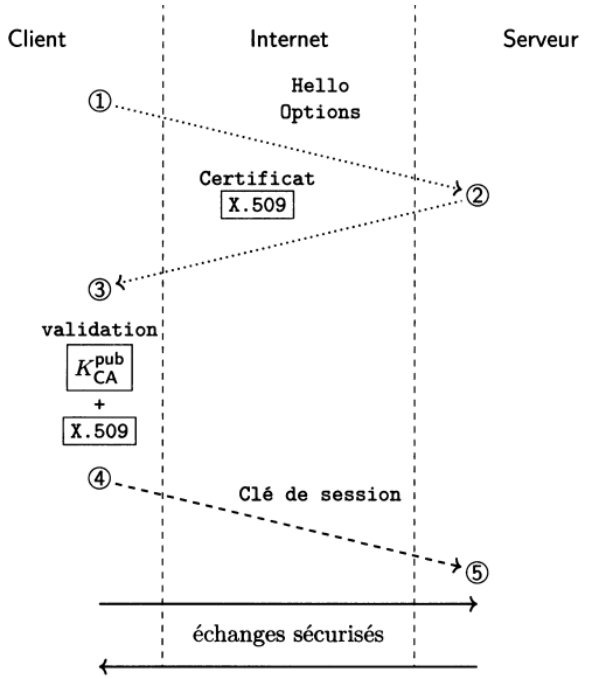
Ces AC émettent des certificats. Ces derniers fonctionnent sur le principe expliqué précédemment, avec quelques considérations techniques supplémentaires. Le format standard de certificat est le format X.509. C'est un format de fichier binaire, contenant entre autres :

- 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 par l'AC.

Le protocole HTTPS qui permet d'établir des communications sécurisées entre un client et un serveur web est en fait simplement le protocole HTTP, auquel une couche de cryptographie a été ajoutée. En effet, HTTPS est simplement la réunion de deux protocoles existants, HTTP, pour l'interaction avec le serveur web et TLS (Transport Layer Security).

L'authentification empêche l'exécution du Man In The Middle attack, et le chiffrement empêche les routeurs et autres machines intermédiaires de lire le contenu des messages.

La phase poignée de main TLS (handshake) se déroule de la manière suivante :



1. Le client envoie un message initial nommé « Hello » ainsi que des options. En particulier, le client indique les différents algorithmes cryptographiques qu'il peut utiliser, ainsi que d'autres paramètres techniques.
2. Le serveur envoie sa réponse, contenant entre autres le certificat X.509 contenant sa clé publique, signée par une AC.
3. Le client vérifie le certificat au moyen de la clé publique de l'AC. Il procède aussi à d'autres vérifications.
4. Le client et le serveur conviennent d'une clé de session pour un algorithme symétrique. Ils peuvent soit choisir de chiffrer une clé choisie par le client avec la clé publique du serveur, soit utiliser une clé de session partagée.
5. Le serveur est authentifié par le client, et les deux ont convenu d'une clé de session. Ils peuvent donc échanger de manière sûre des messages du protocole HTTP en les chiffrant.

Tous ces aspects sont observables en inspectant l'icône de « cadenas » se trouvant dans la barre d'adresse des navigateurs web.

Liens externes

Spécialité Numérique et sciences informatiques : 24 leçons avec exercices corrigés - Terminale - Nouveaux programmes

https://theasciicode.com.ar/ascii-printable-characters/capital-letter-l-uppercase-ascii-code-76.html

https://www.dcode.fr/chiffre-xor

http://www.courstechinfo.be/Techno/PortesLogiques.html

https://fr.wikipedia.org/wiki/Fonction_OU_exclusif