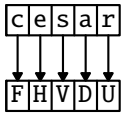
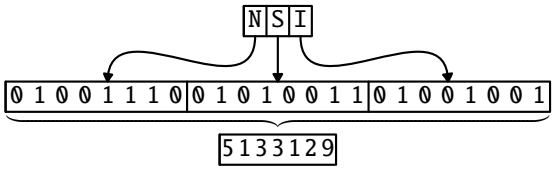
Sécurisation des communications

Cryptographie

Lorsqu'on veut dire un secret à quelqu'un, si on est à côté, il suffit de le lui dire à l'oreille, en vérifiant que personne n'écoute. Mais comment le faire si on ne peut pas se voir physiquement et que le message doit être envoyé et risque d'être intercepté ? Pour cela on peut utiliser la **cryptographie**. Cette science consiste à transformer ou cacher le message pour le rendre incompréhensible par n'importe qui. Seul de destinataire sait comment obtenir le message d'origine. On appelera **chiffrer** l'action de transformer le message pour le rendre incompréhensible et **déchiffrer** le message chiffré pour obtenir le message initial.



Il existe de nombreuses techniques pour chiffrer un message et elles reposent toutes sur un secret qui est partagé initialement entre les deux personnes qui communiquent. On appelle ce secret **la clef de chiffrement**. La méthode de chiffrement la plus connue est la code de César, puisqu'elle aurait été inventée par Jules César. Elle consiste à décaler les lettres de l'alphabet. Par exemple, avec un décalage de 3, A devient D, B devient E, et ainsi de suite. Pour déchiffrer, il suffit de décaler les lettres dans l'autre sens. Cette méthode est efficace si une personne qui intercepte le message ne connaît pas l'algorithme de chiffrement. Sinon, il suffit de tester tous les décalages pour décrypter le message.



Pendant des siècles, l'évolution de la **cryptanalyse**, la science de décrypter les messages, a obligé les cryptographes à créer des méthodes de plus en plus sophistiquées. Le nombre de clefs secrètes a augmenté, jusqu'à arriver à 159×10^{18} de combinaisons pour la machine Enigma utilisée par les Allemands pendant le seconde guerre mondiale. L'arrivée des ordinateurs a démultiplié la puissance de calcul et a permis l'arrivée de nouveaux algorithmes de chiffrement plus complexes. Les textes sont transformés en nombres et ce sont des opérations mathématiques qui sont utilisées pour chiffrer.
De façon générale, pour qu'un algorithme de chiffrement soit intéressant, il doit vérifier les propriétés suivantes :

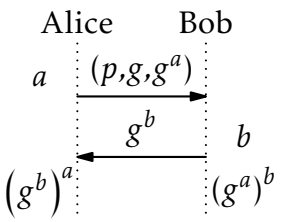
- Sa robustesse ne doit pas reposer sur le secret de la méthode utilisée.
- Chiffrer ou déchiffrer un message doit être rapide.
- Décrypter un message doit être difficile si on ne connaît pas la clef secrète. C'est-à-dire qu'il faut des milliards d'années, en moyenne, pour décrypter un message.

Le problème de l'échange de clefs

Peu importe la technique utilisée, il reste le problème initial du partage de la clef secrète. Imaginons que je veuille communiquer de façon sécurisée avec quelqu'un à l'autre bout du monde, par Internet. Tous les messages que l'on va s'envoyer pourront être lu par quelqu'un qui espionne le réseau. Si nous ne pouvons pas utiliser un autre canal (courier, téléphone…) pour définir une clef secrète, il faut le faire par le réseau. Ce problème était considéré comme insoluble jusqu'à ce que Diffie, Hellman et Merkle proposent une solution en 1976.
Leur protocole repose sur **le problème du logarithme discret**. On prend un nombre premier p et un entier g, appelé **générateur**, tel que $0 < g < p$ et $g^i \ \% \ p \neq 1$ pour tout entier $0 < i < p$. Alors étant donné un nombre k tel que $0 < k < p$, il est très difficile de trouver l'entier a tel

que $k = g^a \% p$. Si Alice et Bob veulent se créer une clef secrète, le principe est le suivant :

- Alice choisit un nombre premier p et un générateur g.
- Elle choisit également un entier a qu'elle garde secret.
- Elle envoie (p,g,g^a) à Bob.
- Il choisit un entier b qu'il garde secret et envoie g^b.
- Chacun peut calculer $k = \left(g^b\right)^a = (g^a)^b$.



À la fin de l'échange, Alice et Bob connaissent tous les deux k, alors que si quelqu'un observe le réseau, il ne peut déduire ni a, ni b et ne peut donc pas calculer k. Ce nombre peut alors servir de clef de chiffrement pour un algorithme de chiffrement. Dans la pratique, on utilise des nombres premiers de 2048 bits, c'est-à-dire qui ont plus de 600 chiffres.

Les algorithmes de chiffrement asymétriques

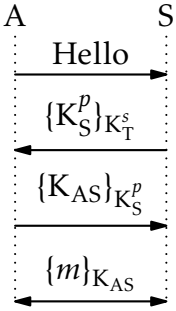
Cette façon de créer un secret n'est pas la seule façon d'échanger de façon sécurisé sur un canal public. En 1977, Rivest, Shamir et Adleman inventèrent RSA, le premier algorithme de **chiffrement asymétrique**. Jusqu'à ce moment là, tous les algorithmes utilisaient la même clef pour chiffrer et déchiffrer. A contrario, dans un système asymétrique, il y a deux clefs. La première est publique et sert à chiffrer, tandis que la seconde est secrète et sert à déchiffrer. Il faut voir cela comme un cadenas et une clef pour l'ouvrir. Le cadenas est public et tout le monde peut l'utiliser pour vérouiller une boîte ou un casier. Seul le possesseur de la clef pourra l'ouvrir et voir le contenu.
Cette fois, l'algorithme repose sur **le problème de la factorisation**. Étant donné un entier $n = pq$, avec p et q premiers, il est difficile de retrouver les deux facteurs.
Il faut d'abord commencer par créer les deux clefs. Alice choisit deux nombres premiers p et q, qui sont secrets, et calcule $n = pq$ et $\varphi(n) = (p-1)(q-1)$. Elle choisit alors un entier e tel que $1 < e < \varphi(n)$ et premier avec $\varphi(n)$. Puis elle calcule son inverse d tel que $ed \% \varphi(n) = 1$. Ce qui est impossible sans connaître p et q. La clef publique est (e,n) et la clef secrète d.
Si Bob veut écrire à Alice, il doit transformer son texte en une liste de nombres m tels que $m < n$. Il calcule pour chacun $c = m^e \% n$, qu'il envoie. Pour déchiffrer, Alice n'a qu'à calculer $m = c^d \% n$. La taille de n est généralement de 1024 ou de 2048 bits.
Un des avantages d'un algorithme comme RSA, c'est qu'il permet de chiffrer les messages ou de les signer. En effet, Alice peut chiffrer un message avec sa clef privée. N'importe qui peut le déchiffrer avec la clef publique et ainsi avoir la certitude que c'est Alice qui a signé le message.

Symétrique ou asymétrique ?

Si les algorithmes de chiffrements asymétriques sont si formidables, alors pourquoi est qu'on utilise toujours des algorithmes symétriques, comme DES ou AES ? Tout simplement parce que les algorithmes de chiffrement symétriques sont plus rapides et plus robustes. Afin d'avoir le même niveau de sécurité, la taille des clefs symétriques sont 10 à 20 fois plus petites. C'est pourquoi, en pratique, les clefs asymétriques sont utilisées pour établir la connexion, construire une clef symétrique. Le reste des communications sont faites avec cette clef symétrique.
De façon générale, on note $\mathrm{K_{AB}}$ la clef symétrique partagée entre A et B, $\mathrm{K_A^p}$ et $\mathrm{K_A^s}$ les clefs publiques et secrètes de A. Les messages chiffrés se notent alors $\{m\}_{\mathrm{K}}$.
Ainsi, seul A peut déchiffrer le message $\{m\}_{\mathrm{K_A^p}}$ et c'est également le seul à pouvoir produire le message signé $\{m\}_{\mathrm{K_A^s}}$

HTTPS

Le protocole HTTP qui permet de naviguer sur le web a un défaut majeur : toutes les données sont envoyées en clair. Ainsi, n'importe qui peut espionner les communications et les données échangées. Le protocole HTTPS vient rajouter une couche de chiffrement, appelé TLS, pour garantir la sécurité des données.



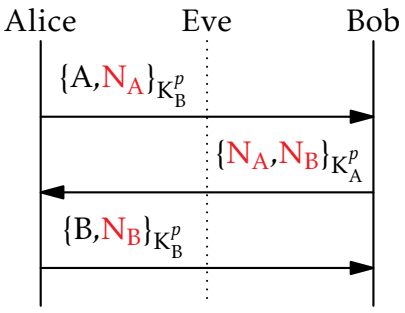
Disons que Alice veuille se connecter à un site se trouvant sur un serveur S. Son navigateur va envoyer un message "Hello" qui signale au serveur qu'elle veut se connecter. Celui-ci répond en envoyant son certificat. Ce certificat $\{K_S^p\}_{K_T^s}$ est en fait sa clef publique signée par un tiers de confiance, qui est une autorité reconnue et garantissant l'autenticité de la clef. Le certificat contient également quelques informations complémentaires comme la date de validité du certificat. Une fois que Alice reçoit ce certificat, son navigateur vérifie sa validité, déchffre la signature avec K_T^p et obtient ainsi K_S^p. Elle génère une clef symétrique K_{AS} pour la suite de la session et renvoie $\{K_{AS}\}_{K_S^p}$, que seul le serveur peut déchiffrer. Ils pourront ensuite communiquer pendant toute la session avec cette clef symétrique.

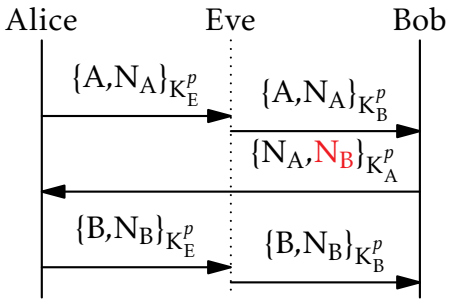
La définition de HTTPS permet d'avoir une grande liberté dans le choix des algorithmes de chiffrement utilisés, ce qui lui permet d'évoluer au fur et à mesure des améliorations de ces algorithmes.

Les protocoles cryptographiques

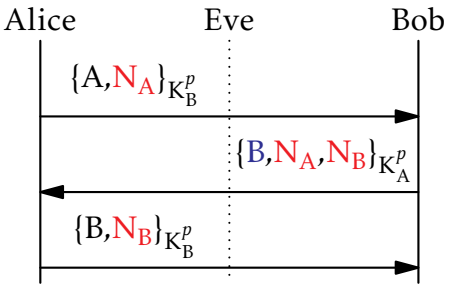
HTTPS est un exemple de **protocole cryptographique**. Il définit un ensemble de messages à échanger entre les différentes parties et les types de chiffrements à utiliser. Un protocole cryptographique a pour but de garantir un certain nombre de propriétés, comme le secret, l'authentification, la non-répudiation ou l'anonymat. Ces propriétés reposent sur la cryptographie.

Par exemple, le protocole de Needham-Schroeder, publié en 1978, a pour but de garantir l'authentification de A et de B entre eux. Les messages sont décrits ci-contre. Les valeurs N_A et N_B sont des **nonces**, c'est-à-dire des nombres générés aléatoirements pour cette session. Ils pourront ensuite utiliser pour générer une clef symétrique et doivent donc rester secrets. Ce protocole a été utilisé par des années sur le web avant qu'on découvre une faille.





En effet, en 1995, Gavin Lowe montre que ce protocole est vulnérable à une attaque de type **man-in-the-middle**. Dans cette attaque, Eve réussit à se faire passer pour Alice auprès de Bob.



Lowe propose une simple correction. Il suffit à Bob de préciser son identité dans sa réponse pour être sûr que Alice parle à la bonne personne.

Cet exemple illustre le fait que même si la cryptographie est parfaite, la conception des protocoles peut également comporter des failles et qu'il est nécessaire de vérifier leur sécurité.