



HTTPS

HyperText Transfer Protocol Secure

Pour plus de sécurité, adoptez les réflexes CASES !

Table des matières

1. C'est quoi ?
2. Comment cela fonctionne-t-il ?
3. Menaces contrées et recommandations

HTTPS est un protocole réseau utilisé pour la navigation sécurisée sur le WWW. Il offre des possibilités d'authentification et de chiffrement pour les sites web nécessitant un certain niveau de sécurité dans leurs échanges avec les navigateurs web. Pour garantir cette sécurité, HTTPS fait usage de méthodes de cryptographie asymétrique pour l'authentification et de méthodes de cryptographie symétrique pour le chiffrement des échanges.



C'est quoi ?

Le protocole HTTP est utilisé par les navigateurs pour récupérer du contenu sur le web (pages HTML, images, applets, fichiers multimédias, etc.) ainsi que pour transmettre des informations, par l'intermédiaire de formulaires par exemple. Puisque cette communication n'est pas sécurisée, une personne malintentionnée pourrait l'intercepter et en extraire des informations, telles que des noms d'utilisateurs et les mots de passe associés. Elle pourrait également altérer les informations transmises ou même se faire passer pour l'une des deux parties.

Applet : petit programme informatique téléchargé via un réseau (notamment Internet) qui est exécuté par une application (généralement le navigateur). L'exécution d'applets permet d'accroître les capacités de l'application.

Pour résoudre ce type de problèmes, SSL/TLS fait **usage de méthodes cryptographiques** et apporte plusieurs caractéristiques supplémentaires au protocole HTTP original :

1. **Authentification** de l'une ou des deux parties communicantes ; soit le serveur seul est authentifié auprès du client, soit client et serveur sont authentifiés l'un auprès de l'autre.
2. **Confidentialité** des échanges ; dans le cas où la communication serait interceptée par un attaquant, celui-ci n'a pas la possibilité d'en déchiffrer le contenu.

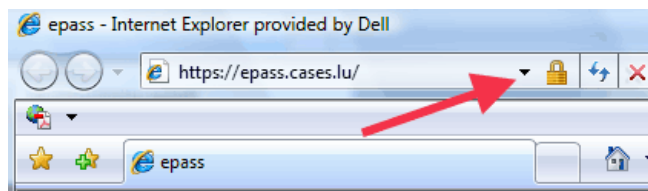
3. Garantie **d'intégrité** des échanges ; client et serveur ont la garantie que les données échangées au cours de leurs communications n'ont pas été altérées par un tiers.

Les méthodes cryptographiques employées sont la cryptographie symétrique pour le chiffrement des données transmises et la cryptographie asymétrique (par l'usage de certificats) pour la signature de la clé de session.

HTTPS, de par ses caractéristiques, est utilisé dans toute une série de domaines pour lesquels les échanges sur le web doivent être sécurisés tels que les applications d'e-banking, le commerce électronique, l'e-mail, et bien d'autres. Pour savoir si le site que l'on est en train de visiter utilise HTTPS il suffit de vérifier **la présence du petit cadenas fermé**.

Car si le cadenas n'est pas fermé ou s'il est barré indique que le certificat présenté par le serveur ne respecte pas tous les critères nécessaires à sa validité et dès lors, il est par exemple possible que le serveur contacté ne soit pas celui qu'il prétend être.

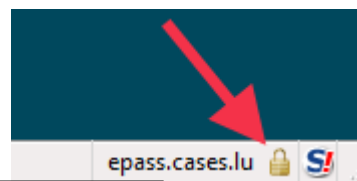
Internet Explorer avant la version 7, le cadenas apparaît dans le bas de l'écran.



Internet Explorer version 7, le cadenas apparaît dans le haut de l'écran.



Mozilla Firefox, le cadenas apparaît au fond de l'écran.



Opera, le cadenas est situé à droite de la barre d'adresse.

2 Comment cela fonctionne-t-il ?

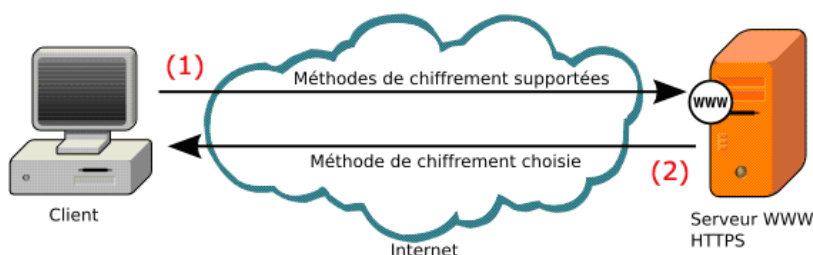
LE MODELE CLIENT-SERVEUR

Dans ce modèle, chaque ordinateur est soit un client, soit un serveur

Un serveur est un ordinateur passif qui attend de pouvoir servir une requête en provenance d'un client. Lorsqu'une requête arrive, il la traite et la sert, puis se remet en attente de la suivante. Plusieurs requêtes provenant d'un même client ou de plusieurs d'entre eux peuvent être servies en parallèle.

Un client est un ordinateur actif qui envoie des requêtes à un serveur et attend d'être servi.

Une fois la réponse du serveur connue, il peut la traiter et en présenter une autre. Plusieurs requêtes peuvent être envoyées simultanément à un même serveur ou à plusieurs d'entre-eux.



Les serveurs sont généralement vus comme étant des machines puissantes avec beaucoup d'espace disque et de mémoire à disposition. Par contre les clients sont considérés comme étant des stations

de travail standards comme des PC par exemple. L'émergence des réseaux peer-to-peer a montré au grand public que tout type d'ordinateur pouvait facilement se transformer en serveur.

HTTP

APERÇU

HTTP est un protocole Client-Serveur standardisé par l'IETF, c'est le plus couramment employé pour surfer sur le web.

Son utilisation consiste en la récupération de fichiers et de flux de données sur le web. Il permet

également d'envoyer à partir d'un client, des fichiers ou des informations via des formulaires. L'utilisation de la version 1.1 d'HTTP est recommandée (voir annexe).

FONCTIONNEMENT

Le fonctionnement d'HTTP est basé sur le modèle Client-Serveur. **Le client est toujours l'initiateur d'une communication**, car c'est lui qui envoie une requête au serveur. Ce dernier réagit en fonction de la requête reçue. Tout comme dans le modèle Client-Serveur, le serveur n'a jamais l'initiative et reste donc passif.

Une requête HTTP est simplement un message **en texte clair** envoyé à un serveur web. Différents types de requêtes existent dans ce protocole. Deux d'entre-elles sont couramment utilisées par les utilisateurs du WWW. Il s'agit des requêtes **"GET"** et **"POST"**. Une requête GET sert principalement à récupérer du contenu en provenance d'un serveur WWW. Elle peut également, tout comme une requête POST, servir à envoyer le contenu d'un formulaire à un serveur.

L'essentiel de la représentation d'une requête GET sur l'URL

<http://www.cases.public.lu/functions/glossaire/index.php>:

```
GET /functions/glossaire/index.php HTTP/1.1
Host: www.cases.public.lu
```

Le but de cette requête est de récupérer le fichier index.php dans le répertoire /functions/glossaire sur le serveur www.cases.public.lu en utilisant la version 1.1 du protocole HTTP. L'essentiel de la réponse du serveur sera "200 OK" suivi du contenu du fichier demandé, index.php dans ce cas-ci.

Une requête GET peut également être utilisée pour transmettre des informations à partir du client vers le serveur, via un formulaire. Une telle requête pourrait par exemple correspondre à une recherche dans le glossaire présent sur le site www.cases.public.lu.

```
GET /functions/glossaire/index.php?letter=A
HTTP/1.1
Host: www.cases.public.lu
```

Par exemple, la requête précédente va renvoyer un document html contenant la liste des mots présents dans le glossaire et commençant par la lettre "A". L'information transmise à www.cases.public.lu est donc la lettre "A" passée par l'intermédiaire de la variable "letter". Il est possible de transmettre plusieurs informations en Une

requête de type **POST** peut également être utilisée pour transmettre des informations à un serveur web. Le format d'une telle requête correspond pour l'essentiel à ceci :

Le résultat sera la liste des définitions contenant le mot "ordinateur".

D'un point de vue utilisateur côté client, la seule différence entre ces 2 types de requêtes intervient au niveau de ce qui est visible dans la barre d'adresse du navigateur utilisé. Dans le cas de la requête de type GET, l'URL visible est <http://www.cases.public.lu/fonctions/glossaire/index.php?letter=A>. Par contre, dans le cas où le type de la requête est POST, l'URL visible est <http://www.cases.public.lu/fonctions/glossaire/index.php>. Le type de requête à utiliser sur un site web dépend des choix faits par les développeurs de celui-ci.

une seule requête en séparant les couples variable/valeur avec le caractère "&". Le nom des variables, tout comme le nom des formulaires n'a pas d'importance, c'est simplement une convention définie par les créateurs du site WWW en question.

```
POST /fonctions/glossaire/index.php HTTP/1.1
Host: www.cases.public.lu
search=ordinateur
```

L'avantage de l'utilisation de la requête GET pour la transmission d'informations à partir du client, est que l'URL présente dans la barre d'adresse du navigateur peut être réutilisée comme bookmark. Cette utilisation n'est cependant pas souhaitable pour des transactions sensibles telles que les transactions bancaires..

SSL/TLS

HISTORIQUE

Le protocole SSL (Secure Socket Layer) a initialement été développé par la société Netscape pour son navigateur WWW. La version 3 fût

reprise et étendue par l'IETF (Internet Engineering Task Force) pour le développement et la standardisation de TLS version 1.

CERTIFICATS ET PKI

Avant de pouvoir communiquer de manière sécurisée, il est nécessaire de mettre en place un moyen d'établir une relation de confiance entre les différents acteurs. Ceci est réalisé au moyen d'une ICP (Infrastructure à Clés Publiques ou encore Public Key Infrastructure (PKI)). Dans ce type d'infrastructure, une autorité de certification est en charge de la gestion des jeux de clés privées-publiques attribués aux différents acteurs. Chaque client d'une ICP reçoit deux clés. La clé privée doit être protégée par le client. La clé publique cependant est incorporée dans un document signé

par l'autorité de certification appelé certificat. Les certificats sont mis à disposition de toutes les personnes intéressées, ils sont publics. Chaque acteur a donc la possibilité de vérifier la validité du certificat auprès de l'ICP. Ceci permet par exemple à un client de vérifier l'authenticité du certificat présenté par un serveur web. Il en va de même dans l'autre sens : un serveur web recevant le certificat d'un client s'y connectant, peut en vérifier l'authenticité en vérifiant la signature que l'autorité de certification y a apposée.

Les certificats utilisés par HTTPS respectent la norme X509 version 3. Un tel certificat contient les informations suivantes :

- Le nom du serveur pour lequel ce certificat à été créé. (p.ex : www.cases.public.lu)
- Le nom de l'émetteur du certificat. Par exemple : LuxTrust SA.
- Le numéro de série du certificat.
- La date à partir de laquelle le certificat est valide et la date à partir de laquelle il sera considéré comme étant expiré.
- La liste des utilisations pour lesquelles ce certificat à été émis. Par exemple : serveur web.
- La clé publique du serveur, ainsi que le nom de l'algorithme avec lequel s'utilise cette clé. Par exemple RSA.
- La signature de l'émetteur du certificat, ainsi que le nom de la méthode utilisée pour la générer. Par exemple: MD5 avec RSA. (cf le dossier sur la cryptographie)

Pour qu'un certificat puisse être considéré comme valide, plusieurs critères doivent être respectés :

- Le nom du serveur web inscrit dans le certificat doit correspondre au nom réel du serveur web,
- La date du jour doit être située entre la date de validité et la date d'expiration,
- La signature de l'ICP émettrice doit pouvoir être vérifiée. Pour ce faire, le navigateur web du client va utiliser le certificat de l'ICP émettrice. Les navigateurs web sont fournis avec les certificats des ICP les plus connues

FONCTIONNEMENT

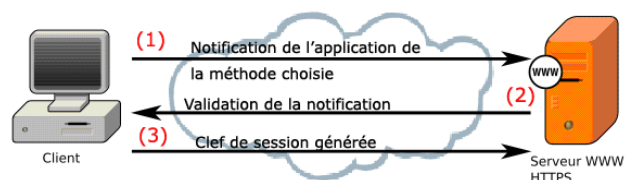
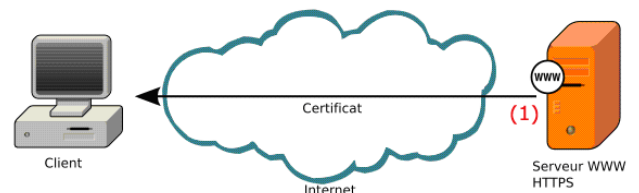
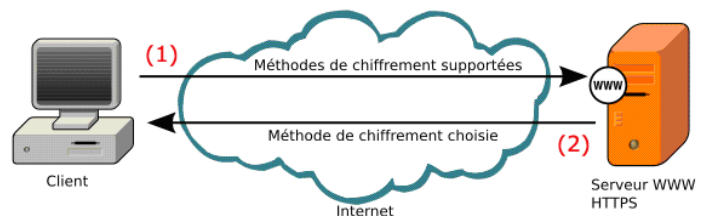
La vie d'une communication SSL, appelée session, passe par plusieurs étapes. Il y a tout d'abord l'établissement de la session, ensuite la transmission d'informations et finalement la clôture de la session.

L'établissement d'une communication SSL/TLS peut être décrite en trois étapes. Elle commence par la négociation de certains paramètres qui seront utilisés dans la suite des échanges. Ce sont notamment le choix des méthodes de chiffrement et de signature. Pour ce faire, le client envoie simplement la liste des différentes méthodes qu'il supporte au serveur. Pour compléter la négociation, celui-ci fait son choix parmi ce qui lui est présenté et indique en retour sa sélection au client.

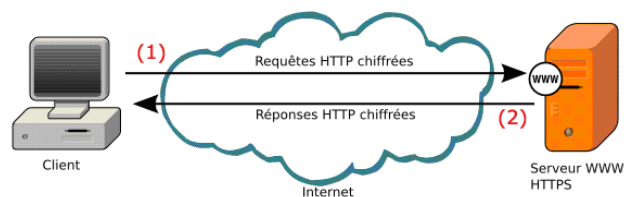
Lors de la seconde étape, le serveur peut envoyer son certificat au client et par la suite demander à celui-ci de lui en fournir un en retour. Ces certificats sont utilisés pour l'authentification des deux parties communicantes

La troisième et dernière étape de l'établissement de la connexion SSL/TLS, consiste pour le client, comme pour le serveur, à avertir l'autre partie que les paramètres négociés peuvent être appliqués et

que le transfert de données peut effectivement commencer.



Dans le cas où il a été convenu entre les deux parties qu'un chiffrement des informations transmises est nécessaire, une clé de session est générée et chiffrée par le client à l'aide de la clé publique du serveur (récupérée dans son certificat). Le serveur étant le seul à posséder la clé privée correspondante, sera aussi le seul à pouvoir déchiffrer la clé de session. Cette clé de session va être utilisée, pour des raisons d'efficacité, comme clé de chiffrement symétrique pour les informations échangées par la suite.



VÉRIFICATION DU CERTIFICAT

Avant de se fier à la sécurité du protocole HTTPS, il est nécessaire de vérifier le certificat d'authentification du serveur. Le fait que le cadenas ne soit pas fermé ou qu'il soit barré indique que le certificat présenté par le serveur ne

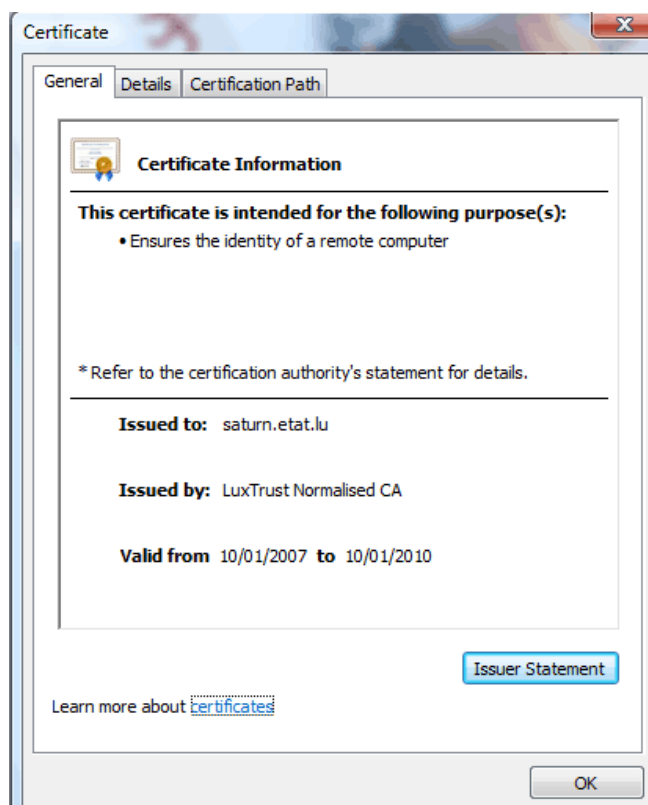
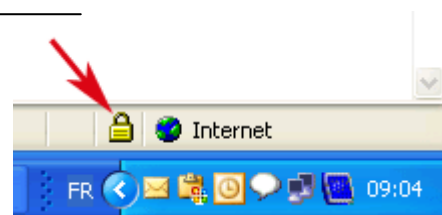
respecte pas tous les critères nécessaires à sa validité et dès lors, il est par exemple possible que le serveur contacté ne soit pas celui qu'il prétend être.

Ici nous considérons un exemple de l'application eTVA du gouvernement luxembourgeois, dont l'URL est la suivante : <https://saturn.etat.lu/etva/index.do>

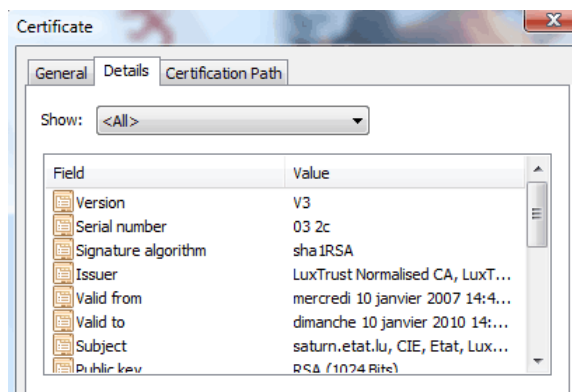
PROCEDURE A SUIVRE POUR MICROSOFT INTERNET EXPLORER

Vérifiez le certificat en double-cliquant sur le cadenas qui se trouve en bas du côté droit de l'Internet Explorer. La présence du cadenas indique l'utilisation du protocole HTTPS.

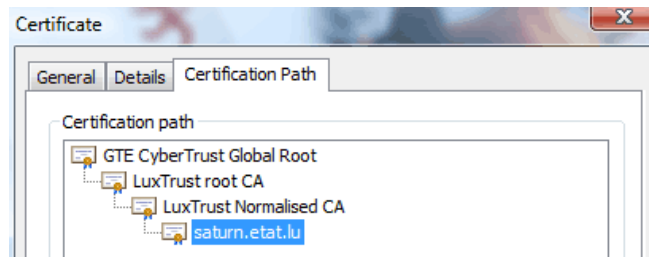
En double-cliquant sur le cadenas, les informations de certificat utilisé vont s'afficher. **Dans l'onglet général**, vous recevez un aperçu du certificat. Le détenteur du certificat, l'émetteur du certificat ainsi que la validité sont affichés. Ce sont des informations que vous devez nécessairement vérifier avant de vous fier à la communication. Dans notre exemple, le détenteur du certificat est identique au serveur sur lequel nous nous trouvons : saturn.etat.lu. La date de validité n'est pas expirée (date de consultation : 02.09.2008). La société d'émission du certificat (LuxTrust) est une société mondialement reconnue.



Dans l'onglet **détails**, vous recevez de plus amples détails sur le certificat.

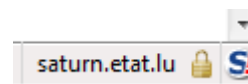


Dans l'onglet **Certification path**, vous pouvez consulter la hiérarchie de certificats utilisés pour assurer la confiance du certificat en question. Dans notre exemple le certificat a été émis par la société LuxTrust. Le certificat de LuxTrust Normalised CA a été signé par LuxTrust root CA qui lui a été signé par GTE CyberTrust Global Root.



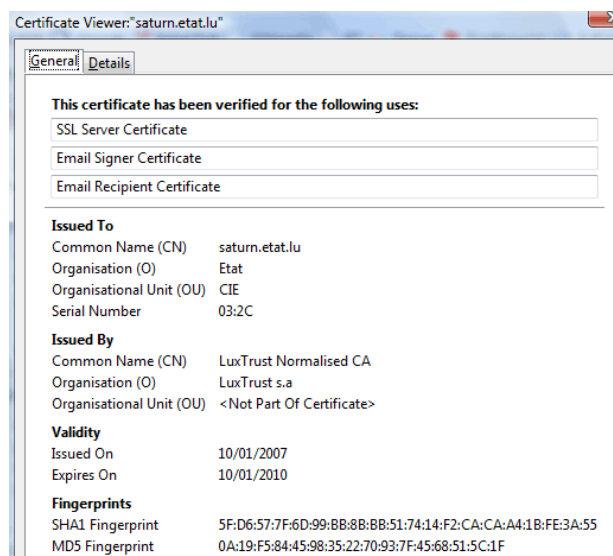
PROCÉDURE À SUIVRE POUR FIREFOX

Vérifiez le certificat en double-cliquant sur le cadenas qui se trouve en bas du côté droit de FireFox. Ce cadenas indique que HTTPS est utilisé.

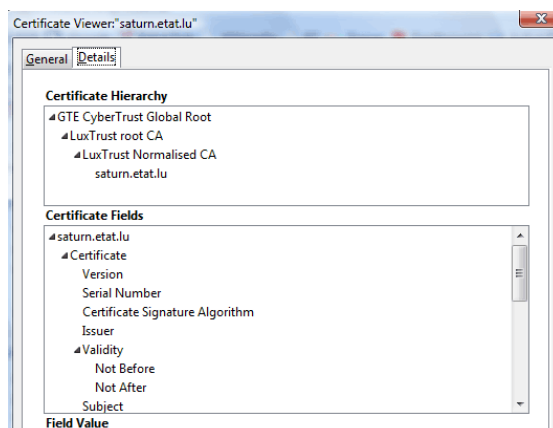


Cliquez sur "voir" pour accéder aux informations relatives au certificat

Dans l'onglet **général** vous recevez un aperçu du certificat. Le détenteur du certificat, l'émetteur du certificat ainsi que la validité sont affichés. Ce sont des informations que vous devez nécessairement vérifier avant de vous fier à la communication. Dans notre exemple, le détenteur du certificat est identique au serveur sur lequel nous nous trouvons : saturn.etat.lu. La date de validité n'est pas expirée (date de consultation : 08.09.2008). Et la société d'émission du certificat (LuxTrust) est une société mondialement reconnue.



Dans l'onglet **Détails**, vous pouvez consulter la hiérarchie de certificats utilisés pour assurer la confiance du certificat en question. Dans notre exemple le certificat a été émis par la société LuxTrust. Le certificat de LuxTrust Normalised CA a été signé par LuxTrust root CA qui lui a été signé par GTE CyberTrust Global Root.



MENACES CONTREES

Les différentes menaces contrées par l'utilisation de HTTPS par rapport à HTTP sont liées aux quatre caractéristiques supplémentaires apportées par l'utilisation de moyens cryptographiques dans SSL/TLS :

1. **Chiffrement** : Grâce au chiffrement, les informations transmises sur le réseau sont illisibles pour un observateur ne possédant pas les clefs cryptographiques utilisées. Cet observateur ne verra qu'une suite d'octets lui semblant distribués aléatoirement. L'utilisateur d'HTTPS a donc la garantie que ses échanges ne seront pas épiés par un tiers extérieur à la communication.
2. **Intégrité** : Cette caractéristique offre la garantie que les informations en provenance du canal sécurisé n'ont pas été altérées, soit par le fait du hasard d'une erreur technique (matérielle ou logicielle), soit par le fait de l'action volontaire d'un tiers (modification intentionnelle). Avec HTTPS, l'utilisateur est donc certain d'être protégé contre l'altération de ses communications.
3. **Authentification** : L'intégrité ne donne aucune garantie concernant la source de l'information reçue. La caractéristique d'authentification offre la certitude que le message reçu vient bien d'une machine possédant la clé privée correspondant à la clé publique qui a servi à établir la communication. Dans ce cas, le client HTTPS a la garantie de communiquer avec le bon serveur.

La non-répudiation n'est pas établie dans HTTPS. Seuls les échanges lors de l'établissement de la session SSL/TLS sont signés. Le reste des données échangées ne l'est pas. C'est pour cela que, par exemple, les banques ont développé d'autres moyens techniques pour offrir la possibilité d'effectuer des transactions à leurs clients.

RECOMMANDATIONS

Il est souhaitable de prendre le temps de vérifier la validité des certificats utilisés lorsque des informations sensibles doivent être échangées avec un site web (numéro de carte VISA, informations personnelles, etc.).

Ce n'est pas parce qu'un site dispose d'un certificat valide que le bon usage des données transmises est garanti. Il se peut très bien qu'un site frauduleux ait acheté un certificat valide et donc qu'il puisse faire croire à ses clients qu'ils se trouvent sur un site WWW de confiance. Il vaut toujours mieux traiter avec un site ayant une réputation sans faille et bien établie plutôt qu'avec un site nouvellement arrivé et dont personne ne sait rien.

ANNEXES

HTTP a été standardisé par l'**IETF** (Internet Engineering Task Force). La version 1.0 est spécifiée dans le document RFC 1945 et la version 1.1 dans le document RFC 2616. (L'usage de la version 1.1 étant actuellement recommandé).