HAL
open science

Certificats (électroniques) : Pourquoi ? Comment ?

Jean-Luc Archimbaud

To cite this version:

Jean-Luc Archimbaud. Certificats (électroniques) : Pourquoi ? Comment ?. 2000. hal-00561730

HAL Id: hal-00561730

https://hal.science/hal-00561730v1

Preprint submitted on 1 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Certificats (électroniques)

Pourquoi ? Comment ?

Jean-Luc Archimbaud CNRS/UREC [jla]

22 décembre 2000 (V3)

Ce document est disponible en ligne (avec une version peut-être plus récente) : http://www.urec.cnrs.fr/securite/articles/certificats.kezako.pdf

De nombreux articles ont déjà été écrits au sujet des certificats, des autorités de certification, des infrastructures de gestion de clés, … [Articles UREC]. Les uns sont techniques et supposent que le lecteur connaît déjà tous les mécanismes sous-jacents. Ils s'adressent plutôt aux spécialistes du domaine. Les autres, à l'opposé, décrivent les applications et les nouvelles fonctionnalités qu'apportent les certificats mais ne parlent ni des principes, ni de l'environnement nécessaire. Comme toutes les nouveautés plutôt complexes, ce domaine demande à être présenté sous plusieurs éclairages. On tente ici un autre angle d'approche.

Cet article essaie de répondre de manière simple aux questions que peuvent se poser les personnes qui désirent **comprendre le sujet** sans se plonger dans les détails. Ce sont les décideurs qui vont définir la manière d'introduire ces objets dans leur administration ou leur entreprise, les ingénieurs qui vont les mettre en place mais aussi les utilisateurs éclairés qui veulent les utiliser à bon escient. Il devrait répondre à deux questions : comment ça marche ? A quoi ça sert ?

Les certificats sont à la base des outils de sécurité. Mal compris, mal conçus, mal administrés, mal utilisés, ils peuvent donner un leurre de sécurité sans en ajouter aucune, voire au contraire simplifier la vie des pirates. Donc le « comment ça marche ? » est important pour tous les décideurs et tous informaticiens qui les installeront et qui formeront les utilisateurs.

Cet article tente donc d'éclairer le lecteur sur **les principes des certificats et des mécanismes associés** à leur utilisation, **des utilisations possibles** de ces certificats, **des infrastructures à mettre en place** pour les gérer, **des applications actuelles** qui peuvent les utiliser, mais aussi **des limites** de tout cet ensemble. Ce document n'est pas destiné aux spécialistes de sécurité et simplifie volontairement certains points pour les rendre plus accessibles à tous.

(NDLR : Attention, le sujet n'est pas trivial. Si vous ne vous êtes jamais penché sur ces mécanismes et que vous comprenez sans effort l'intégralité, bravo !).

1. Définitions des services de base en sécurité

Dans ce document certains termes de sécurité [Glossary] seront très souvent utilisés. Il est nécessaire de les définir avec le sens dans lequel ils seront employés.

L'**authentification** est l'assurance de l'identité d'un objet, généralement une personne, mais cela peut aussi s'appliquer à un serveur, une application, … Dans la vie courante, la présentation de la carte nationale d'identité et la signature manuelle assurent un service d'authentification.

L'**intégrité** d'un objet (document, fichier, message …) est la garantie que cet objet n'a pas été modifié par une autre personne que son auteur. Sur une feuille de papier toute modification est visible d'un simple coup d'œil. Sur un document électronique (courrier électronique, fichier Word, …) non sécurisé, cette détection est impossible.

Sur les documents électroniques, la **signature électronique** est un des mécanismes qui permet d'assurer l'authentification de l'émetteur et l'intégrité de l'objet transmis.

La **confidentialité** est l'assurance qu'un document ne sera pas lu par un tiers qui n'en a pas le droit. Les documents papiers qui doivent rester secrets sont généralement stockés dans des coffres et sont transportés sous plis cachetés.

Sur les documents électroniques, le **chiffrement** permet d'assurer la confidentialité.

Une quatrième fonction de sécurité est appelée « **non répudiation** ». Comme ce terme l'indique, le but est que l'émetteur d'un message ne puisse pas nier l'avoir envoyé et le récepteur l'avoir reçu. Les transactions commerciales ont absolument besoin de cette fonction. Le reçu que l'on signe au livreur, la lettre recommandée sont des mécanismes de non répudiation. Les certificats permettent d'assurer ce service. Dans la communauté enseignement-recherche, cette fonction n'est pas primordiale si ce n'est pour certains actes administratifs (votes, notations, transferts de crédits, …). Pour alléger le nombre de pages, nous ne développerons pas cette fonction dans cet article.

Ces besoins ont toujours existé pour les documents « papier ». Pourquoi y a-t-il un nouveau problème ?

La nouveauté est que les documents manipulés sont maintenant électroniques, ils se présentent sous forme de fichiers stockés sur des ordinateurs. Situation aggravante pour la sécurité, **ils circulent en clair sur les réseaux informatiques**. Or il est techniquement possible, pour une personne mal intentionnée vis à vis d'une autre personne, d'accéder aux fichiers de celle-ci sur son poste de travail pour les lire ou les modifier, d'écouter ses communications Internet ou Intranet, ... Si des protections n'ont pas été installées et certaines précautions prises, ces infractions sont même simples à réaliser.

Il a donc été nécessaire de créer un ensemble de mécanismes pour assurer les 4 fonctions de sécurité sur les documents et les nouveaux modes de communication électroniques. Ceux-ci peuvent reposer sur les certificats.

N'oublions pas le dernier terme préféré du spécialiste sécurité : le «**contrôle d'accès** ». On regroupe sous ce vocable, les mécanismes logiciels ou matériels qui permettent d'autoriser ou pas l'accès à différentes ressources (messages, fichiers, bases de données, applications, machines, réseaux, …). L'accès peut être autorisé ou non suivant l'utilisateur, la machine de l'utilisateur, le réseau de l'utilisateur, … Les mécanismes de contrôle d'accès sont ainsi très divers ; pour la sécurité physique : serrure, lecteur de badge, … ; pour la sécurité logique : filtres dans les équipements de communication pour laisser passer ou non certains trafics réseau, mot de passe, carte à puce, … Là aussi les certificats peuvent simplifier ces mécanismes.

(NDLR : pour l'instant c'est plutôt simpliste comme article)

2. Quelques lacunes des applications réseau actuelles

Si l'on prend comme exemple le CNRS [CNRS], les 1300 laboratoires de cet organisme sont connectés à l'Internet par Renater [Renater]. Néanmoins, toutes les communications qui pourraient emprunter cet outil très performant et économique, en particulier celles avec un caractère officiel, ne passent pas par le réseau. Pourquoi ?

Prenons quelques exemples, dans les applications de gestion mais aussi dans l'informatique scientifique.

Au CNRS, la **diffusion des notes officielles** se fait uniquement sous forme papier, via le courrier postal. Pourquoi, alors que chaque directeur d'unité et même chaque personnel a une adresse électronique ? Simplement parce que les applications de messagerie couramment utilisées au CNRS, qui sont celles utilisées dans l'Internet, n'assurent aucune des 4 fonctions de base de sécurité : l'authentification de l'émetteur du message, l'intégrité et la confidentialité du message transmis, la non-répudiation. Il est ainsi très facile de se faire passer pour quelqu'un d'autre et d'émettre un message en prenant son identité. Cette grave lacune est connue, mais chaque agent CNRS utilise néanmoins la messagerie électronique, l'outil est vraiment trop pratique. Dans un usage courrant, il y a peu de risque d'usurpation d'identité car pour un pirate le jeu n'en vaut pas la chandelle. Mais on ne peut pas prendre ce risque avec les documents officiels signés par le directeur général ou par un délégué régional. Ce serait de l'inconscience.

Les notes officielles ne demandent généralement que l'authentification de l'émetteur, la signature, et l'intégrité du contenu comme services de sécurité. D'autres documents pourraient circuler sur le réseau si on pouvait garantir en plus la confidentialité. Ce sont par exemple les documents utilisés pour les **élections**, les **notations** et plus globalement la **gestion du personnel** et la **gestion financière**. Actuellement la transmission de ces documents se fait évidemment par courrier postal.

Comme tout organisme, la gestion du CNRS repose sur de nombreuses **applications de gestion** (financière, comptable, …). Chaque application a son propre contrôle d'accès, souvent avec un compte et un mot de passe pour chaque utilisateur. Ce système est lourd car il multiplie l'administration des comptes par le nombre d'applications de gestion. Il engendre aussi une grosse faille en sécurité. En effet, beaucoup de gestionnaires accèdent à plusieurs applications et doivent donc se souvenir de plusieurs mots de passe. De peur de les oublier, ils notent ces mots de passe près de leur poste de travail … avec les dangers que cela représente.

Une des caractéristiques du CNRS est d'être un organisme «ouvert » au sens où il est formé d'unités de recherche propres, uniquement CNRS, mais surtout de nombreuses unités associées (aux universités, à autres organismes de recherche, …). Il est ainsi impossible de délimiter le contour géographique précis du CNRS et de construire un réseau privé informatique regroupant « tous les ordinateurs du CNRS », comme le font les entreprises dans leur Intranet. Dans l'Intranet d'une entreprise toutes les machines sont facilement identifiables. Elles sont numérotées (avec une adresse IP) dans des intervalles connus (les adresses de réseau) et nommées avec un suffixe identique (le nom de domaine). Cet Intranet est connecté avec l'Internet en un ou deux points avec des contrôles d'accès très stricts. Dans cette configuration il est très simple de limiter les accès à des informations (pages Web, bases de données, logiciels, …). Il suffit de vérifier l'adresse IP ou le nom des machines qui essaient d'y accéder. Au CNRS les machines sont numérotées dans des plages d'adresses très variées qui souvent appartiennent aux universités et des noms aussi très divers, souvent des domaines

universitaires. Le suffixe cnrs.fr est utilisé uniquement par les services administratifs et certaines unités propres. Il ne peut pas en être autrement d'après la définition des unités associées et la technologie Internet. Ainsi **le CNRS est privé de toutes les possibilités d'un Intranet** :

- Il est impossible de mettre sur des serveurs Web des informations réservées aux agents CNRS car on ne sait pas en restreindre l'accès à ceux-ci.

- Dans les entreprises, le contrôle d'accès simple avec un Intranet est utilisé pour la diffusion de logiciels achetés avec une licence entreprise ou pour contrôler l'accès à des bases de données payantes. Au CNRS, sans Intranet, il faut par exemple refaire un contrôle avec le nom d'utilisateur et son mot de passe pour la diffusion de logiciels achetés via le ministère MENRT, de même pour l'accès via l'Institut de l'Information Scientifique et Technique (INIST) aux publications scientifiques en ligne de l'éditeur ELSEVIER. Dans les deux cas, le CNRS est lié par un contrat qui limite l'utilisation ou l'accès à son personnel. Il faut donc qu'il mette en place les contrôles nécessaires pour respecter ses engagements.

- Il est très difficile de créer sur des serveurs des espaces de libres échanges électroniques de documents pour des groupes de travail ou des communautés

- …

Sur un autre registre, un problème perdure depuis longtemps : la **transmission du mot de passe en clair sur le réseau**. Ainsi de nombreux chercheurs en mission à l'extérieur et qui se sont connectés sur une machine de leur laboratoire ont eu leur mot de passe découvert par des pirates qui avaient installé des logiciels d'écoute sur le réseau local de leur lieu de mission. Ce mot de passe sert ensuite au pirate à se connecter sur la machine du chercheur. La parade est de trouver une authentification des utilisateurs différente du simple mot de passe ou de faire circuler celui-ci chiffré sur le réseau, mais elle demande des logiciels spécifiques.

Une nouvelle fonctionnalité de sécurité apparaît maintenant comme une priorité dans **les projets CNRS de grilles de calcul et de données** [Datagrid] qui visent à construire une toile planétaire regroupant des capacités de calcul, de mémoire, de stockage, de visualisation de données, ... Ces projets vont nécessiter des contrôles d'accès élaborés que l'on ne sait pas résoudre actuellement. Ces projets préfigurent certainement de nombreuses autres **applications distribuées** qui demanderont à toute personne d'un laboratoire CNRS de pouvoir s'authentifier.

On pourrait trouver une solution sur mesure à chacun des problèmes ci-dessus. Mais chacune ne résoudrait qu'un seul problème tout en étant grosse consommatrice de ressources humaines pour la mise en place et l'administration. Une autre stratégie est d'**avoir une fondation commune qui va permettre de combler toutes ces lacunes. Ce sont les certificats**.

(NDLR : Le chapitre 7 développe comment ceci va être possible).

3. Chiffrement, empreinte, signature, certificats : principes

Ce paragraphe tente d'expliquer simplement, les mécanismes logiques qui permettent de réaliser les fonctions de sécurité : authentification, intégrité, confidentialité, avec des certificats. Il est destiné aux personnes qui ne connaissent pas le domaine de la sécurité informatique.

3.1 Chiffrement

Pour assurer la **confidentialité** d'un document électronique, on chiffre le texte du document. Cette opération consiste à appliquer une fonction mathématique (en fait c'est un ensemble de fonctions) avec des caractéristiques très particulières sur le texte. Cette fonction utilise une variable, la **clé de chiffrement**, qui est une suite de bits quelconque. Une fois le texte chiffré, il est illisible. Pour obtenir la version lisible, il faut le déchiffrer, c'est à dire appliquer une autre fonction mathématique, compatible avec la première, avec une autre variable, la **clé de déchiffrement**. Ces 2 fonctions mathématiques sont appelées **algorithmes de chiffrement**. La valeur de la clé de déchiffrement dépend évidemment de la valeur de la clé de chiffrement et uniquement le possesseur de la clé de déchiffrement peut déchiffrer le texte. Lorsque l'on désire transmettre un document confidentiel à un correspondant à travers le réseau, on chiffre le document sur son poste de travail avec une clé de chiffrement et on envoie la version chiffrée. Le destinataire déchiffre le document sur son poste de travail avec la clé de déchiffrement, qu'il est le seul à connaître. Si une troisième personne intercepte le texte durant le transfert, il ne pourra pas le déchiffrer car il ne connaîtra pas la valeur de la clé de déchiffrement.

Il faut noter que les algorithmes de chiffrement, c'est à dire les formules mathématiques, sont publics et ont fait l'objet de standardisation. C'est le secret de certaines clés qui permet à ces algorithmes d'assurer le service de confidentialité.

Voici les étapes de la transmission d'un texte confidentiel de Philippe à Nicole.

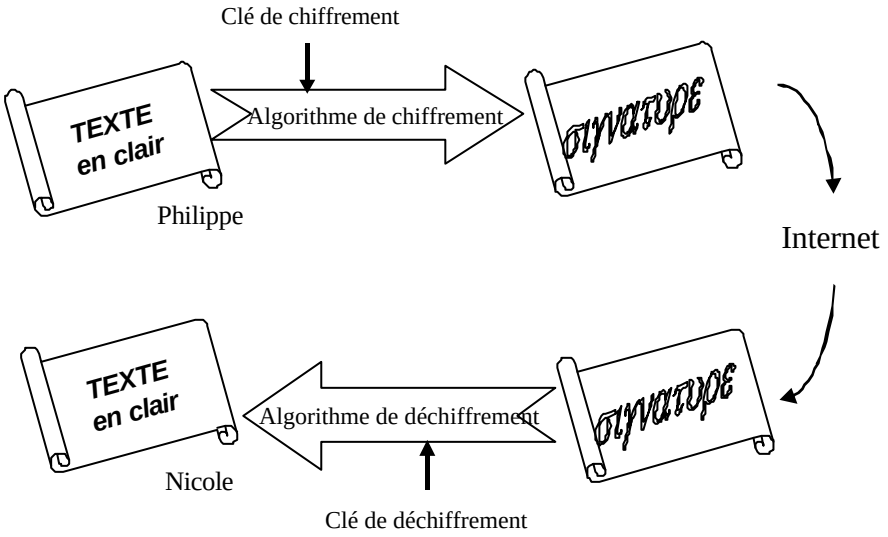

Figure 1 : chiffrement

Il y a deux grandes familles d'algorithmes de chiffrement : symétriques et asymétriques.

3.1.1 Algorithmes symétriques

Dans les **algorithmes symétriques,** aussi appelés algorithmes **à clé secrète, la clé de chiffrement est la même que la clé de déchiffrement**. De ce fait, pour que le texte chiffré ne soit lisible que par le destinataire, la valeur de cette clé doit être un secret partagé entre l'émetteur et le destinataire uniquement. Ceci explique le qualificatif de « clé secrète ». **DES** [DES] est l'algorithme symétrique historiquement le plus connu. Il utilise des clés (de chiffrement et de déchiffrement) qui font 56 bits (c'est la taille réellement utilisée). Une version améliorée, qui le remplace maintenant, **Triple DES [Triple DES] utilise des clés de 112 bits.** Le prochain se nommera peut-être Advanced Encryption Standard [**AES**].

Les premiers algorithmes symétriques datent de la fin des années 70 et utilisent des fonctions mathématiques «simples ». L'avantage est que **les opérations de chiffrement et de déchiffrement sont rapides** à exécuter sur des ordinateurs classiques.

Par contre, **le problème est la gestion des clés**. En effet, chaque clé que l'on utilise avec un correspondant doit être secrète et unique. On a donc autant de clés que de correspondants et il faut trouver un moyen d'échanger chaque clé secrète avec chaque correspondant de manière sûre (pas question d'utiliser le fax du secrétariat ou la messagerie électronique non sécurisée). Si ceci est possible entre un groupe restreint de personnes, c'est impossible à plus grande échelle, par exemple pour échanger des messages chiffrés avec tous nos correspondants sur l'Internet.

3.1.2 Algorithmes asymétriques

L'autre ensemble d'algorithmes sont les **algorithmes asymétriques ou à clé publique**. Ils ont été conçus pour utiliser des clés qui possèdent plusieurs propriétés:

- **La clé de chiffrement est différente de la clé de déchiffrement** (d'où le terme asymétrique).
- **Les 2 clés (une pour chiffrer, l'autre pour déchiffrer)** sont crées ensembles avec une fonction mathématique. Elles **forment un couple**, l'une ne va pas sans l'autre, mais **il est impossible avec une des clés de découvrir l'autre**.
- **Tout texte chiffré avec une des clés (de chiffrement ou de déchiffrement) peut être déchiffré avec l'autre clé (de déchiffrement ou de chiffrement) et uniquement avec celle-ci**.

En pratique, pour utiliser ces algorithmes, il faut générer un couple de clés (l'une pour chiffrer, l'autre déchiffrer) pour chaque utilisateur. La personne le fera elle-même sur sa machine ou quelqu'un de confiance le fera pour elle. Elle gardera sa clé de déchiffrement secrète, on l'appellera ainsi clé privée. A l'inverse elle rentra sa clé de chiffrement publique et la diffusera (on dit aussi la publiera) le plus largement possible. On la trouvera dans des annuaires électroniques par exemple. Ainsi **le couple de clés est formé d'une clé privée secrète pour déchiffrer et d'une clé publique pour chiffrer**.

Maintenant quand Philippe voudra par exemple envoyer un message chiffré avec un algorithme asymétrique à Nicole, son outil de messagerie cherchera dans un premier temps la clé publique de Nicole. L'outil interrogera un annuaire électronique pour trouver cette clé. L'ayant trouvée, il conservera sa valeur dans un répertoire local pour les utilisations futures. Ensuite l'outil chiffrera le texte du message avec la clé publique de Nicole. Ce texte illisible ne pourra alors être déchiffré qu'avec la clé privée de Nicole, que Nicole est la seule à connaître. Ainsi le message pourra transiter via le réseau sans risque d'être déchiffré. Arrivé sur l'ordinateur de Nicole, le texte sera déchiffré avec la clé privée de Nicole.

(NDLR : si, si, ce n'est pas magique, c'est logique et ça marche)

Voici comme exemple, l'émission par Philippe d'un texte chiffré à Nicole avec un algorithme de chiffrement asymétrique.

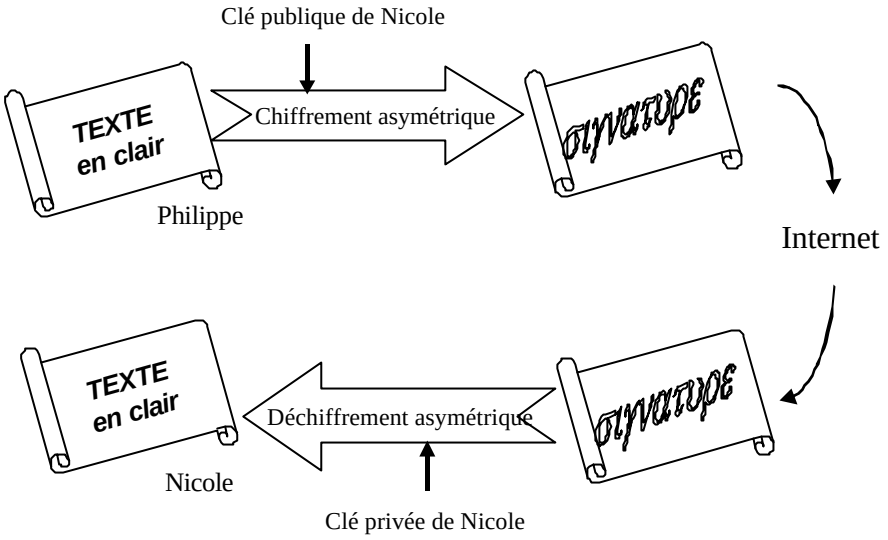

Figure 2 : chiffrement avec algorithme asymétrique

Dans l'autre sens, quand Nicole enverra un texte chiffré à Philippe elle le chiffrera avec la clé publique de Philippe. Celui-ci le déchiffrera avec sa clé privée.

RSA [RSA], du nom des 3 inventeurs Rivest, Shamir, Adleman, est l'algorithme de chiffrement asymétrique le plus répandu.

Ce découplage entre clé publique et clé privée est très utile pour une utilisation « planétaire » du chiffrement. Alors que les algorithmes symétriques obligent à échanger un secret, la clé secrète, avec chaque interlocuteur, là il suffit d'avoir un annuaire qui permette de trouver la clé publique de chaque internaute et ce système peut fonctionner entre tous les internautes. Quand un utilisateur voudra envoyer un message chiffré à un correspondant, il consultera l'annuaire qui lui indiquera la clé publique de son correspondant. Avec cette clé, il chiffrera le message. Celui-ci ne pourra être déchiffré qu'avec la clé privée du correspondant, donc que par le correspondant.

Ainsi **les propriétés des algorithmes asymétriques**, sans intérêt au premier abord, **vont permettre de s'affranchir du problème de la gestion des clés et ainsi d'envisager de déployer l'utilisation du chiffrement à très grande échelle**.

Mais il reste un problème**.** Avec les algorithmes asymétriques, **le temps pour les opérations de chiffrement et de déchiffrement est long.** Ainsi sur un ordinateur courant, RSA (algorithme asymétrique) est de 100 à 1000 fois plus lent que le Triple DES (algorithme symétrique). La clé de session est un moyen pour atténuer ces mauvaises performances.

3.1.3 Clé de session

Pour entre autre, contourner les très mauvaises performances en temps de traitement des algorithmes asymétriques, une astuce est couramment utilisée dans les logiciels. Elle consiste

à utiliser les deux types de chiffrement, asymétrique et symétrique, lors du même transmission.

Pour envoyer un message chiffré, le programme émetteur ne chiffrera pas le message complet avec un algorithme asymétrique. Il chiffrera en asymétrique, avec la clé publique du destinataire, uniquement un petit nombre aléatoire (quelques dizaines de caractères) choisi par lui. Ce nombre servira de clé secrète qui sera utilisée pour chiffrer de manière symétrique le texte. A l'arrivée, le programme du destinataire déchiffrera cette clé secrète, chiffrée « en asymétrique », avec sa clé privée. Munie de la clé sécrète ainsi déchiffrée, il déchiffrera ensuite le message, chiffré « en symétrique ».

Dans ce processus, l'algorithme asymétrique n'est utilisé que sur quelques dizaines de caractères, cette opération sera donc relativement rapide. Le message, qui peut être un fichier de plusieurs gigabytes, sera lui chiffré et déchiffré avec un algorithme symétrique. Le temps de traitement de l'ensemble sera donc du même ordre que si l'on avait utilisé que du chiffrement symétrique.

D'autre part, l'introduction du chiffrement asymétrique permet de s'affranchir du problème de la gestion de la clé secrète des algorithmes symétriques. En effet cette clé secrète n'a pas besoin d'être connue avant la communication par le destinataire. Elle est choisie par l'émetteur et le destinataire en prend connaissance en déchiffrant une partie de ce qu'il a reçu.

Ce processus a deux étapes est employé par les outils sécurisés de messagerie, transfert de fichiers, navigation, … La clé secrète ne servant que pour l'opération en cours, est appelée **clé de session**. Au prochain transfert le programme de l'émetteur choisira une autre clé de session.

Un intrus qui récupère le document chiffré ne pourra pas déchiffrer la clé de session car il ne possède pas la clé privée du destinataire ; et sans cette clé de session, il ne pourra pas déchiffrer le texte du message.

Exemple de transfert d'un texte chiffré avec utilisation d'une clé de session.

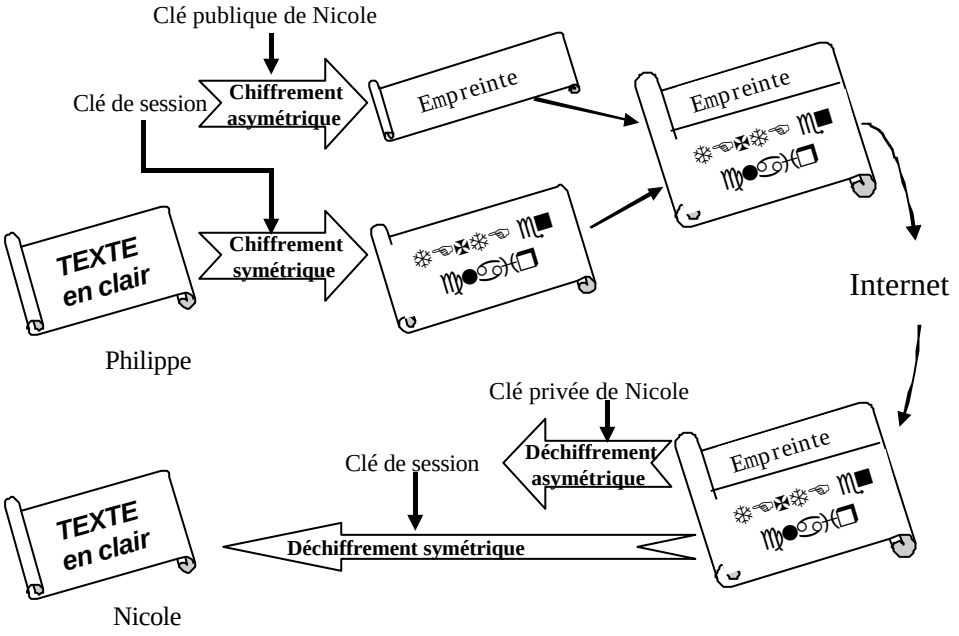

Figure 3 : chiffrement avec clé de session

(NDLR : une petite pause après la lecture de ce schéma ?)

3.1.4 Longueur des clés

De nombreuses techniques ont été imaginées pour essayer de déchiffrer un document chiffré sans connaître la clé de déchiffrement. L'opération s'appelle **décrypter** (bien que souvent par abus de langage décrypter soit utilisé à la place de déchiffrer). Les militaires ont été traditionnellement les spécialistes dans ce domaine, le but étant d'intercepter les transmissions chiffrées de l'adversaire et d'essayer de les décrypter.

Il faut aussi savoir que le décryptage est théoriquement toujours possible. Ce qui le rend en pratique très difficile, voire impossible, est le temps qui est nécessaire pour arriver à décrypter un document chiffré. Si on disposait d'ordinateurs « ultra rapides » avec une puissance de calcul « illimitée », tous les textes chiffrés pourraient être décryptés. Mais ce n'est pas le cas.

D'autre part, **plus la longueur (nombre de bits) de la clé est grande plus il est difficile de décrypter un texte chiffré** (concrètement il faut plus de temps de calcul) avec un algorithme de chiffrement solide (il faut aussi que l'algorithme soit mathématiquement bon). La puissance des machines augmentant, pour garantir cette impossibilité de décryptage, on utilise des clés de longueur de plus en plus grande.

La législation française [Décrets chiffrement] s'est ainsi adaptée pour autoriser l'utilisation des produits de chiffrement avec des clés plus longues, de manière à ce que l'on puisse utiliser des produits relativement surs. La longueur de clé autorisée pour une utilisation libre d'un produit de chiffrement est passée en mars 1999 de 40 à 128 bits.

3.2 Signature électronique

Les paragraphes précédents ont décrit comment est assurée la fonction de confidentialité avec le mécanisme de chiffrement.

La signature électronique est un des mécanismes qui permettent d'assurer les fonctions d'authentification et d'intégrité. Il est utilisé en particulier dans la messagerie électronique.

Pour générer une signature électronique, il faut dans un premier temps utiliser une **fonction de hachage**. C'est une fonction mathématique qui à partir d'un texte de n'importe quelle longueur génère un nombre, suite de bits de taille fixe, bien inférieure à la taille du texte initial. Cette fonction est telle que si un bit du texte d'origine est modifié, le résultat de la fonction sera différent. Cette suite de bits est ainsi appelée **condensé** ou **empreinte. MD5** (MD pour Message Digest) est une fonction de hachage très répandue, elle crée une empreinte de 128 bits [MD5]. **SHA** (Secure Hash Algorithm), autre fonction, crée des empreintes de 160 bits [SHA].

Avant d'envoyer le message, l'outil logiciel émetteur calcule l'empreinte du message, résultat d'une fonction de hachage appliquée au message. Il chiffre ensuite cette empreinte par un algorithme asymétrique avec sa clé privée. Ce résultat est appelé signature électronique. Avant l'envoi, cette signature est ajoutée au message, qui devient un message signé.

Le logiciel du destinataire qui reçoit l'ensemble déchiffre cette empreinte chiffrée avec la clé publique de l'émetteur. Puis il recalcule la fonction de hachage sur le message reçu et compare le résultat avec l'empreinte déchiffrée. Si les deux sont égaux, cela veut dire que le message n'a pas été modifié durant le transfert et que l'émetteur est authentifié.

En effet, si le message a été modifié, les 2 empreintes seront différentes. De plus, être capable de déchiffrer, avec la clé publique d'une personne, une empreinte chiffrée, prouve que cette empreinte a obligatoirement été chiffrée avec la clé privée de la personne, clé que seul possède l'émetteur. Cela authentifie donc l'émetteur. On peut rappeler qu'une des

propriétés du couple clé privée - clé publique est que tout ce qui est chiffré avec une des clés peut être déchiffré avec l'autre clé et uniquement avec celle-ci.

(NDLR : on espère que le schéma suivant sera plus clair que ces explications tordues !)

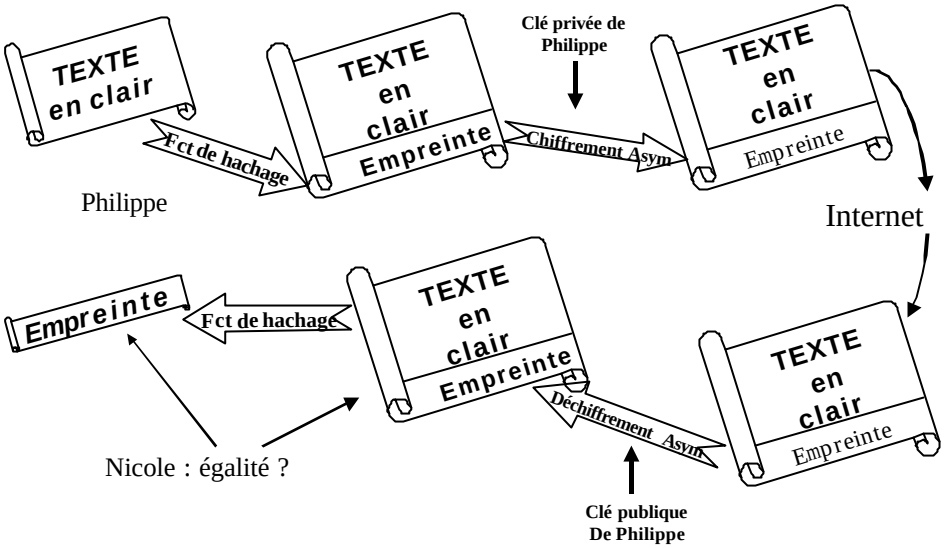

Figure 4 : signature

Nous avons décrit séparément les mécanismes de chiffrement et de signature. Mais les logiciels courants peuvent cumuler les deux fonctions. Ils permettent par exemple de transmettre un message chiffré et signé. Les logiciels appliqueront alors généralement la signature avant le chiffrement à l'émission, et le déchiffrement puis la vérification de la signature à la réception. Mais ils peuvent aussi inverser l'ordre de réalisation de ces opérations.

3.3 Certificats

Nous avons décrit les mécanismes qui permettent d'assurer les 3 fonctions de base de sécurité avec le couple de clés privée–publique et les algorithmes de chiffrement asymétriques. Mais il y a une énorme lacune dans les raisonnements précédents. On a considéré qu'un utilisateur connaissait la clé publique d'une personne simplement en consultant un annuaire ou un serveur Web ou … et ainsi il la considérait comme vraie.

Mais qu'est-ce qui garantit que la clé publique de Philippe qu'un utilisateur a ainsi récupérée est la bonne ? Il ne faut pas oublier que tout ceci fonctionne de manière électronique, sur l'Internet, sans contact direct donc sans moyen visuel de reconnaissance d'une personne. Un pirate, François, a pu modifier l'annuaire ou le serveur Web qui contient la clé publique de Philippe. Il a pu par exemple remplacer la clé publique de Philippe par la sienne. Une fois cette mascarade commise, François pourra lire les courriers confidentiels destinés à Philippe et signer des messages en se faisant passer pour Philippe.

(NDLR : les plus assidus peuvent découvrir eux-même pourquoi, les autres peuvent lire la petite explication qui suit …).

Ce sera la conséquence si un utilisateur, Nicole, croit détenir la clé publique de Philippe alors que c'est celle de François. En effet, si Nicole envoie un message chiffré à Philippe, elle va le chiffrer avec la clé publique de Philippe. Si celle-ci est en fait la clé publique de François, alors François pourra déchiffrer ce message destiné à Philippe avec sa clé privée. François pourra donc lire le courrier confidentiel de Philippe.

Autre possibilité, François pourra envoyer un message signé à Nicole avec une signature générée avec sa clé privée et en se faisant passer pour Philippe. Nicole qui recevra le message vérifiera la signature du message avec ce qu'elle croit être la clé publique de Philippe. La vérification sera correcte, donc Nicole pensera que le message vient de Philippe.

(NDLR : il n'est pas interdit de se faire des petits schémas pour mieux comprendre …).

Il a donc fallu créer un mécanisme supplémentaire, le certificat électronique, pour assurer la validité de la clé publique.

Un certificat est l'équivalent d'une carte d'identité ou d'un passeport. Un passeport contient des informations concernant son propriétaire (nom, prénom, adresse, …), la signature manuscrite, la date de validité, ainsi qu'un tampon et une présentation (forme, couleur, papier) qui permettent de reconnaître que ce passeport n'est pas un faux, qu'il a été délivré par une autorité bien connue.

Un certificat électronique contient des informations équivalentes. Le format reconnu actuellement est le **format X509V3** [X509V3].

(NDLR : l'explication du nom nous conduirait à une époque que les moins de quarante ans ne peuvent pas connaître)

C'est un petit fichier, qui contient au moins les informations suivantes :

- Le nom de l'autorité (de certification) qui a créé le certificat
- Le nom et le prénom de la personne
- Son entreprise (CNRS par exemple)
- Son service (au CNRS, le nom du laboratoire)
- Son adresse électronique
- Sa clé publique
- Les dates de validité du certificat
- Des informations optionnelles
- Une signature électronique

Cette **signature électronique** est calculée sur les informations contenues dans le certificat comme dans le cas d'un message électronique explicité ci-avant. **La signature est l'empreinte de ces informations chiffrée avec la clé privée de l'autorité de certification qui a délivré ce certificat.**

Qu'est-ce qu'une autorité de certification ? L'équivalent de la préfecture pour un passeport. C'est une entité, structure technique et administrative, qui délivre des certificats. Le chapitre 4 explicitera en détails comment tout ceci fonctionne.

Dans l'immédiat, pour notre compréhension, on peut considérer que pour un organisme comme le CNRS, l'autorité de certification est un service du CNRS qui délivre des certificats aux personnes qui travaillent dans les laboratoires CNRS.

Ce service a, préalablement à toute action, généré un couple de clés publique-privée pour lui-même. Ensuite il a très largement diffusé la valeur de sa clé publique pour que tous les agents CNRS en aient connaissance. Concrètement, tous les programmes qui ont des fonctions de sécurité sur les postes informatiques des laboratoires CNRS ont été configurés avec la clé publique de l'autorité de certification CNRS en mémoire. Dans un troisième temps ce service peut alors commencer à créer et délivrer des certificats aux personnels des laboratoires, certificats signés avec la clé privée de l'autorité de certification CNRS.

Parallèlement, les annuaires électroniques CNRS accessibles par l'Internet, sont mis à jour pour contenir le certificat de chaque personnel.

Quand Nicole veut communiquer de manière sécurisée avec Philippe qui travaille dans un laboratoire CNRS, par exemple lorsqu'elle veut lui envoyer un message chiffré, le logiciel de messagerie de Nicole a besoin de connaître la clé publique de Philippe. Si ce logiciel ne connaît pas cette clé, il peut interroger l'annuaire électronique du CNRS pour récupérer le certificat de Philippe. Ce certificat est signé par l'autorité de certification CNRS. Le poste de Nicole contenant en mémoire la clé publique de cette autorité de certification, le logiciel de messagerie peut vérifier la signature de ce certificat pour s'assurer que ce document a bien été créé par l'autorité de certification CNRS et n'a pas été modifié. Avec cette assurance, le logiciel de messagerie peut récupérer la clé publique de Philippe contenue dans ce certificat et l'utiliser avec confiance.

(NDLR : vous pouvez relire ce paragraphe plusieurs fois, faire une petite pause, éventuellement prendre un remontant …) .

Voici schématiquement la vérification du certificat de Philippe Cale avec extraction de la clé publique de ce dernier.

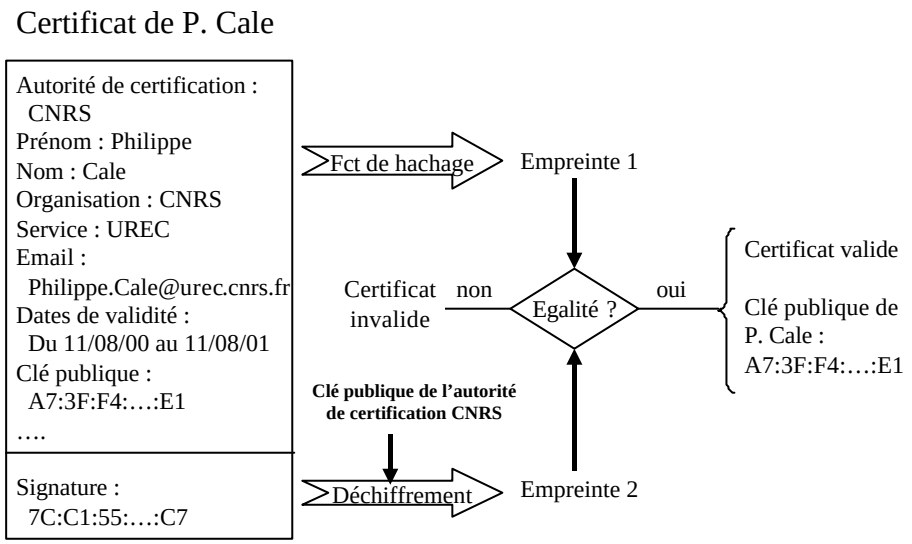

Figure 5 : vérification de certificat – récupération de la clé publique

Evidemment, les dates de validité du certificat sont aussi vérifiées avant de le déclarer valide.

3.4 Annuaires

Comme nous l'avons déjà montré, il est nécessaire de connaître le certificat de son correspondant (qui renferme sa clé publique), pour communiquer de manière sécurisée avec lui. Les outils informatiques sécurisés stockent les certificats qu'ils utilisent (voir au chapitre 6, l'exemple de Netscape). Ainsi ils n'ont besoin d'obtenir le certificat d'une personne qu'une seule fois. Ils intègrent aussi des mécanismes qui transmettent les certificats automatiquement. Si Philippe envoie un message signé à Nicole, son outil de messagerie sécurisé enverra en même temps le certificat de Philippe. L'outil de messagerie de Nicole le

récupérera avec le message envoyé et le stockera. Donc après quelques échanges les outils de chaque utilisateur auront les certificats des principaux correspondants des utilisateurs.

Néanmoins, on peut avoir besoin de communiquer avec des personnes avec lesquelles on n'a jamais échangé. Pour cela il faut un annuaire. Ce besoin d'annuaire est encore plus fondamental pour tous les serveurs réseaux (serveurs Web par exemple) qui voudront contrôler les certificats des utilisateurs qui les accèdent. Il est préférable qu'ils se référent à une même base d'informations à jour qui contienne les certificats, un annuaire, au lieu que chaque serveur gère sa propre base de données locale de certificats.

Le standard d'annuaire reconnu et implémenté par les principaux outils est maintenant LDAP, Light Directory Access Protocol [LDAP]. Il intègre en standard dans le format de ses enregistrements un champ destiné à contenir les certificats d'une personne.

Il y a donc tous les éléments nécessaires pour diffuser les certificats sur l'Internet. Le chapitre 4 reviendra sur la mise en place d'annuaires LDAP

3.5 Extension de ces mécanismes

Jusqu'à présent les mécanismes de sécurité ont été décrits en prenant la messagerie électronique entre personnes comme exemple. Mais ils peuvent s'appliquer et ils sont utilisés par beaucoup d'autres applications et d'autres objets.

Une application comme un service Web peut par exemple posséder un couple de clés et un certificat. Cette application présentera ce certificat à tous les utilisateurs qui y accéderont. Ceux-ci pourront ainsi être assurés qu'ils sont bien sur le bon serveur et la bonne application. Cela paraît superflu pour un serveur d'informations mais quand le serveur permet d'effectuer des transactions financières ou des achats, c'est obligatoire. La clé publique du service Web pourra aussi être utilisée pour chiffrer tous les échanges entre le client et le serveur. Pour ce faire, des mécanismes un peu différents de la signature, les applications HTTPS et SSL décrites au chapitre 5, seront mis en œuvre. Plus généralement toute application qui utilise le réseau (transfert de fichiers, accès interactif, accès à des bases de données, calcul distribué, …) pourra avoir un certificat et utiliser les mécanismes de sécurité.

D'autres objets peuvent aussi avoir des certificats. Ce sont les ordinateurs, les équipements réseau (routeurs …), les groupes de personnes, les listes de diffusion par exemple.

4. Autorité de certification et infrastructure de gestion de clés

Les mécanismes décrits ci-avant utilisent des certificats. Mais où peut-on les acquérir ? Qui les crée ? Avec quelles informations ? Comment sont-ils gérés ?

(NDLR : un problème est à peine résolu que de nouvelles questions arrivent. Y aura-t-il une fin ?)

4.1 Quelle autorité de certification ?

De même qu'une carte d'identité ou qu'un passeport, un certificat est délivré par une autorité, que l'on qualifie « de certification ».

Mais on est dans le monde de l'Internet où tout est électronique et planétaire, sans frontière ni gouvernement. Ainsi, techniquement, cette autorité peut être n'importe quelle association, société, organisme, individu, ... Actuellement il n'existe pas de gouvernement qui

délivre des certificats, ni d'organisations structurées et indépendantes comme celles qui affectent les numéros IP ou les noms de domaine. Par contre de nombreuses sociétés commerciales se sont déjà lancées dans ce commerce qui s'annonce très lucratif. Ce sont elles qui vendent les certificats.

La confiance que l'on accordera à un certificat va dépendre du sérieux de l'autorité qui l'aura délivré. De plus, on voit très bien le risque encouru par une entreprise ou un organisme dont la carte d'identité des employés aurait été créée par une autorité ni habilitée, ni contrôlée par elle-même. **Le choix de l'autorité de certification dans une organisation ou une entreprise est une décision stratégique**.

Le CNRS a ainsi décidé de créer sa propre autorité de certification qui pourra délivrer des certificats à toutes les personnes qui travaillent dans les unités CNRS [CA CNRS].

4.2 Quelle infrastructure de gestion de clés ?

Quelle que soit l'autorité de certification choisie, il faut faire d'autres choix. Comme il existe un circuit de procédures et de vérifications, des personnes habilités, … pour délivrer les cartes d'identité, il faut mettre l'équivalent en place. Il faut ainsi décider qui va recueillir et vérifier les informations données par une personne lorsqu'elle va demander un certificat, suivant quelles procédures, qui va créer le certificat, qui va le lui délivrer, pour quelle durée, où va-t-il être stocké, où va-t-on pouvoir récupérer les certificats d'autres personnes, … **Il faut définir ce que l'on appelle une architecture de gestion des certificats**. **IGC** (Infrastructure de Gestion de Clés [IGC]), et **PKI** (Public Key Infrastructure) sont les deux sigles les plus connus pour la désigner.

Les normes internationales décrivent les différents éléments fonctionnels d'une IGC. En simplifiant, l'architecture est constituée :

- D'**autorités d'enregistrement**.

Ce sont les mairies ou les préfectures, c'est à dire les guichets auxquels s'adressent les utilisateurs qui désirent obtenir un certificat. L'autorité d'enregistrement **vérifie l'identité du demandeur**, s'assure que celui-ci possède bien un couple de clés privée-publique et récupère la clé publique du demandeur. Elle **transmet ensuite ces informations** (informations d'identité du demandeur ainsi que sa clé publique) **à l'autorité de certification**. Une autorité d'enregistrement peut être un secrétariat avec une personne habilitée qui :

- vérifie une pièce d'identité présentée par le demandeur (action 1 dans la figure 6 ci-après)
- crée un couple de clés pour l'utilisateur (action 2 dans la figure 6). Ceci est réalisé avec un logiciel spécifique sur un ordinateur dédié et déconnecté du réseau (par prudence).
- remet une disquette à l'utilisateur qui contient la clé privée générée (action 3)
- garde la clé publique de l'utilisateur (action 3)
- transmet avec un message électronique signé une demande de certificat (contenant les informations d'identité et la clé publique du demandeur) à l'autorité de certification (action 4)

La transmission des demandes doit se faire de manière sécurisée, personne ne doit pouvoir modifier la demande durant le transport par exemple. Pour ce faire, les autorités d'enregistrement ainsi que l'autorité de certification ont des certificats et utilisent les mécanismes d'authentification, d'intégrité et de confidentialité pour communiquer entre eux.

Ceci n'est qu'un exemple de procédures, elles peuvent être très différentes selon l'organisation que l'on met en place.

• D'une **autorité de certification**.

Celle-ci reçoit les demandes de création de certificats venant des autorités d'enregistrement. Elle vérifie la validité de la signature des messages reçus, garantie de l'intégrité de la demande et de l'authentification des émetteurs. **Elle crée les certificats et signe ces certificats en utilisant sa clé privée.** Elle envoie les certificats aux utilisateurs et en parallèle les transmet au service de publication. Une autorité de certification a donc un couple de clé privée-publique pour signer les certificats. Si la clé publique est le plus largement connue possible, la clé privée est au contraire ultra confidentielle et doit être très bien protégée. Car si un tiers prend connaissance de cette clé, il pourra générer des faux certificats.

Le travail de l'autorité de certification peut être assuré par une personne habilitée d'un service central qui possède la clé privée de l'autorité de certification ou plutôt le mot de passe et la clé du coffre qui permet d'accéder et d'utiliser cette clé. La génération des certificats peut être réalisée par un logiciel spécifique sur un ordinateur portable entreposé dans un coffre fort. Sur cet ordinateur est aussi stockée la clé privée de l'autorité de certification. La personne habilitée « autorité de certification » reçoit les demandes de certificat par messagerie électronique sur un poste connecté au réseau (action 4). A chaque demande, elle vérifie la signature électronique de l'autorité d'enregistrement et transfère les informations sur une disquette. Elle ouvre le coffre, démarre l'ordinateur portable, insère la disquette (action 5), génère le certificat avec une signature en utilisant la clé privée de l'autorité de certification et transfère le certificat créé sur la disquette (action 6). Elle remet l'ordinateur dans le coffre et referme ce dernier. Avec la disquette, sur le poste connecté au réseau, elle envoie par messagerie électronique le certificat à l'utilisateur (action 7) et au service de publication (action 8).

Là encore ce n'est qu'un exemple de procédures.

Le choix et la mise au point de ces procédures, dont la description est fastidieuse (NDLR : ça c'est bien vrai !) est néanmoins fondamental pour la fiabilité de l'ensemble. Plus que les techniques de chiffrement, ce sont ces procédures qui sont un des talons d'Achille des certificats. Car si ces procédures sont mal définies ou mal appliquées, un intrus pourra par exemple s'emparer de la clé privée de l'autorité de certification et générer des faux certificats qui feront écrouler l'ensemble. Un groupe de travail inter-ministériel a déjà établi des recommandations précises sur ces procédures qui devront être suivies par toutes les administrations.

• D'un **service de publication**.

Celui-ci rend disponible les certificats émis par l'autorité de certification (action 9). Il publie aussi la liste des certificats valides et des certificats révoqués (voir chapitre suivant). Concrètement ce service peut-être rendu par un annuaire électronique LDAP ou un serveur Web accessibles par l'Internet.

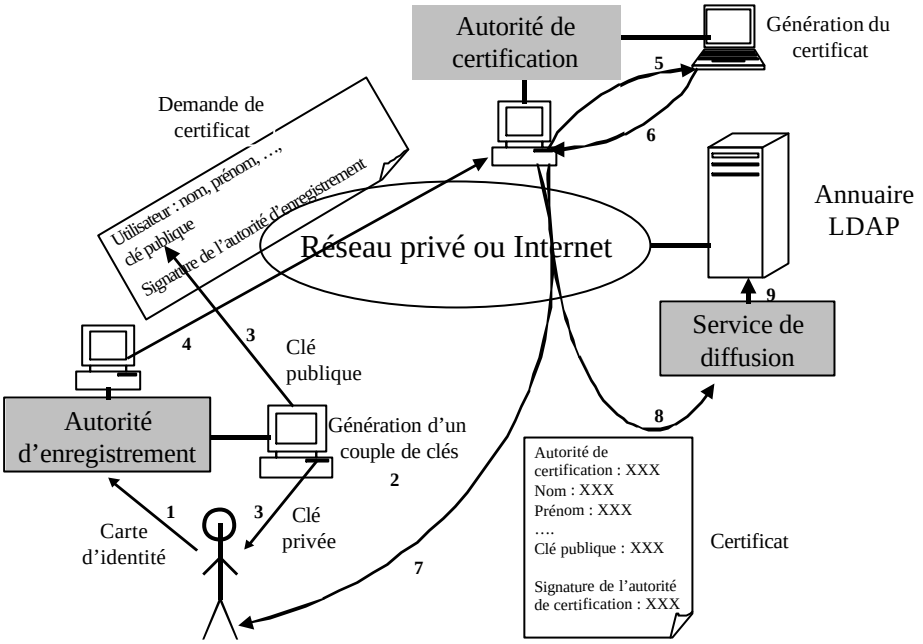

Figure 6 : exemple d'étapes pour la création d'un certificat
(NDLR : une autre petite pause pour comprendre ce schéma ?)

4.3 Durée de vie d'un certificat

Lorsque l'autorité de certification délivre un certificat, celui-ci contient sa date de création et une date de fin de validité. Généralement, comme de nombreuses cartes professionnelles, un certificat de personne dans une entreprise a une durée de vie fixe par défaut, un an par exemple. Si la personne est un stagiaire, un contractuel, un visiteur … cette durée peut correspondre exactement à la durée de sa présence dans l'entreprise.

Mais cette date n'est pas suffisante pour invalider un certificat dans certains cas. En effet, une personne peut quitter une entreprise ou changer de service ou se faire dérober sa clé secrète. Dans ce cas il faut invalider son certificat courant. La méthode est la même que pour les cartes bancaires. **Chaque autorité de certification publie régulièrement la liste des certificats révoqués**, qui ne sont plus valides. Cette liste est généralement dans un annuaire LDAP, accessible par le Web. Pour garantir son origine et son intégrité, elle est signée par l'autorité de certification. De même que les commerçants récupèrent régulièrement la liste des cartes bancaires non valables, il faut que les outils sécurisés chargent régulièrement, de manière automatique ou avec intervention de l'utilisateur, cette liste de certificats révoqués.

(NDLR : ce chapitre est parfaitement clair. Est-ce normal ?)

4.4 Quelques questions et options

Lorsque l'on décide de mettre en place une Infrastructure de Gestion de Clés comme c'est le cas au CNRS, il est nécessaire de répondre à de nombreuses questions fondamentales et prendre certaines options. Quelques exemples donneront une idée des problèmes concrets sur lesquels il faut se pencher.

• **Des certificats pour qui ?** Se limite-t-on aux personnes rémunérés par le CNRS, les agents ? Sachant que le personnel des laboratoires n'est jamais uniquement CNRS et que les autres employeurs comme les universités n'ont pas encore de plan pour mettre en place une IGC, ne serait ce pas trop réducteur ? Evidemment, il ne faut pas se limiter aux agents CNRS. Faut il des certificats par personne ou aussi par fonction ? Faut il plusieurs certificats par

personne ? Faut il aussi envisager des certificats pour les applications et les serveurs ? Oui, pour avoir des serveurs Web sécurisés. Mais ceci nécessite une autorité d'enregistrement spécifique, laquelle ? …

• **Quelle IGC mettre en place ?** Certainement une autorité d'enregistrement dans chaque délégation régionale, mais quel service de la délégation ? Faut-il une seule autorité de certification pour le CNRS ou plusieurs ? Faut-il délivrer des certificats à la demande ou d'office à tout le personnel ?

• **Quelles informations optionnelles doit contenir le certificat ?** Se limiter au nom du laboratoire ou mettre le maximum d'informations : département scientifique, délégation, fonction, … ?

• **Quelle durée de validité** pour un certificat ? Peut-être un an pour le personnel permanent, pour le personnel temporaire (CDD, thésards, stagiaires, …) se limiter à la fin de leur contrat ?

• **Comment gérer la liste des certificats révoqués** ? Où mettre cette liste ? Dans un annuaire LDAP ? Avec quelle périodicité la mettre à jour ? Journellement ? Mais quelle durée de vie pour cette liste ? Comment les utilisateurs la récupère-t-elle ? Avec quelle périodicité ?

• **Assure-t-on un séquestre ou une sauvegarde des clés privées ?** En effet, si un utilisateur perd sa clé privée (elle est stockée dans un fichier qui donc peut être détruit accidentellement), il ne peut plus déchiffrer les documents qu'il avait chiffrés précédemment. Il ne peut donc plus accéder à certains de ses propres documents. De plus une nouvelle loi demande de pouvoir fournir en cas d'enquête les clés privées permettant de déchiffrer certains documents. Pour ces 2 raisons il faut envisager de sauvegarder les clés privées de tous les utilisateurs. Si c'est le cas, comment assure-t-on ce service ?

• **Des clés pour faire quoi ?** En effet les experts recommandent d'utiliser au moins 2 clés différentes : une pour signer (qui n'a pas besoin d'être sauvegardée) et une pour chiffrer (qui a le besoin inverse). Génère-t-on un seul type de clé au départ avec une seule fonction (la signature par exemple) ? Ou les 2 types de clé ? Ou une clé multi usages ?

• **On a besoin d'un annuaire LDAP.** Comment le mettre en œuvre ? Doit il être décentralisé ou centralisé ? Quelles informations doit-il contenir ?

• **Quelles recommandations ou contraintes imposer aux utilisateurs pour qu'ils protègent leur clé privée** (ne la perdent pas, ne la divulgue pas, la protège contre le vol, …) **?** Faut-il mettre en œuvre des sauvegardes automatiques (ou séquestre) de ces clés et des certificats ? Faut-il imposer que cette clé et le certificat soient sur une carte à puce ?

(NDLR : que de questions !)

Rassurez vous, expliciter les réponses à chacune de ces questions n'est pas le sujet de cet article. Cette liste d'ailleurs très incomplète n'était destinée qu'à vous montrer que si l'utilisation d'un certificat paraît simple et ses potentialités très prometteuses, **la mise en place de toute l'architecture est un projet d'envergure**. De plus, le domaine est encore émergent, l'expérience est rare et parcellaire. Mais tout ceci est le problème des personnes chargées de définir l'IGC, et est totalement transparent pour l'utilisateur.

5. Quelques applications et standards

Regardons maintenant comment les mécanismes de sécurité décrits dans le chapitre 3 sont implémentés dans des applications courantes :

- S/MIME pour la messagerie électronique,
- HTTPS et SSL pour les communications Web,
- SSH et SSF pour des applications interactives,
- IPSec pour les communications entre équipements (de réseau ou stations).

5.1 S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions) **est un standard de messagerie électronique** [S/MIME], implémenté entre autres par les outils de messagerie Netscape Messenger et Outlook (Internet Explorer).

Un message électronique non sécurisé est transporté sur l'Internet sous la forme suivante (que nous avons simplifiée pour une meilleure compréhension) :

From : Philippe.Cale@urec.cnrs.fr
To : Serge.Montau@cru.fr
Date : 18 septembre 2000
Subject : CR réunion Paris
****
Je te joins le compte-rendu de la dernière réunion.
Philippe
****
Type de fichier : Word
Nom du fichier : CR.doc

<Fichier Word contenant le compte-rendu>
****

Ce message est envoyé par Philippe Cale à Serge Montau le 18/09/00. Il a pour sujet « CR réunion Paris ». Il contient 2 lignes de message ainsi qu'un fichier (attaché) Word, appelé CR.doc. Les étoiles sont des caractères de séparation des différentes parties du message.

S/MIME permet de signer un message.

Avec une signature, le message transporté prend alors la forme :

From : Philippe.Cale@urec.cnrs.fr
To : Serge.Montau@cru.fr
Date : 18 septembre 2000
Subject : CR réunion Paris
Type de message : Signé au format PKCS7
****
Je te joins le compte-rendu de la dernière réunion.
Philippe
****
Type de fichier : Word
Nom du fichier : CR.doc

<Fichier Word contenant le compte-rendu>
****
Signature :
MIIFsgYJKoZIhvc…
…
…JTMQsCQYqsdQ
****

Les outils de messagerie effectuent les opérations de création et de vérification de signature décrites dans un paragraphe précédent sur la signature.

Avant d'envoyer le message l'outil de messagerie ajoute 2 informations supplémentaires par rapport au précédent :

. « Type de message » est un champ qui indique que le message est signé, avec un format de présentation PKCS7 [PKCS7], jargon qui fait partie du standard S/MIME.

. « Signature : … » donne la valeur de la signature : MIIF…sdQ. Pour calculer cette signature, sur le poste de Philippe Cale, une fonction de hachage a été appliquée sur le texte du message « Je te joins … Philippe » ainsi que sur le contenu du fichier Word. Le résultat a été ensuite chiffré avec la clé privée de Philippe Cale, pour donner la signature.

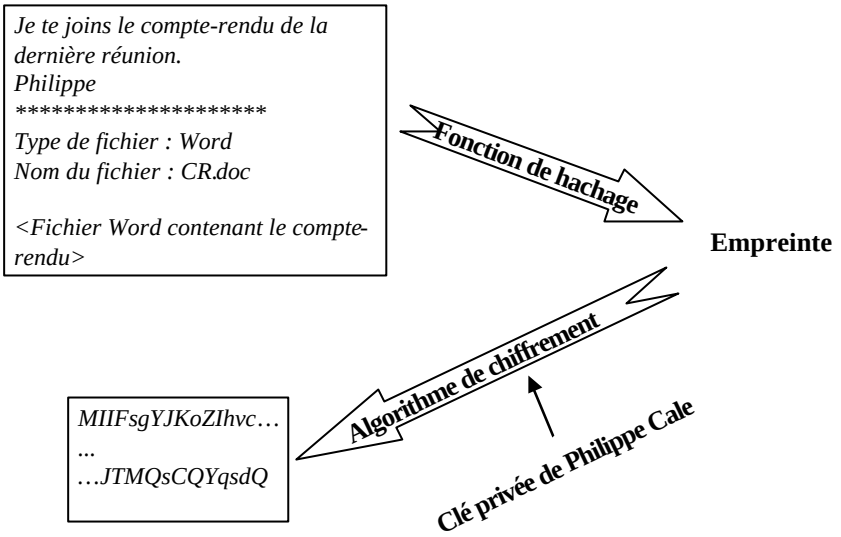

Figure 7 : génération de signature S/MIME

A l'arrivée le traitement suivant est effectué :

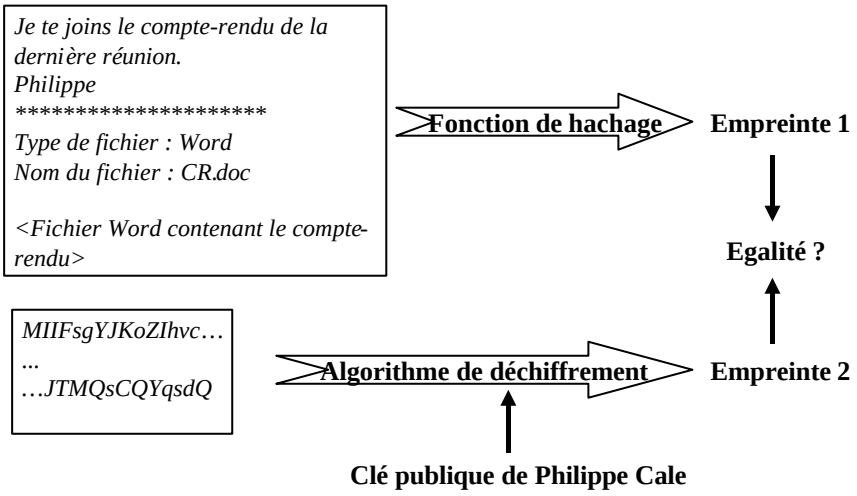

Figure 8 : vérification de signature S/MIME

Si l'empreinte 1 et identique à l'empreinte 2, cela prouve que le texte du message et le fichier Word n'ont été modifiés durant le transport (**garantie de l'intégrité**) et que la signature a bien été chiffrée au départ avec la clé privée de Philippe Cale (**authentification de l'émetteur)**.

On peut noter que S/MIME a été conçu de manière à ce que le message signé conserve les entêtes standards de messagerie non sécurisée From, To, … Ceci assure la compatibilité avec les outils qui «ne comprennent pas» S/MIME. Reçu par de tels outils le message pourra néanmoins être lu, mais sans garantie sur l'identité de l'émetteur ni sur l'intégrité du contenu.

S/MIME permet aussi de chiffrer un message.

Le même message d'exemple chiffré prend la forme (simplifiée ici) :

From : Philippe.Cale@urec.cnrs.fr
To : Serge.Montau@cru.fr
Date : 18 septembre 2000
Subject : CR réunion Paris
Type de message : chiffré au format PKCS7

MIAGCSqGSIb3DQEHA6CAMIACA …
…
… g+h7gBDhCfCAAAAAAAAAAAAAA=

« Type de message : chiffré au format PKCS7 » indique que ce message est chiffré sous un certain format.

MIA…AA= est le résultat du chiffrement du texte du message et du fichier Word avec la clé publique du destinataire Serge Montau. Uniquement celui-ci pourra ainsi déchiffrer ce message, avec sa clé privée, garantie de la confidentialité.

Au départ, l'outil de messagerie de Philippe Cale effectue le traitement suivant :

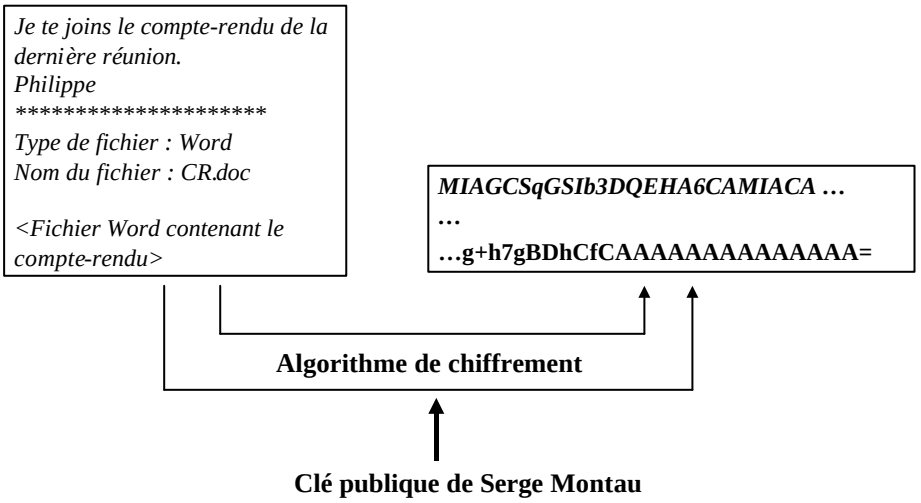

Figure 9 : chiffrement S/MIME

A l'arrivée l'outil de messagerie de Serge Montau déchiffre le message avec sa clé privée :

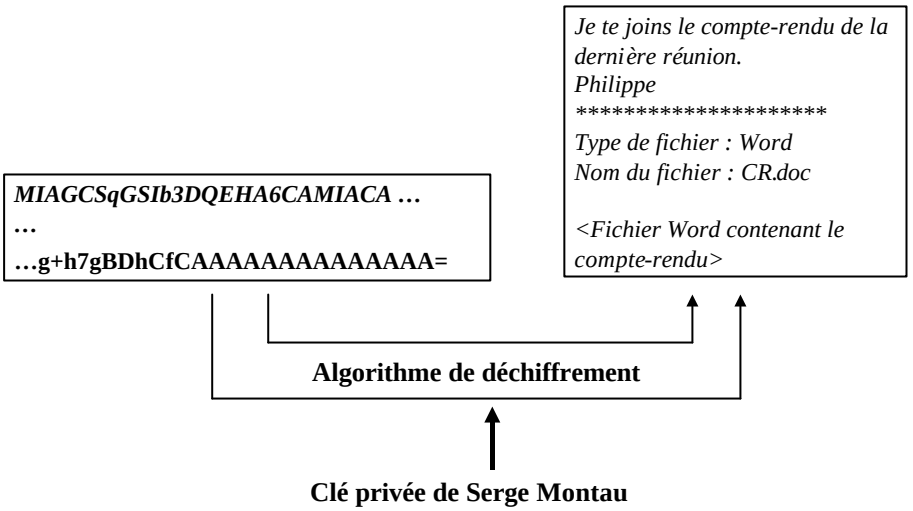

Figure 10 : déchiffrement S/MIME

S/MIME permet de combiner signature et chiffrement, c'est à dire d'envoyer un message signé puis chiffré, les opérations se déroulant dans ce sens. Le logiciel émetteur effectue les 2 opérations représentées figure 7 (génération de signature) puis figure 9 (chiffrement). Le logiciel récepteur suit la chronologie : figure 10 (déchiffrement) puis figure 8 (vérification de signature).

Par ces mécanismes S/MIME permet d'authentifier l'émetteur d'un message électronique et garantit la confidentialité et l'intégrité du message. Pour ce faire **il faut que les 2 correspondants possèdent chacun leur couple de clé publique-privée ainsi que la clé publique de l'autre, c'est à dire le certificat de l'autre**.

5.2 SSL et HTTPS

SSL (Secure Socket Layer) est un protocole développé par Netscape. La version 3 est en cours de standardisation par l'IETF [SSL]. **Dans une application en mode client-serveur,** en utilisant les certificats, **il permet d'authentifier les extrémités et d'assurer la confidentialité et l'intégrité des échanges de données**. Conceptuellement, il s'insère entre l'application (http, telnet, …) et les couches logicielles réseau (TCP pour être précis). Lorsque la session est établie entre le client et le serveur, toutes les données qui transitent sur le réseau sont chiffrées et signées.

HTTPS [HTTPS] est l'implémentation de http (« le protocole Web ») avec SSL. Les pages Web sur un serveur HTTPS sont désignées avec la chaîne « https » à la place de « http » (exemple : https://www.services.cnrs.fr/csec/).

HTTPS avec SSL sont souvent utilisés en mode « dissymétrique », où uniquement le serveur possède un certificat, le client n'en possédant pas. Ce certificat du serveur permet de chiffrer (avec une clé de session) toutes les données transmises dans les 2 sens et d'authentifier le serveur. Pour authentifier le client, le serveur demandera classiquement un mot de passe ou un numéro de carte bancaire ou … au client. **L'avantage est que ces**

données d'authentification ne circuleront pas en clair sur le réseau, tous les échanges étant chiffrés. C'est ainsi que fonctionne un **serveur Web sécurisé**, celui de votre banque peut-être.

Généralement, lorsqu'un client avec son navigateur se connecte sur un serveur Web sécurisé, ce dernier lui envoie son certificat (pour fournir sa clé publique …). Si ce certificat a été délivré par une autorité de certification reconnue par le navigateur du client, il est accepté de manière transparente pour l'utilisateur; sinon le navigateur demande à l'utilisateur s'il accepte ce type de certificat.

HTTPS et SSL peuvent aussi fonctionner de manière symétrique où à la fois le serveur et le client ont un certificat. Dans ce cas, l'authentification du client pourra se faire avec le certificat de celui-ci, sans mécanisme supplémentaire de mot de passe … Cela sera donc plus simple.

5.3 SSH et SSF

SSH (Secure Shell) [SSH] est un ensemble d'outils qui permettent d'avoir entre autres des sessions interactives en mode telnet ou en mode X, des transferts de fichiers et des exécutions de commandes à distance, avec authentification forte de l'utilisateur et du serveur et chiffrement des données transmises. Concrètement il remplace les commandes Unix rlogin, rcp, rsh et permet d'établir des sessions X chiffrées. L'intérêt premier est de se protéger contre l'écoute pirate des mots de passe circulant en clair sur les réseaux lorsque l'on utilise les commandes telnet, ftp, …depuis des sites extérieurs.

SSH utilise les algorithmes de chiffrement asymétrique avec une clé de session. Chaque utilisateur génère sur sa station, un couple de clés publique-privée. Il transmet sa clé publique au serveur sur lequel il accède à distance. Ensuite SSH utilise les mécanismes classiques décrits au chapitre 3 pour assurer authentification, intégrité et confidentialité des échanges. **SSH n'utilise donc pas actuellement les certificats**. Mais conceptuellement, il y aurait peu de développement pour ajouter cette fonctionnalité.

SSF (Secure Shell Français) [SSF] est une adaptation de SSH écrite par B. Perrot de l'IN2P3, destinée à l'usage sur le territoire français en conformité avec la législation française.

5.4 IPSec

Sur le réseau Internet et sur les réseaux informatiques en général, le flot des données est transporté en paquets de quelques centaines de caractères. Outre les données, chaque paquet contient des informations comme les adresses Internet (IP) de la machine origine et destinataire. Toutes ces données peuvent être lues ou modifiées par des pirates en écoute sur les lignes du réseau. IPSec **(IP Security)** permet, avec des algorithmes de chiffrement et des clés, de chiffrer le contenu de ces paquets et d'authentifier les deux éléments physiques qui dialoguent (routeur ou station, non pas les utilisateurs). C'**est un protocole défini par l'IETF** [Docs IPSec]. IPSec **peut être mis en oeuvre entre 2 équipements du réseau (routeur ou station).** Il est totalement transparent pour les utilisateurs, ce sont les administrateurs des réseaux et des stations qui le mettent en place et l'administrent.

Les algorithmes de chiffrement et les mécanismes utilisés peuvent être divers dans IPSec. Ils sont négociés au début de la connexion entre les équipements. Mais les algorithmes asymétriques, les fonctions de hachages, les couples de clés publique-privée et les certificats constituent le cœur du système. Avec IPSec, **ce sont les équipements qui possèdent des certificats**.

5.5 Différences entre ces applications

Les différentes applications et standards présentés précédemment apportent les mêmes fonctions de sécurité : authentification, intégrité, confidentialité. Ces fonctions n'étant pas utilisées par les mêmes objets, la sécurité engendrée ne touche pas les mêmes domaines.

IPSec par exemple apporte de la sécurité entre des éléments du réseau. S'il est utilisé entre 2 routeurs, toutes les communications, d'où qu'elles viennent, qui transiteront entre ces routeurs seront chiffrées donc confidentielles entre ces routeurs. Par contre, sur le réseau local, entre le routeur local et la station de l'utilisateur, elles ne seront plus chiffrées (sauf si IPSec va jusqu'à la station). De plus, IPSec n'assurera pas l'authentification des utilisateurs ou des serveurs, il authentifiera uniquement les 2 routeurs (dans cet exemple).

Différemment, S/MIME entre 2 stations, authentifiera chaque utilisateur émetteur de chaque message mais sera exclusivement limité à la messagerie. Donc les autres applications (accès Web, telnet, …) entre les 2 stations ne bénéficieront d'aucun service de sécurité supplémentaire.

Ainsi, ces services ne sont pas concurrents mais complémentaires. Selon les besoins, on peut choisir une des différentes applications sécurisées présentées, souvent même plusieurs.

6. Exemple d'implémentation des certificats dans Netscape

De même que son concurrent Internet Explorer, Netscape, logiciel de navigation et de messagerie Internet, intègre en standard les certificats, S/MIME, HTTPS et SSL. Regardons rapidement à travers Netscape, comment cela se concrétise sur le poste de travail d'un utilisateur.

6.1 Fonctions de base

Certaines fonctions existent dans le logiciel Netscape. Elles permettent de :

- Générer un couple de clés privée-publique. Cette fonction est utilisée quand l'utilisateur crée lui-même son couple de clés lors de sa demande de certificat.
- Stocker la clé privée et verrouiller son accès avec un mot de passe local. Quand l'utilisateur voudra utiliser sa clé privée, il devra entrer ce mot de passe. On paramètre généralement Netscape pour n'avoir à entrer ce mot de passe qu'une fois par session Netscape. On pourra ensuite utiliser cette clé privée autant de fois que nécessaire pour la messagerie, HTTPS et SSL sans besoin de redonner ce mot de passe pendant la session.
- Exporter la clé privée et le certificat de l'utilisateur dans un fichier accessible avec un autre mot de passe local. Ce fichier devra alors être sauvegardé (sage précaution, car il ne faut surtout pas perdre sa clé privée !). Il pourra aussi être copié sur une disquette par exemple, pour être utilisé sur un autre poste, éventuellement avec Internet Explorer, ... Ainsi une même clé privée et le certificat associé peuvent être utilisés sur une autre machine et avec un autre logiciel.
- Récupérer sur un serveur Web le certificat d'une autorité de certification, ce qui permet de récupérer la clé publique de cette autorité. Ceci étant fait, Netscape fera confiance à cette autorité et acceptera tous les certificats valides signés par cette autorité.
- Récupérer à partir d'un serveur LDAP ou Web la liste des certificats révoqués pour une autorité de certification.

• Signer et chiffrer un message avant envoi. Dans Netscape Messenger, il suffit de sélectionner « signature » ou/et « chiffrement » à la création d'un message. On peut même choisir que tous les messages que l'on envoie soient signés, voire chiffrés par défaut. Evidemment, pour signer il faut posséder un certificat et une clé privée et pour chiffrer il faut avoir le certificat du destinataire (dans une table décrite ci-après).

• Vérifier la signature d'un message reçu. Cette opération est réalisée par l'application Netscape de manière transparente pour l'utilisateur. Le certificat de l'émetteur est recherché dans une des tables Netscape. L'application s'assure que celui-ci n'a pas été révoqué en consultant la liste de certificats révoqués de l'autorité, puis extrait la clé publique de cet utilisateur. Avec celle-ci l'application vérifie la signature. S'il y a échec, l'utilisateur est averti par une icône rouge « Signature invalide » lorsque le message est affiché.

• Déchiffrer un message reçu. Cette opération nécessite un accès à la clé privée de l'utilisateur. Il y aura donc éventuellement une demande d'entrer le mot de passe local. Si le déchiffrement se passe bien le message est affiché en clair. Il est à noter que le message reste stocké chiffré dans la boite aux lettres.

• Passer en mode HTTPS-SSL. Ce passage se fait « automatiquement » piloté par le serveur auquel on accède ou dès que l'URL commence par «https : ». Il se peut que ce mode entraîne l'utilisation de la clé privée. Dans ce cas il faudra éventuellement entrer le mot de passe local pour y accéder comme dans le cas de la messagerie.

6.2 Gestion de tables

Pour réaliser l'ensemble des fonctions précédentes, Netscape gère aussi un ensemble de tables :

• Des autorités de certification auxquelles l'utilisateur fait confiance. Cette table n'est pas qu'une liste de noms, c'est aussi une liste de certificats d'autorité de certification, donc de clés publiques de ces autorités. L'utilisateur peut ajouter ou supprimer certaines autorités dans cette liste. Pour chaque autorité, une liste de certificats révoqués peut aussi être stockée. Attention, par défaut, à l'installation de Netscape cette table contient déjà une liste copieuse d'autorités (que je vous conseille de supprimer).

• Des certificats d'utilisateurs récupérés à partir d'un annuaire LDAP ou dans des messages signés reçus précédemment.

• Des certificats de serveurs Web accédés en HTTPS-SSL.

• Des certificats de l'utilisateur. L'utilisateur peut avoir plusieurs certificats ; il peut choisir d'utiliser par défaut un certificat différent pour la messagerie et pour accéder à un site Web.

Les descriptions précédentes montrent que le logiciel Netscape dispose de toutes les fonctions nécessaires pour utiliser les certificats. Pour vérifier, il n'y a qu'à dérouler le menu « Security » de la partie navigateur. Ainsi, lorsque l'utilisateur a obtenu un certificat, récupérer le certificat de son autorité de certification ainsi que la liste des certificats révoqués, l'utilisation de la signature, du chiffrement, … est trivial.

(NDLR : en place de toute cette description, ou pour mieux la comprendre, utiliser un certificat avec Netscape sur son poste de travail est recommandé)

7. Comment combler les lacunes des applications réseau actuelles (décrites dans le chapitre 2) ?

Lorsqu'une infrastructure de gestion de clés sera en place au CNRS, chaque personnel disposera d'un certificat qui contiendra au moins son nom, prénom, laboratoire d'affectation, adresse électronique ainsi que sa clé publique. Associé à cette infrastructure sera parallèlement en place tout un système d'annuaires LDAP permettant de retrouver le numéro d'agent d'une personne, sa fonction, son département scientifique, sa branche d'activité professionnelle, sa délégation, … Tout le personnel du CNRS aura aussi mis à jour son navigateur avec le certificat de l'autorité de certification CNRS, pour faire confiance aux certificats émis par cette autorité.

Dans cette configuration, si l'on reprend les exemples du chapitre 2, dans l'ordre de ce chapitre :

La diffusion des notes officielles pourra se faire par messagerie électronique au standard S/MIME. Les messages seront signées électroniquement par l'émetteur (le Directeur Général, un Délégué, …). Le récepteur pourra vérifier automatiquement l'origine du message et son intégrité.

Les votes, notations, … qui demandent la fonction de confidentialité, pourront aussi se faire par messagerie électronique S/MIME avec la fonction de chiffrement.

Toutes les applications de gestion pourront baser leurs contrôles d'accès sur les certificats. Les utilisateurs n'auront plus qu'un seul mot de passe à connaître, celui permettant localement d'utiliser leur clé privée, les services informatiques n'auront plus à gérer des couples utilisateur-mot de passe. Les applications contrôleront le certificat des utilisateurs et avec les informations complémentaires contenues dans les annuaires, en déduiront les droits d'accès de ces utilisateurs.

On pourra créer des Intranet « logiques » dans les laboratoires, les délégations, au niveau de l'organisme … Il suffira de contrôler l'accès aux pages Web, non pas sur le numéro IP ou le nom de la station de l'utilisateur, mais sur son certificat en utilisant HTTPS et SSL. On pourra par exemple choisir d'ouvrir un ensemble de pages à tous les utilisateurs qui possèdent un certificat CNRS, ces pages seront en fait l'Intranet CNRS général. De très nombreuses autres combinaisons de contrôles d'accès seront possibles avec les informations contenues dans les certificats et les annuaires associés. On pourra par exemple autoriser l'accès à des pages uniquement aux directeurs de laboratoires d'un département scientifique. Il suffira que chaque directeur ait un certificat et que l'on dispose d'un annuaire avec la fonction et le département de chaque personne.

On pourra faire les mêmes types de contrôles mais pour accéder à des logiciels avec une licence organisme ou ministère, ou à des bases de données électroniques comme ELSEVIER.

L'utilisation de certificats pourra aussi éviter de transporter en clair sur le réseau les mots de passe lors d'accès à distance, en utilisant SSL.

Enfin pour les applications avec des ressources totalement distribuées, chaque élément pourra posséder un certificat (utilisateur, machine, disque, …) et tous les contrôles d'accès pourront reposer sur cette carte d'identité.

A quand toutes ces applications ? Pas tout de suite. Tout ceci ne se mettra pas en place sans efforts. Il faudra effectuer certains développements logiciels, peut-être acheter des logiciels, mettre en place de nouvelles procédures, changer les anciennes, prendre de nouvelles habitudes, … Cela prendra du temps avant de se généraliser.

(NDLR : tout a un prix)

Mais si l'on dispose déjà d'une base solide de certificats, toutes ces applications pourront progressivement se développer en s'appuyant sur ces éléments de confiance. De plus, l'informatique et les réseaux évoluant à la vitesse que l'on sait, de nouvelles applications vont rapidement arriver qui intégreront en standard les certificats.

Preuve que ce n'est pas une vision trop utopique, la législation est déjà prête dans ce domaine. En effet, une loi qui accepte la signature électronique comme une preuve au même titre que la signature manuelle a déjà été votée et le décret d'application est en préparation [Décret signature électronique].

(NDLR : pourtant les législateurs prennent leur temps et sont souvent «en retard » dans le domaine des nouvelles technologies)

8. Ce que ne résoudront pas les certificats

Comme ce document tente de l'expliquer, les certificats peuvent rendre de très nombreux services. Néanmoins, il faut émettre quelques réserves.

(NDLR : oui, après cet encensement des certificats, il faudrait peut-être mettre des bémols)

Dans les mécanismes des certificats tout n'est pas résolu. On peut citer comme exemples une lacune et un danger.

La lacune est technique. La révocation des certificats est basée sur une liste qu'il faut concrètement télécharger régulièrement. Ceci est contraignant et lourd. Des standards sont en cours d'élaboration pour accéder à cette liste dynamiquement et automatiquement mais ils ne sont pas encore implémentés dans Netscape ou Internet Explorer.

Le danger, beaucoup plus important, est le problème de la confiance. En effet, toute cette mécanique nécessite des procédures strictes et sérieuses (dans la gestion des certificats, …) pour assurer les garanties qui sont affichées. Mais s'il s'avère que ces procédures ne sont pas fiables, il y aura des malversations, des faux certificats, … Si ces incidents sont trop nombreux, alors plus personne ne fera confiance aux certificats et ceux-ci n'auront plus aucune valeur. Ce sera la mort des certificats.

Ceci est d'autant plus préoccupant que tout ce secteur est totalement libéralisé, laissé aux entreprises privées. Or, celles-ci peuvent avoir tendance à négliger les procédures (coûteuses) pour un profit à court terme. C'est un peu comme si on laissait le soin aux multinationales de la grande distribution de délivrer les cartes d'identité et les passeports. Des certifications et des vérifications par des organismes gouvernementaux vont certainement se mettre en place dans certains pays, mais pas partout et ils tardent. Il n'y a par exemple pas encore d'autorité de certification gouvernementale française qui pourrait signer et certifier (après certains contrôles) celle du CNRS.

Regardons maintenant **du côté de la sécurité informatique,** telle qu'on la pratique aujourd'hui. Dans ce domaine, **les certificats ne vont pas résoudre tous les problèmes**. En étant provocateur, on peut même se demander, s'ils amélioreront réellement la sécurité actuelle.

Ce n'est pas une conséquence automatique. **Les certificats sont de très bons outils mais ce ne sont que des outils**. Outre le problème des IGC mal conçues et des certificats mal gérés, il reste le problème de l'utilisation. Car mal protégée par les utilisateurs, la clé secrète (et le certificat associé) ne sera pas plus fiable qu'un mot de passe qui circule en clair sur le réseau ou qui est noté sous le clavier. En effet, l'ensemble va reposer sur la confidentialité de la clé privée de l'utilisateur. Si celui-ci la communique autour de lui (concrètement divulgue le mot de passe qui permet d'y accéder), celle-ci aura la même valeur qu'un mot de passe partagé par plusieurs personnes.

Mais on peut penser que si l'on explique aux utilisateurs que la clé privée sert non seulement à accéder à des applications mais aussi à prouver leur signature dans leurs messages qui auront un caractère officiel, avec preuve légale, … ils comprendront rapidement que c'est un secret encore plus important que le code pour utiliser leur carte bancaire. Imposer de stocker cette clé et le certificat sur une carte à puce avec un code d'accès peut aussi aider à mieux matérialiser ces éléments et ainsi faire acquérir les bons réflexes aux utilisateurs.

Mais finalement, ceci n'est pas le problème des certificats. C'est un problème classique de sécurité : **la sensibilisation des utilisateurs. Elle devra être faite dès les premières utilisations de certificats**.

Les certificats ne résoudront pas non plus les problèmes d'intrusions ou de rupture de service par l'utilisation de failles de sécurité dans les logiciels, la prolifération de virus en tous genres, … Il faudra toujours des architectures sécurisées, des contrôles d'accès dans ces architectures, des anti-virus, …

On peut prévoir aussi que les certificats comme tout nouveau service informatique apporteront leurs propres problèmes de sécurité.

Mais ne noircissons pas trop le futur. **Si l'introduction des certificats dans un organisme ou une entreprise est correctement réalisée, avec professionnalisme et méthode, le gain en terme de sécurité et de nouvelles facilités de contrôles devrait être très important**.

On peut penser aussi que **plus que l'amélioration de l'existant en terme de sécurité, l'intérêt des certificats réside dans les nouveaux services qu'il sera beaucoup plus facile de mettre en place**, en particulier dans des structures très éclatées géographiquement comme le CNRS.

(NDLR : merci à Marie-Agnès, Marie-Claude, Nicole et Sophie pour leur relecture et leurs commentaires)

9. Références

- [AES] Advanced Encryption Standard : The block cipher Rijndael
  http://www.esat.kuleuven.ac.be/~rijmen/rijndael/index.html
- [Articles UREC] quelques articles sur le sujet des certificats
  http://www.urec.cnrs.fr/securite/certifications.html.

- [CA CNRS]
  http://www.dsi.cnrs.fr/bo/2000/08-09-00/416-bo080900-dec000381bpc.htm
- [CNRS] Centre National de la Recherche Scientifique
  http://www.cnrs.fr
- [Datagrid]
  http://grid-france.in2p3.fr/
- [DES] Data Encryption Standard
  http://www.itl.nist.gov/fipspubs/fip46-2.htm
- [Décrets chiffrement] : Décrets 99-199, 99-200
  http://www.internet.gouv.fr/francais/textesref/cryptodecret99199.htm
  http://www.internet.gouv.fr/francais/textesref/cryptodecret99200.htm
- [Décret signature électronique]
  http://www.internet.gouv.fr/francais/textesref/pagsi2/signelect-projdecret/sommaire.htm
- [Glossary] RFC2828 : Internet Security Glossary
  http://www.pasteur.fr/infosci/RFC/28xx/2828
- [HTTPS] RFC2817 : Upgrading to TLS Within HTTP/1.1
  http://www.pasteur.fr/cgi-bin/mfs/01/28xx/2817
- [IGC]
  http://www.scssi.gouv.fr/document/igc.html
- [IPSec] ensemble de documents sur IPSec
  http://www.hsc.fr/ressources/veille/ipsec/index.html.fr
- [jla]
  http://www.urec.cnrs.fr/jla
- [LDAP] : un ensemble de références
  http://www.cru.fr/ldap/
- [MD5] RFC1321 The MD5 Message-Digest Algorithm
  http://www.pasteur.fr/cgi-bin/mfs/01/13xx/1321
- [PKCS7] RFC2315 : PKCS #7: Cryptographic Message Syntax Version 1.5
  http://www.pasteur.fr/infosci/RFC/23xx/2315
- [Renater] Réseau NAtional de la Technologie de l'Enseignement et de la Recherche
  http://www.renater.fr
- [RSA]
  http://rsasecurity.com/rsalabs/
- [SHA] Secure Hash Standard
  http://www.itl.nist.gov/fipspubs/fip180-1.htm
- [S/MIME] S/MIME Working Group
  http://www.imc.org/ietf-smime/
- [SSF] serveur de documentations et de programmes sur SSH et SSF
  http://ccweb.in2p3.fr/securite/ssf/
- [SSH] page d'accueil pour la communauté des utilisateurs de SSH
  http://www.ssh.org/
- [SSL] Introduction to SSL
  http://developer.netscape.com/docs/manuals/security/sslin/index
- [Triple DES] RFC 1851: The ESP Triple DES Transform
  http://www.pasteur.fr/infosci/RFC/18xx/1851
- [X509V3] RFC2459 : Internet X.509 Public Key Infrastructure Certificate and CRL Profile
  http://www.pasteur.fr/cgi-bin/mfs/01/24xx/2459