La signature électronique, contexte, applications et mise en oeuvre.

Jean-Luc Parouty, Roland Dirlewanger, Dominique Vaufreydaz

La signature électronique, contexte, applications et mise en œuvre.

Jean-Luc Parouty
INRIA / Direction des Réseaux et des Systèmes d'Information (DRSI)
655 Avenue de l'Europe - 38330 Montbonnot – France
Jean-Luc.Parouty@inria.fr

Roland Dirlewanger
CNRS - Délégation Aquitaine et Poitou-Charentes (DR15)
Esplanade des Arts et Métiers - 33402 Talence cedex
rd@dr15.cnrs.fr

Dominique Vaufreydaz
INRIA Rhône-Alpes / projet PRIMA
655 Avenue de l'Europe - 38330 Montbonnot – France
Dominique.Vaufreydaz@inrialpes.fr

Résumé

Depuis toujours, le document papier est notre support privilégié dés lors qu'il nous est nécessaire de conserver le témoignage d'un accord entre plusieurs parties. Traditionnellement, et à défaut de pouvoir en protéger l'intégrité, l'usage de sceaux ou de signatures, permet de garantir l'authenticité de tels documents. Avec l'utilisation croissante des outils de communication « immatériels », que sont le téléphone, le fax ou encore l'Internet, le problème de la protection de nos échanges est devenu particulièrement critique.
Les progrès conjugués des mathématiques et de l'informatique ont permis, depuis les années 1970, de disposer progressivement d'un panel complet de solutions algorithmiques et de standards adaptés à la certification de nos documents électroniques.
Avec la criticité croissante de nos échanges et la disponibilité de solutions techniques avérées, la mise en place progressive d'un cadre juridique adapté est venu compléter l'ensemble.
Après un bref rappel des techniques et un tour d'horizon du contexte juridique, nous nous intéresserons aux différents outils disponibles et à leurs limites, puis nous présenteront brièvement les travaux réalisés autour d'un prototype d'applet signeuse : Sign@tor.

Mots clefs

Signature électronique, Parapheur électronique, Certificats, Signature simple, Signature avancée, Contractualisation, Niveaux de contractualisation, Signator, Formulaire HTML, XML Signature, XAdES.

1. Rappels / Définition

1.1 Principe de la signature électronique

La signature électronique repose sur deux familles d'algorithmes, qui seront utilisés de manière complémentaire :
- des algorithmes de chiffrement dit « asymétriques » ou à « clef publique ».
- des fonctions de hachages

1.1.1 Algorithmes asymétriques

Le concept d'algorithmes à clef asymétrique a été présenté pour la première fois en 1976 par Whitfield Diffie et Martin Hellman[1].
Il faudra néanmoins attendre 1978, pour qu'un tel système soit présenté, par Ronald Rivest, Adi Samir et Leonard Adelman[2]. Ainsi naquit l'algorithme RSA[3].
Le principe d'un algorithme asymétrique est relativement simple. Un couple de clefs numériques est construit de manière à ce que le cryptogramme généré à partir d'un texte clair et l'une des clefs ne puisse être aisément retrouvé qu'avec l'autre clef.

[1] Whitfield Diffie et Martin Hellman, *National Computer Conference*, 1976.
[2] Rivest, R. L., Shamir, A., Adelman, L. A.: *A method for obtaining digital signatures and public-key cryptosystems*; Communications of the ACM, Vol.21, Nr.2, 1978, S.120-126
[3] RSA est constitué des initiales des trois auteurs Rivest, Adelman et Shamir.

En pratique, l'une des clefs est conservée secrètement (clef privée), tandis que l'autre est diffusée publiquement (clef publique).
Seul le propriétaire d'une clef privée (Alice dans notre exemple) aura pu chiffrer un texte déchiffré avec la clef publique associée (figure 1).
L'authenticité du document peut être ainsi garantie.

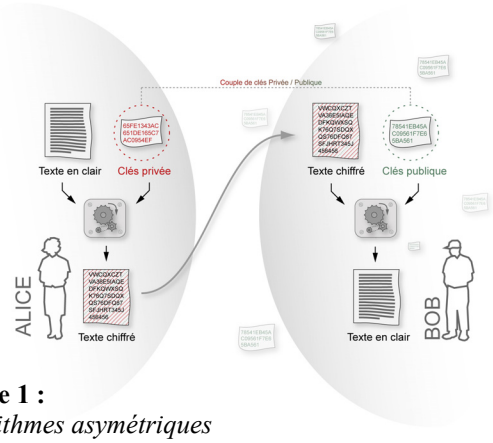

Figure 1 :
Algorithmes asymétriques

1.1.2 Fonction de hachage

Les fonctions de hachages sont des fonctions à sens unique et « sans collision », générant une sortie de taille fixe (appelée condensat ou empreinte), caractéristique des données fournies en entrée.

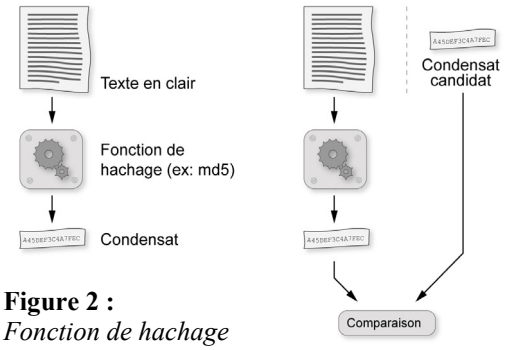

Figure 2 :
Fonction de hachage

Ces fonctions sont dites à sens unique car il est impossible de retrouver les données initiales à partir de l'empreinte.

Une fonction est dite « sans collision » ou « injective » lorsqu'il est réputé très difficile de trouver deux sources différentes conduisant à un même résultat.

Le calcul du condensat d'un document et la comparaison de celui-ci avec sa valeur initiale permet de contrôler l'intégrité d'un document (figure 2).

1.1.3 Construction et vérification d'une signature

La signature électronique va faire appel à ces deux familles d'algorithmes, afin de pouvoir garantir l'authenticité et l'intégrité d'un document (figure 3).
Les algorithmes asymétriques usuellement utilisés sont RSA et DSA[4], les fonctions de hachages les plus courantes dont MD5[5] et SHA[6].

1.2 Limite du modèle - Problématique de la confiance dans la clef

Le processus de vérification de la signature repose totalement sur la confiance qu'a le vérificateur dans la clé publique de l'émetteur. Dans l'exemple précédent, l'attaque classique consiste à transmettre à Bob par un moyen quelconque la clé publique d'un tiers et de le convaincre qu'il s'agit de celle d'Alice. Tout message signé avec la clé privée correspondante sera considéré par Bob comme étant signé par Alice.

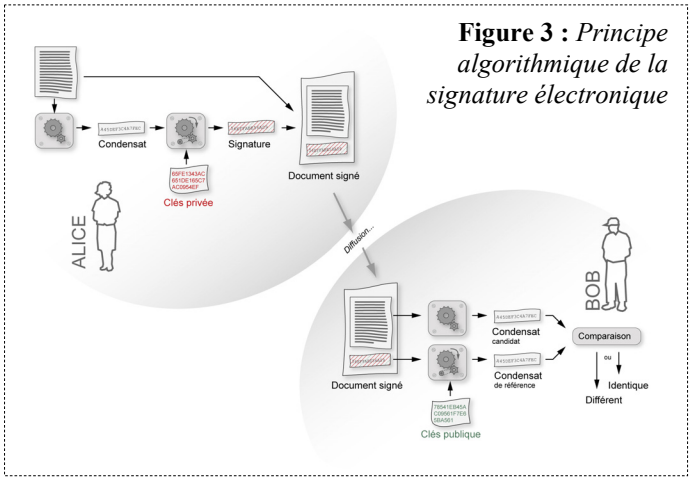

Figure 3 : *Principe algorithmique de la signature électronique*

La transmission à Bob de la clé publique d'Alice doit donc s'effectuer par un moyen sûr. Pourquoi pas de la main à la main ? C'est la technique de l'anneau de confiance (*public key ring*) qui est utilisé dans PGP[7] : Alice signe un certain nombre de clés publiques dont elle peut certifier le titulaire. Elle transmet à Bob sa clé publique, par un moyen sûr, et la liste des clés publiques qu'elle a signées, par un moyen quelconque. Bob peut

[4] DIGITAL SIGNATURE STANDARD (DSS), Federal Information Processing Standards, Publication 186, 1994 May 19, http://www.itl.nist.gov/fipspubs/fip186.htm
[5] R.L. Rivest, RFC 1321: *The MD5 Message-Digest Algorithm*, Internet Activities Board, 1992, http://www.ietf.org/rfc/rfc1321.txt?number=1321
[6] SECURE HASH STANDARD, Federal Information Processing Standards, Publication 180-1, 1995 April 17, http://www.itl.nist.gov/fipspubs/fip180-1.htm
[7] P.R. Zimmermann, *The Official PGP User's Guide*, Boston, MIT Press, 1995

alors choisir d'accorder sa confiance dans toutes ces clés, et, éventuellement, dans toutes les clés publiques signées par les titulaires de celles-ci. Ce processus peut, en quelques échanges, certifier un grand nombre de clés. Toutefois, si la population devient très importante ou si les liens entre les utilisateurs ne permettent pas de faire des échanges sûrs, il paraît difficile de garantir qu'à tout moment un utilisateur dispose dans son anneau de confiance de toutes les clés publiques dont il a besoin.

La solution consiste à faire signer une information sur l'identité ainsi que la clé publique de chaque utilisateur par une autorité en laquelle tous les partenaires ont confiance. Chaque utilisateur n'a besoin de récupérer de façon sûre qu'une seule clé publique, celle de l'autorité. Grâce à cette clé, il peut valider les clés publiques de tous les utilisateurs.

C'est cette idée qui est à la base des certificats X.509 et des infrastructures de gestions de clés (IGC) : un certificat contient des informations sur son titulaire (nom, prénom, adresse électronique, organisme, …), des informations sur l'autorité qui a signé le certificat, une date de début et une date de fin de validité, la clé publique du titulaire, les usages autorisés pour ce certificat, etc. Le tout est signé par la clé privée de l'autorité qui a émis le certificat. Cette dernière est appelée autorité de certification. Elle dispose elle même d'un certificat, soit auto-signé, soit signé par une autorité de niveau supérieur.

Afin de faciliter la vérification des signatures, les documents ou messages signés contiennent le certificat du signataire. Ainsi, pour vérifier la signature des messages reçus d'Alice, Bob effectue les opérations suivantes :

- il extrait le certificat d'Alice du document ou du message
- il vérifie la signature du certificat en utilisant la clé publique contenue dans certificat de l'autorité de certification qui a émis le certificat d'Alice.
- il vérifie la validité du certificat (dates de validité, non révocation, etc...)
- il vérifie la signature du message en utilisant la clé publique contenue dans le certificat d'Alice.

En fin de compte, la confiance dans les clés publiques d'une population potentiellement grande d'utilisateurs est ramenée à la confiance dans un petit nombre de clés publiques d'autorités de certification.

1.3 Horodatage

Nombreuses sont les procédures administratives qui mettent en jeu la date d'envoi des documents : dépôt de candidatures, réponses à des appels d'offres, déclarations diverses, etc. Les mécanismes traditionnels utilisent le cachet de la poste.

La dématérialisation de telles procédures passe par un équivalent électronique à ce cachet de la poste. Le format PKCS#7 qui décrit les conteneurs de données signées dans la messagerie électronique inclut un champ pour indiquer la date à laquelle le message a été signé. C'est toutefois la date et l'heure du poste de travail du signataire qui sont utilisés dans ce champ. L'émetteur peut donc « tricher » en indiquant des dates fausses. De la même façon, un tiers mal intentionné ayant récupéré une clé privée associée à un certificat périmé peut ramener artificiellement la date de sa machine dans la période de validité du certificat et émettre des documents signés.

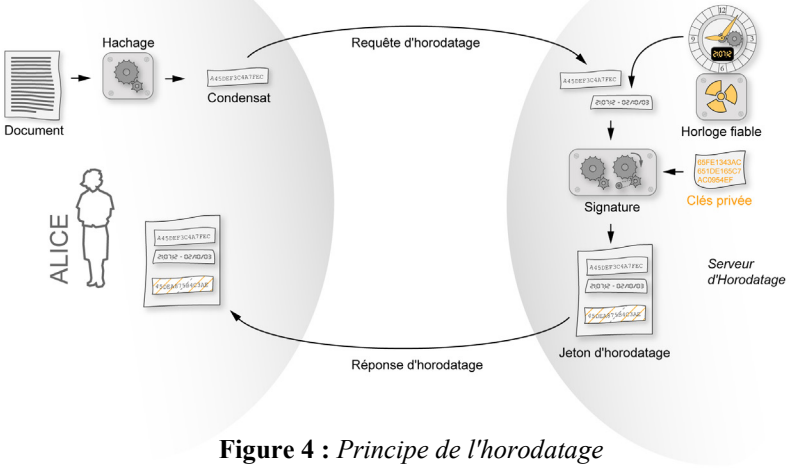

Figure 4 : *Principe de l'horodatage*

La signature électronique d'un document est donc indissociable d'un mécanisme qui garantit que le document existait à une date et une heure donnée et n'a pas été altéré depuis. C'est l'horodatage. Il consiste à transmettre à une autorité de confiance, appelée autorité d'horodatage, une requête comprenant le condensat du document à horodater. L'autorité renvoie au demandeur un jeton d'horodatage, le condensat, une date et une heure, le tout signé par l'autorité d'horodatage (figure 4).

1.4 Signature ou visa

Le titulaire d'un certificat X.509 peut authentifier des documents grâce à la signature électronique. Il peut également s'authentifier auprès de sites WWW qui utilisent le protocole HTTP sur SSL ou TLS.

Imaginons une application WWW authentifiée qui gère les congés dans un établissement. Bob remplit un formulaire WWW de demande de congés. L'application WWW notifie Alice, son supérieur hiérarchique. Elle accède au formulaire WWW qui affiche la demande de Bob ainsi que deux boutons « j'accepte cette demande » et « je refuse cette demande ».

Si cette application WWW est authentifiée par certificats, on peut admettre que seul Bob a pu transmettre la demande et seule Alice a pu l'accepter ou la refuser. Pourtant, il n'y a pas eu de signature électronique[8]. En pratique, ce mécanisme ne peut pas résister à un contentieux. En effet, si l'application s'appuie sur une base de données, les administrateurs de cette base ont la possibilité de rajouter ou modifier des enregistrements correspondant à la demande de congés de Bob.

Dans ce type d'applications, on parlera plutôt de visa électronique. Le visa électronique nécessite de la part de l'utilisateur qui l'appose un mécanisme d'authentification équivalent à celui de la signature électronique. Il peut toutefois être détourné par des personnes disposant de privilèges élevés sur l'application qui utilise ce visa. Dans des organisations où ces personnes sont réputées de bonne foi, on pourra accepter un visa électronique au même titre qu'une signature. Il faudra toutefois bien garder à l'esprit que le visa ne peut pas avoir la même valeur probante que la signature électronique.

1.5 Normes et standards

Il existe un nombre considérable de normes et de standards qui régissent les certificats, la signature électronique, l'identification des algorithmes cryptographiques, les messages signés, etc. Nous nous focaliserons dans ce paragraphe sur celles qui seront utiles dans la suite.

Avant tout, rappelons-nous que les certificats X.509 sont directement dérivés des travaux de l'ISO (*International Standardisation Organisation*) sur l'interconnexion de systèmes ouverts (OSI, *Open Systems Interconnection*). Au centre de ces travaux, il y a la représentation pour l'être humain d'une part, pour les machines d'autre part, de données structurées manipulées par les divers protocoles. Un langage évolué (ASN.1, *Abstract Syntax Notation One*) a été conçu pour décrire ces structures. Des mécanismes (BER, *Basic Encoding Rules*) ont été définis pour encoder de façon standard ces structures de données sur n'importe quelle plate-forme afin qu'elle puisse être décodées et reconstituées sur n'importe quelle autre.

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 807 (0x327)
    Signature Algorithm: md5WithRSAEncryption
    …
    … Description en clair du contenu d'un certificat.
    … Cette section n'a pas de syntaxe propre. Elle est
    … destinée à la lecture par des êtres humains.
    … On peut la considérer comme un commentaire
    …
-----BEGIN CERTIFICATE-----
MIIEYjCCA0qgAw (…)
Codage Base64 du certificat. Seule cette
section et les deux balises importent.
(… )JHGJYUY
-----END CERTIFICATE-----
```

Figure 5 : *Certificat X509 au format PEM*

Les mécanismes de codage de BER produisent des données binaires. Traditionnellement, les technologies de l'Internet préfèrent manipuler des chaînes de caractères ASCII organisées en lignes plus ou moins compréhensibles directement par un humain. Le compromis entre ces deux usages a priori orthogonaux a été d'utiliser le codage Base64 qui permet de transformer un document binaire en une suite de caractères ASCII. La partie Base64 est entourée d'une balise de début et d'une balise de fin caractéristique du contenu. Toute chaîne de caractères qui se trouve en dehors de ces balises est considérée comme du commentaire. Ce format est communément appelé le format PEM (*Privacy Enhanced Mail*), du nom des standards de l'Internet relatifs à l'authentification et à la confidentialité dans la messagerie électronique.
La figure 5 donne une idée de la représentation au format PEM d'un certificat X.509.

[8] En fait, si. Dans la phase d'authentification du protocole SSL ou TLS, le serveur demande au client de signer une suite d'octets dont il a généré aléatoirement une partie. Il s'assure ainsi que le client a accès à la clé privée du certificat de l'utilisateur.

Le format DER (*Distinguished Encoding Rules*) correspond aux données binaires contenues dans un fichier PEM. C'est donc le résultat du décodage Base64 des données contenues entre la balise de début et la balise de fin. Il faut un interprète ASN1 pour les afficher ou les manipuler, comme par exemple, « openssl[9] » ou « pp[10] ».

Les PKCS (*Public-Key Cryptography Standards*) définissent douze standards (de PKCS#1 à PKCS#12) pour spécifier l'utilisation d'algorithmes de chiffrement ou d'échanges de clés, les interfaces avec des modules cryptographiques, les certificats, les requêtes de certificats, les conteneurs de diverses natures. Par exemple, PKCS#6 décrit les certificats. C'est un sur-ensemble des certificats X.509.

Arrêtons-nous sur les deux PKCS les plus courants : le 7 et le 12. Le PKCS#7 décrit les conteneurs de données. Il peut s'agir de données signées, chiffrées, signées et chiffrées ou bien d'une liste de certificats d'autorités de certification, etc.

Le PKCS#12, également connu sous le nom de PFX, décrit les mécanismes qui permettent de garantir l'intégrité et la confidentialité de données comme des clés privées, des certificats d'utilisateurs ou d'autorités de certification. C'est le format utilisé pour sauvegarder ou pour transporter des certificats d'utilisateurs et les clés privées correspondantes. Les données contenues dans un fichier au format PKCS#12 sont protégées par un certain nombre de clés. L'une sert au contrôle d'intégrité des données, les autres servent à chiffrer/déchiffrer des données confidentielles comme les clés privées. En pratique, les outils usuels comme les navigateurs n'utilisent qu'une seule clé pour toutes ces opérations. Cette clé est définie par l'utilisateur au moment où il exporte son certificat et sa clé privée. Elle est demandée au moment où il importe son certificat.

Le format S/MIME[11] définit une collection de types MIME permettant d'envoyer et de recevoir des données signées, chiffrées ou les deux à la fois via la messagerie électronique. Les données en question peuvent être elles mêmes des parties MIME, comme dans le cas où on signe un message électronique contenant un document attaché. S/MIME décrit notamment les types « *multipart/signed* », « *application/pkcs7-signature* » et « *application/pkcs7-mime* ».

A ces standards « historiques » peuvent être ajouté plusieurs nouveaux standards mieux adaptés à nos besoins fonctionnels, en provenance notamment du monde XML.

XML Signature

Ce premier standard est le résultat du groupe de travail « *XML Signature WG* » qui est un groupe conjoint du W3C et de l'IETF. De ces travaux sont issus une « *Recommandation* » du W3C et une « *Standard track* » de l'IETF, « *XML-Signature Syntax and Processing*[12] » / *RFC3275*[13].

```
<?xml version="1.0" encoding="UTF-8" ?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
                <CanonicalizationMethod (…) />
                <SignatureMethod Algorithm=(…) />

                <Reference URI="#object">
                     <DigestMethod Algorithm=(…) />
                     <DigestValue>7/XT(…)Ysk=</DigestValue>
                </Reference>
  </SignedInfo>
  <SignatureValue>ov3HOo(…)3L4=</SignatureValue>
  <KeyInfo>
                <KeyValue>
                     <RSAKeyValue>
                               <Modulus>q07(…)WkArc=</Modulus>
                               <Exponent>A(…)AB</Exponent>
                     </RSAKeyValue>
                </KeyValue>
  </KeyInfo>
  <Object Id="object">some text</Object>
</Signature>
```

Figure 6 : *Signature XML*

XML Signature a été conçu pour permettre la signature des échanges XML. Pour cela, l'essentiel des fonctionnalités offertes par les signatures de type PKCS ont été reprises, avec un certain nombre d'apports propres à XML, dont le principal est sans doute la possibilité offerte de pouvoir ne signer qu'une partie de l'arbre XML, autrement dit d'un document.

La signature d'un document va consister à construire une liste de références (URI), puis de calculer pour chacune d'entres elles un condensat. Cette liste de condensat fera à son tour l'objet d'un hachage, qui sera signé. On pourra alors ajouter divers éléments, tels que le certificat du signataire.

Les références étant des URI, ces dernières

[9] http://www.openssl.org
[10] http://www.mozilla.org
[11] RFC 2311
[12] http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/
[13] http://www.ietf.org/rfc/rfc3275.txt

peuvent être internes ou externe au document, concerner tout ou partie de celui-ci, etc.
La figure 6 offre un exemple de signature XML. A noter, que la maîtrise d'ASN1 est ici facultative :-)

Plusieurs implémentations de *XML Signature* sont actuellement disponibles[14], on peut citer :
- **XMLsec**[15], diffusé sous licence MIT et qui peut appuyer sur les couches cryptographiques d'OpenSSL, Mozilla (NSS), GnuTLS ou Microsoft MSCryptoAPI. XMLsec propose une API C++.
- **Apache XML Security**[16], du groupe Apache, diffusé sous licence Apache Software Licence, qui propose une double API, Java et C++.

XML Advanced Electronic Signatures (XAdES)
XAdES est une extension de XML Signature. Les extensions concernent notamment le domaine de la non-répudiation, en définissant des formats XML pour les « *Signatures électroniques avancées* » susceptibles de rester valides pendant de grandes périodes, conformément à la « *Directive Européenne 1999/93/EC* ».
XAdES[17] est le résultat des travaux de la section STF 178 de l'ETSI[18].
Une « note W3C » reprend ces spécifications, en vue d'une recommandation W3C[19].

Concernant la mise en œuvre de XAdES, si les implémentations ne sont pas extrêmement nombreuses, il est néanmoins incontournable de citer l'initiative OpenXAdES, qui, comme le nom l'indique, est une initiative ouverte autour de XAdES.

Dans le but de favoriser leurs échanges, la Finlande et l'Estonie ont déployés une carte d'identité électronique commune[20], permettant l'authentification forte et la signature électrique. Accessoirement, c'est également une carte d'indenté reconnue dans 19 pays européens ;-) 30 € environ, valable 3 ans.

Afin d'être en conformité avec la « *Directive européenne 1999/93/CE* », un important travail à été entrepris au sein d'un projet d'architecture de signature électronique appelé « *DigiDoc* », qui s'est appuyé sur les spécifications de XAdES.

A ce jour, le projet *DigiDoc* offre (côté Estonie[21]) :
- Un client, permettant de vérifier une signature électronique et de signer un document avec sa carte d'identité électronique
- Un portail *DigiDoc*, offrant, outre les fonctions du client téléchargeable, la possibilité de signature multiple et collaborative (dépôt pour signature à 1 ou n signataires)
- Un format de signature [basé sur XAdES]
- Un ensemble de librairies, permettant l'implémentation de clients.

La *Smart Card* (traduction Finlando-anglaise ;-) est actuellement en production aussi bien côté Finlande que côté Estonie[22]…

2. Aspects juridiques

2.1 Contexte juridique

Le cadre juridique définissant le statut de la signature électronique en France, et plus généralement en Europe, est le résultat de la transposition de la directive européenne 1999/93/CE.
Les différents textes sont :
- 1999 : **Directive Européenne 1999/93/CE**
- 2000 : **Loi n°2000-230 du 13 mars 2000** :
  Prise en compte de la signature électronique au sein du code civil.

---

[14] http://www.w3.org/Signature/2001/04/05-xmldsig-interop.html
[15] http://www.aleksey.com/xmlsec
[16] http://xml.apache.org/security/index.html
[17] Référence ETSI : « ETSI TS 101 903 V1.1.1 (2002-02) »
[18] European Telecommunications Standards Institute (ETSI), http://www.etsi.org, Portail technique : http://portal.etsi.org
[19] Référence W3C : http://www.w3.org/TR/2003/NOTE-XAdES-20030220/
[20] Les organismes Finlandais et Estoniens du projets sont : le « *AS Sertifitseerimiskeskus* » et le « *Finnish Väestörekisterikeskus* » accessibles aux URLs suivantes : http://www.sk.ee et http://www.fineid.fi
Pour en savoir plus, voir « *The Estonian ID Card and Digital Signature, Concept Principles and Solutions, Whitepaper Version: June 5, 2003* » : http://www.id.ee/file.php?id=122
[21] Pour en savoir plus : http://www.id.ee
[22] Les statistiques Estoniennes, indiquent que plus de 300.000 « id-kaart » sont actuellement en circulation, dont 10% pour des non estoniens.

- 2001 : **Décret n° 2001-272 du 30 mars 2001**
  Transposition de la Directive Européenne 1999/93/CE
- 2002 : **Décret n° 2002-535 du 18 avril 2002**
  Attribution du rôle de certificateur à la DCSSI
- 2002 : **Arrêté du 31 mai 2002**
  Attribution du rôle d'accréditeur au COFRAC, pour l'évaluation des prestataires de certification électronique.

2.2 Synthèse

Les textes définissent deux types de signatures ;

La signature électronique simple, qui n'est pas présumée fiable jusqu'à preuve du contraire. En cas de contestation, c'est donc à celui qui veut se prévaloir des effets juridiques de cette signature d'apporter la preuve de la fiabilité du système mis en œuvre.

La signature électronique présumée fiable, qui ne peut être contestée qu'en apportant la preuve de sa non fiabilité.

Pour être « *présumée fiable* », un « *procédé de signature électronique* » doit remplir trois conditions :
- La signature électronique est sécurisée
- La signature électronique est établie grâce à un dispositif sécurisé de création de signature
- La vérification de la signature électronique repose sur l'utilisation d'un certificat électronique qualifié, émis par un prestataire de service de certification électronique.

Les différentes exigences sont résumées dans la figure 7.

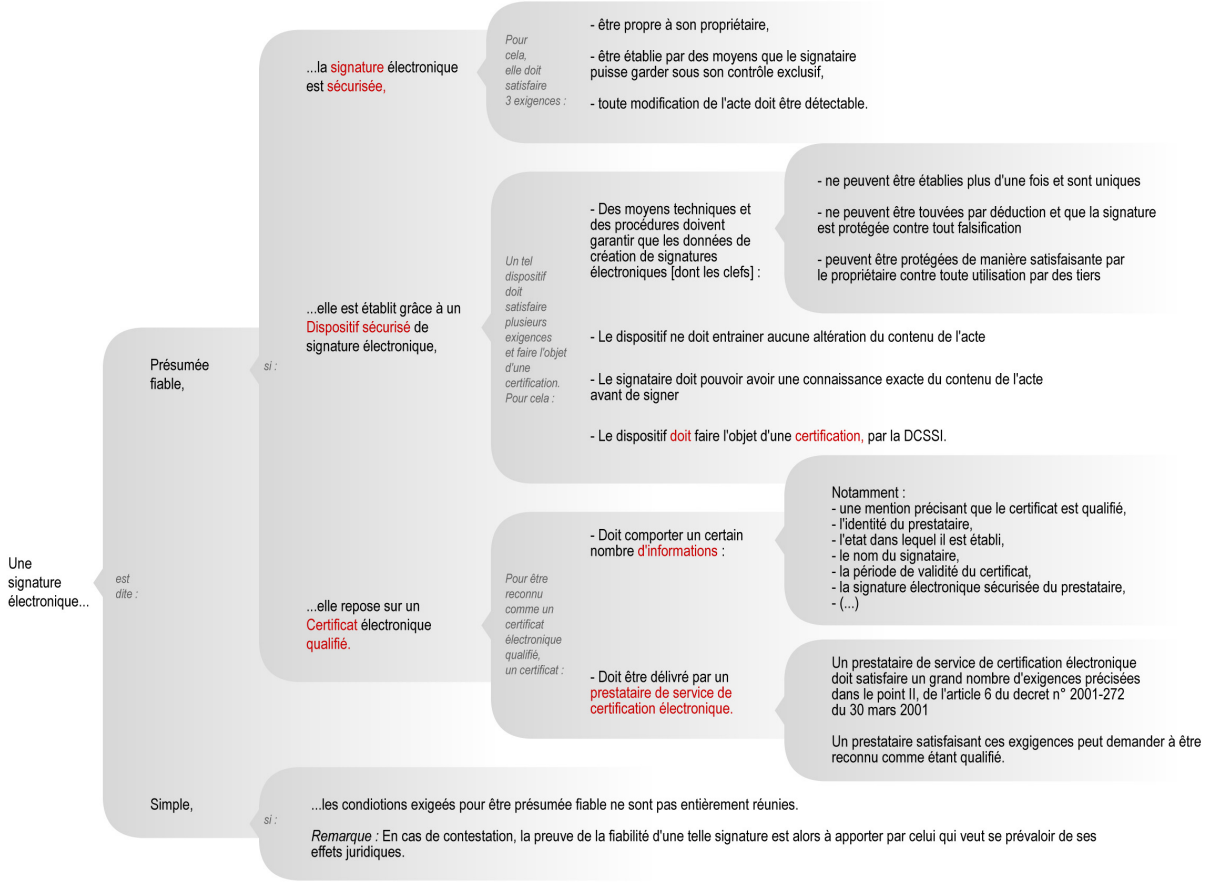

Figure 7 : *Cadre juridique de la signature électronique en France*

3. Contractualisation des actes électroniques

3.1 Niveau de contractualisation

Par « contractualisation d'une procédure électronique », nous entendons le fait de conférer à une procédure électronique une valeur contractuelle.

Cela passe généralement par l'acceptation des différentes parties des mêmes termes d'un contrat, avec une preuve de cette acceptation réciproque.
Dans ce contexte, la disponibilité d'une « *signature électronique présumée fiable* » au sein d'un document électronique devrait offrir une solution parfaitement adaptée…

Mais les choses ne sont pas aussi simple ;-)
Comme nous l'avons vu, il n'est pas facile de pouvoir répondre à l'ensemble des exigences indispensables à la mise en œuvre d'une telle signature.

Aussi dans la pratique, plusieurs niveaux de « contractualisation » peuvent être identifiés[23] :

- **Un premier niveau, basé sur la simple dissuasion, repose sur une authentification déclarative sans véritable contrôle**. On peut citer les « demande d'extrait de casier judiciaire (bulletin n°3). » où le formulaire électronique de demande, se contente de préciser : « *L'extrait de casier judiciaire ne peut être demandé que par la personne qu'il concerne ou son représentant légal s'il s'agit d'un mineur ou d'un majeur sous tutelle.* » Et de préciser : « *Se faire délivrer l'extrait de casier judiciaire d'un tiers est sanctionné par la loi (article 781 du Code de procédure pénale).* »

- **Un deuxième niveau de contractualisation peut être définit lorsque des moyens organisationnels et techniques sont mis en œuvre afin de garantir « au mieux » l'identité des différents acteurs**, sans pour autant que l'on puisse parler de « preuve » au sens juridique du terme. On peut citer en exemple les achats « en lignes » où le vendeur se contente d'identifier une carte bleue (et peut-être son propriétaire ;-) Plus « sûr » en terme de sécurité, mais tout aussi dépourvu de « valeur de preuve », l'usage d'un accès restreint par login/mot de passe sur un système d'information n'est guère plus satisfaisant d'un point de vue contractualisation.

- **Un troisième niveau peut être défini, lorsque l'on rentre dans le périmètre de la « signature électronique simple ».** Un document électronique signé est recevable d'un point de vue juridique, même si en cas de contestation la preuve de sa fiabilité reste à apporter. A noter que beaucoup de procédures électroniques faisant intervenir une authentification « forte », par certificats, sont réalisées au travers de formulaires Web et donc ne conduisent pas à la fourniture d'un document signé en tant que tel. Un document signé avec un certificat du trésor public peut avoir, en revanche, valeur de preuve.

- **Dernier niveau, celui de la « *signature électronique présumée fiable* ».** Les conditions devant être remplies sont nombreuses et difficiles à réunir, mais la valeur juridique d'un document signé avec une signature de ce type est équivalente à un document papier. Il existe peu d'exemples de contractualisation de ce niveau.

L'administration électronique est classiquement découpée en 4 niveaux d'interactivité ou de services[24], seul le dernier niveau est véritablement concerné.

Les moyens à mettre en oeuvre sont rapidement très importants vis-à-vis des bénéfices et/ou des risques.

3.2 Choix d'un niveau de contractualisation

Le choix d'un niveau de contractualisation sera le résultat d'une dichotomie entre :
- Le coût des moyens à mettre en œuvre
- Le coût lié aux différents risques
- Les bénéfices attendus

[23] Les niveaux présentés concernent la valeur juridique des procédures, non leur niveau de sécurité au sens informatique du terme.
[24] Niveau 1 : information, Niveau 2 : possibilité de récupérer les formulaires, Niveau 3 : aide électronique comme le calcul des impôts, Niveau 4 : Délivrance directe d'un service ou d'un produit en ligne, exemple : télé déclaration, demande de passeport, etc.

Ainsi, les achats par carte bleue, que ce soit via Internet ou par téléphone, sont exposés à la contestation de l'acheteur. Le vendeur ne pouvant garantir l'identité du « porteur » de la carte bleue, l'acheteur (supposé) aura donc *a priori* gain de cause.

Néanmoins, le bénéfice induit par la souplesse de ce mode de vente, reste (manifestement) très supérieur au coût des transactions contestées.

Dans le cas du trésor public et de la télé déclaration, l'utilisation d'une technologie de « *signature simple* » est également une forme de compromis.
- Le processus est protégé par une authentification forte, offrant une bonne confiance sur le plan technique.
- Il n'y a pas d'acte signé présumé fiable, mais la déclaration est juridiquement recevable (même si le système reste sans doute en deçà de la procédure papier)
- La procédure électronique permet de grosses économies dans le traitement des dossiers par rapport au papier[25]
- Un meilleur service est offert, etc.

4. État de l'art

4.1 Signature numérique dans Microsoft Office XP

4.1.1 Service de certificats de Windows

La certification et les fonctionnalités de signature numérique sont des services offerts par les systèmes d'exploitation de Microsoft. Ce service repose sur l'utilisation d'un fournisseur de services cryptographiques (le CSP[26] pour *Cryptographic Service Provider*) dont l'instance par défaut est le Microsoft CSP v1.0. Il est cependant possible d'installer d'autres modules de certification. Le CSP Microsoft supporte un certain nombre de normes et de standards (X509v3, PKIX, CRL v2, S/MIME, SSLv3, TLSv1, PKCS #7, #10, #12, etc.) et permet l'utilisation de signatures numériques pour la messagerie électronique, la navigation sur le Web, la mise en place de systèmes de fichiers chiffrés (EFS[27]) et les documents bureautiques depuis la version « XP » de la suite bureautique Office.

L'accès au service de cryptographie est réalisée au moyen d'une interface de programmation standardisée pour tous les CSP : la *CryptoAPI*, actuellement en version 2.0. Elle permet une gestion du stockage des certificats en « magasins », en fonction de leur niveau de confiance ou de leur niveau hiérarchique, de leur importation ou de leur exportation. Elle est de plus complètement intégrée au *framework* .NET de Microsoft. La vérification de la révocation des certificats est faîte auprès de l'autorité de certification au moyen des CRL (*Certificate Revocation Lists*). Au sein d'un réseau de type Active Directory, il est possible de diffuser ces CRL au moyen de différents protocoles comme HTTP, LDAP et SMB. Le protocole OCSP n'est pas supporté.

4.1.2 Signature numérique de documents bureautiques

Comme nous l'avons précédemment dit, depuis sa version dite « XP » ou 2002, l'utilisateur de la suite bureautique de Microsoft dispose de fonctionnalités de signature numérique des documents. Celles-ci s'appliquent pour 3 types de documents : les présentations, les documents mis en page et les tableaux. Cette signature numérique de documents est réalisée à différents niveaux comme le montre la figure 8.

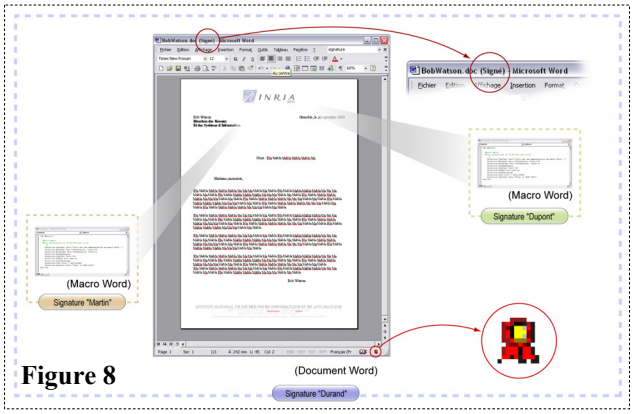

Figure 8

Dans un premier temps, au sein de chaque document, les macros peuvent être signées. Le document entier peut ensuite être signé à son tour en utilisant le même certificat numérique ou un autre. La signature multiple de document est aussi supportée. Il est donc possible à une ou plusieurs personnes de contresigner un même document. Cependant, dans tous les cas, la signature d'un document n'entraîne pas la signature des macros qu'il contient et réciproquement.

[25] Le nombre de télé déclaration aura été de 600.000 durant les mois de mars-avril 2003 (déclaration des revenus 2002), avec 12 millions de connexion sur le portail fiscal.
[26] Microsoft a fait le choix de définir sa propre interface de service cryptographique avec le CSP. Netscape a définit, pour sa part le PKCS#11, IBM le CDSA (*Common Data Security Architecture*), etc.
[27] EFS (*Encrypting File System*). Pour en savoir plus : http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp

4.1.3 Limites

D'un point de vue fonctionnel, si la signature multiple d'un document est possible, la traçabilité n'est que partielle car les signatures sont « juxtaposées » et non « englobantes ».
Il est donc impossible de reconstituer l'historique d'un document signé par plusieurs signataires.

Plus grave, la non disponibilité des spécifications des formats de documents (Word, Excel, PowerPoint, etc.) est rédhibitoire quant à la confiance qu'il est possible d'accorder à ces documents :

- On ne sait pas ce que l'on signe
- La vérification par une tierce application n'est pas possible

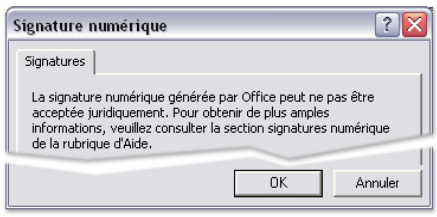

Figure 9 : *Avertissement Windows*

Si la « *signature électronique résumée fiable* » n'est donc pas envisageable, l'utilisation dans un cadre de « *signature simple* » n'est pas non plus évidente : comment apporter la preuve qu'une solution inconnue puisse être de confiance ?
Pour finir, Microsoft lui-même, bien conscient des limitations du système en affiche clairement les « limites juridiques » au moment de signer (figure 9);

4.2 Adobe Acrobat 6

Adobe concentre une part extrêmement importante de ses activités dans l'édition électronique, dont le fer de lance est la suite *Acrobat*, avec le format de document *pdf* et dont la version 6 intègre désormais la possibilité de signer des documents.

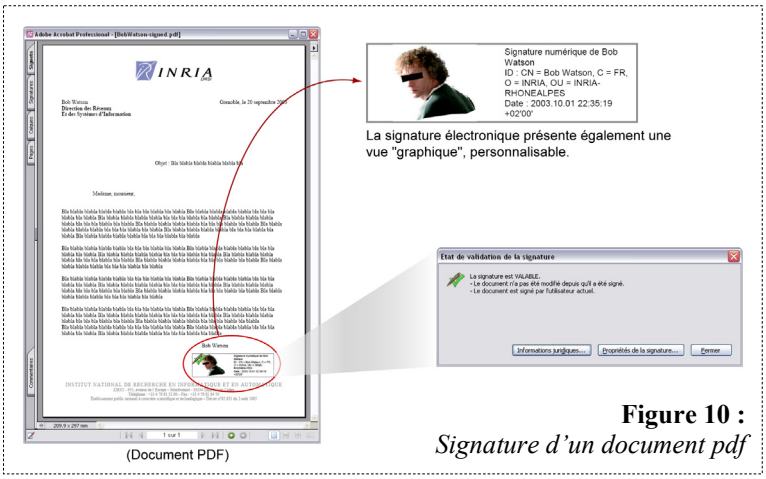

Figure 10 :
Signature d'un document pdf

A l'inverse de Microsoft, qui conserve jalousement les spécifications de ses formats de documents, Adobe a fait le choix de diffuser ouvertement les spécifications du format *pdf*[28].

Dans une logique d'interopérabilité[29], il est clairement spécifié que les formats de signatures doivent être compatibles avec les standards émis par le groupe PKIX de l'IETF. En pratique, les signatures mises en œuvre dans un document *pdf* sont des PKCS#7.

A noter que dans l'architecture mise en œuvre par Adobe, les mécanismes de signature doivent pouvoir s'appuyer sur des dispositifs d'authentifications divers, tels des lecteurs d'empreintes digitales ou rétiniennes, etc.
Autre caractéristique intéressante dans la signature de documents *pdf*, une identité visuelle peut être associée à la signature électronique. Il est ainsi possible de personnaliser – au sens humain du terme – une signature électronique. La vérification de la signature se faisant, par exemple, en cliquant sur le symbole visuel. Ce dernier pouvant être une signature manuscrite scannée, un tampon, une photo, etc. (figure 10).

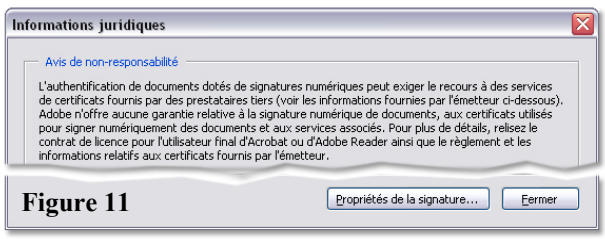

Figure 11

Enfin, côté cadre juridique, un « avis de non responsabilité » balise ici encore précisément le chemin (figure 11).
Néanmoins, le fait de pouvoir disposer des spécifications du format de document, d'utiliser des standards reconnus dans une optique d'interopérabilité affichée et de pouvoir disposer de plusieurs implémentations concurrentes fait que cette solution est pour le moins intéressante.

---

[28] Les spécifications du format pdf, dans sa version 1.5 [Correspondant à la version 6 d'Acrobat] sont disponibles à l'URL : http://partners.adobe.com/asn/tech/pdf/specifications.jsp (1172 pages…)
[29] *PDF Reference, fourth edition, Adobe® Portable Document Format Version 1.5,* Signature Interoperability (8.7.1), page 661

5. Problèmes liés à la signature électronique

La signature électronique est un moyen beaucoup plus fiable de vérifier l'authenticité d'un document. La vérification d'une signature électronique ne nécessite aucune expertise particulière. Ce qui n'est pas le cas pour la signature manuelle. En effet, comme on l'a vu lors de l'évasion de trois détenus de la prison de Borgo en mai 2001, il est assez facile de créer des faux documents papier qui peuvent induire gravement en erreur les personnes à qui ils sont destinés.

Malgré tout, la généralisation de la signature électronique n'est pas encore pour demain. Elle souffre d'un certain nombre de handicaps.

- **l'attachement à l'individu** : la signature manuelle est difficile à imiter. Avec un niveau d'expertise suffisant, il est possible de prouver si une telle signature est authentique ou non. Dans le cas de la signature électronique, il est aisé de dupliquer sa clé privée et son certificat et de les transmettre à un collaborateur. L'utilisation de jetons cryptographiques externes ne résout que partiellement cet inconvénient : on peut prêter son jeton. Certes, l'acte de dupliquer son certificat ou de prêter son jeton est en contradiction avec les chartes de bon usage. Malheureusement, face à des prétextes d'urgence ou de soi-disant efficacité, est-il possible de faire respecter les chartes ?

- **la pérennité** : pour vérifier une signature électronique, il faut disposer du document original sous une forme électronique ainsi que des outils implémentant les divers algorithmes. Si des documents ont été transmis par courriers électroniques signés, il faudra être en mesure de les collecter dans les boites à lettres des utilisateurs afin de les archiver sur des supports de longue durée. Est-ce que les contrôleurs de la Cours des comptes auront des outils pour vérifier la régularité des pièces signés électroniquement ? Aurons-nous encore dans 10, 20 ou 50 ans les outils qui permettent de lire ces supports ? de vérifier les signatures ?

- **un déplacement de la charge de travail** : la signature électronique est bien implantée dans la plupart des outils de messagerie. On peut donc imaginer transposer toutes les procédures papier vers des procédures utilisant des documents électroniques signés. Le gain serait évident : en délais de transmission des documents originaux, en coûts d'affranchissement, en saisies des données du papier vers les systèmes d'information. Si on focalise sur l'acte de signature lui-même, on constate qu'actuellement, les responsables de nos structures, signent un nombre important d'actes, classés dans des parapheurs. Les dossiers ayant été instruits par des collaborateurs, le signataire n'a besoin que de quelques secondes par acte. Il retransmet ensuite le parapheur à ses collaborateurs qui font suivre chaque dossier vers son destinataire final. Dans le cas de la signature par messagerie électronique, ce processus est difficile à mimer. En tout état de cause, il risque de demander au signataire davantage de réflexion pour chaque acte. Ce dernier n'est pas prêt à assumer cette charge, même si elle se traduit par un gain pour d'autres. La solution passe vraisemblablement par un mécanisme de *workflow*, piloté par une application WWW et qui joue le rôle de *parapheur électronique*.

- **habilitations** : la signature électronique d'un document peut être aisément vérifiée par le destinataire. Toutefois, rien n'indique que le signataire est autorisé à signer un tel acte. Ce problème d'habilitation n'est pas spécifique à la signature électronique mais cette dernière ne l'a pas résolu. Les certificats d'attributs permettront certainement d'accompagner la signature d'un document des habilitations du signataire. Il y a toutefois peu de chances que les outils de messagerie standards puissent interpréter correctement ces attributs. Là aussi, la mise en place de mécanismes de *workflow* pourrait pallier à ce problème.

- **Fracture technologique** : comme beaucoup d'évolutions, l'utilisation croissante de ces technologies risque de conduire à un schéma d'exclusion. Si au sein d'une organisation structurée, comme une entreprise ou une administration, des mécanismes d'accompagnement sont disponibles (formation, support, etc.), leur mise en œuvre dans un contexte plus large est souvent plus difficile, plus coûteuse. Tout le monde sera-t-il capable d'effectuer une télédéclaration ? La mise en place de tels outils ne risque t-elle pas de se faire au détriment des procédures traditionnelles ? Quel sera le poids des usagers « historiques » à terme ?

6. Sign@tor

6.1 Objectifs et contexte

Le projet *Sign@tor* est le résultat d'une multiple constatation :

- La plupart de nos systèmes d'information sont accessibles via des interfaces HTML.
- Aucune solution satisfaisante ne permet la signature de formulaires HTML[30]
- Il existe plusieurs architectures cryptographiques et il est difficile de n'en considérer qu'une seule.

Si au sein d'une organisation homogène, le contrôle et la maîtrise des postes de travail permet de disposer d'un parc relativement cohérent, dés lors que l'on va vouloir atteindre un public plus large, l'hétérogénéité des systèmes et des outils devient un facteur essentiel à intégrer.

L'objectif de *Sign@tor* est de proposer une solution à la signature de formulaire HTTP dans un environnement ouvert, multi-plateforme, en s'appuyant sur l'une ou l'autre des différentes couches cryptographiques disponibles et avec la possibilité d'en intégrer aisément de nouvelles.

Les différentes couches cryptographiques, implémentant les principaux « standards[31] » et dont le support est souhaité, sont :

- **Windows**, qui propose deux interfaces : *CryptoAPI* et *.NET*[32]
- **Netscape**, qui propose une solution open source, NSS[33]
- **Java**, qui ne propose qu'une implémentation partielle.

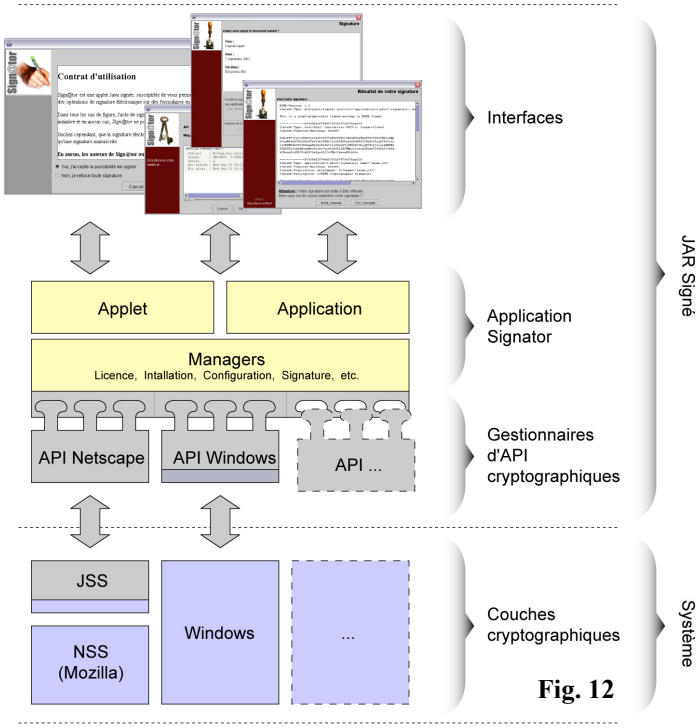

Fig. 12

Outre la fonction de « *signature simple* », *Sign@tor* doit pouvoir offrir un certain nombre de services tels que l'archivage des textes signés, la journalisation des actes, l'archivage parallèle distant, la possibilité d'horodater, etc.
De la même façon que pour la gestion des différentes couches cryptographiques, l'adjonction de nouveaux services doit pouvoir être simple et modulaire.

Enfin, et c'est encore là un élément essentiel, l'interface et la robustesse du code sont des éléments primordiaux étant donné la criticité de l'application. Un soin tout particulier concernant ces aspects a donc été pris tout au long des phases d'étude et de développement.

Sign@tor est le fruit d'une coopération entre **l'INRIA** et le **CNRS**.

6.2 Architecture logicielle

La technologie retenue est *Java*, avec un « conditionnement » en applet signée, afin de pouvoir accéder aux ressources systèmes[34]. Les raisons de ce choix sont celles énoncées précédemment ; portabilité de la solution, robustesse de la technologie, possibilité de disposer d'une interface graphique de qualité, etc.

---

[30] Une session HTTPS permet de garantir authentification, intégrité et confidentialité, mais sans pour autant que les informations échangées ne soient signées.
[31] Parmi les standards incontournables : x509, pkcs7, pkcs12, SMIME
[32] L'architecture.Net offre un support pour les algorithmes symétriques, asymétriques et de hachages usuels. Une gestion minimale des certificats X509 est également pourvue. A noter que .NET intègre une implémentation complète de XML Signature [W3C].
Pour en savoir plus : http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconcryptographyoverview.asp
[33] *Overview of NSS, Open Source Crypto Libraries* : http://www.mozilla.org/projects/security/pki/nss/overview.html
[34] Une applet signée peut se voir attribuer des privilèges, lui permettant de s'exécuter en dehors de la « sand box ».

Afin de permettre une bonne adaptabilité aux besoins et un développement conjoint, indispensable étant donné le spectre des technologies utilisées, une architecture logicielle totalement modulaire a été retenue (figure 12).

Les différents composants intervenants dans le fonctionnement de *Sign@tor* sont:
- Les **gestionnaires d'API** qui font l'interface entre l'application et les couches cryptographiques.
- Les **Agents**, qui prennent en charge les services d'archivage, de journalisation, etc.
- Les **fonctions**, offrant les grandes fonctions, comme la signature.

6.3 Mise en oeuvre

L'Applet est insérée dans le document HTML, et apparaît sous la forme d'une petite icône.

L'interface entre le document HTML et *Sign@tor* se fait par des appels *Javascripts*, permettant d'effectuer les différentes étapes du processus de signature. Le déclenchement proprement dit de celle-ci intervenant, par exemple, lorsque l'on cliquera sur le bouton « Envois » du formulaire. La signature d'un formulaire comprendra les étapes suivantes (figure 13) :

- Les paramètres du formulaire sont soumis à l'Applet, sous la forme de couples (*nom-de-paramètre*, *valeur*)
- Un format de structuration des données est précisé (concaténation simple, tableau html, xml, etc.)
- Un format de sortie est précisé (S/MIME ou PKCS #7)
- La demande de signature est déclenchée.
- Le résultat est récupéré et rangé dans un champ *hidden* (par exemple)
- Le formulaire est envoyé…

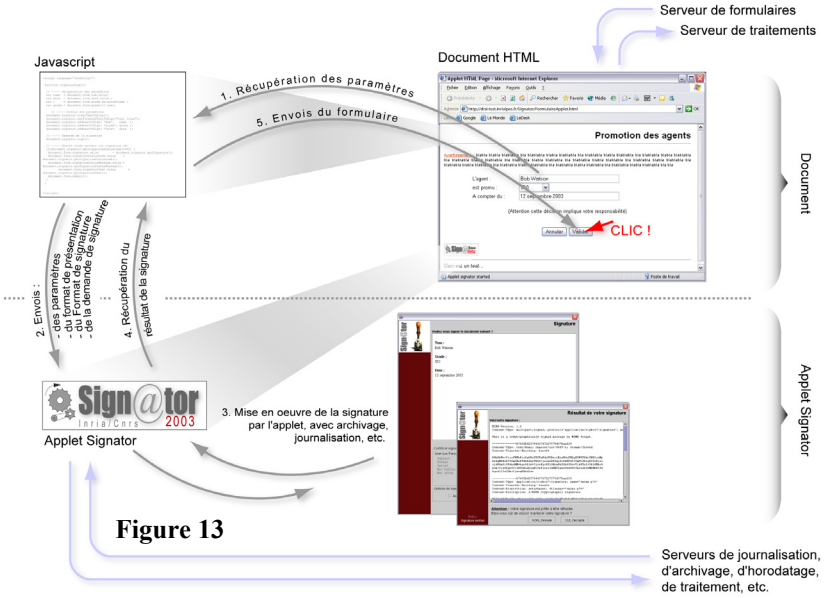

Figure 13

Lorsque *Sign@tor* va être sollicité pour la première fois, un processus d'acceptation de licence, d'installation et de configuration va devoir être effectué. L'objectif de la licence est de clairement présenter au signataire le fait qu'il engage sa signature en tant que tel, l'installation va consister à installer dans le *homedir* du signataire un espace pour les journaux et les archives. L'étape de configuration va lui permettre de sélectionner ses préférences.
Une fois ces étapes effectuées, le panneau de signature lui sera directement proposé.

La plupart des éléments, tel que le libellé de la licence, les textes d'avertissement, les options accessibles à l'utilisateur, etc. sont entièrement paramétrables. L'ensemble des préférences de l'utilisateur sont également sauvegardées, dans l'espace *Sign@tor* de son *homedir*.

6.4 Bilan & perspectives

Le prototype actuel fonctionne et est capable de signer en s'appuyant sur *l'API Mozilla*. L'implémentation actuelle intègre l'ensemble des contraintes de modularité et des fonctionnalités de contrôle. La faisabilité du concept est ainsi démontrée. Reste maintenant à en faire un produit utilisable en exploitation ;-)

En plus de la signature, d'autres fonctions doivent pouvoir être intégrées, comme la demande de certificat ou la gestion de CA, permettant ainsi de couvrir d'autres « segments » de nos besoins. *Sign@tor* peut ainsi être vue comme une sorte de plateforme cryptographique, polyvalente et ouverte, côté client.

Il ne semble pas exister beaucoup de solutions comparables, notamment dans le monde open source :-(

Une réflexion est en cours concernant l'évolution fonctionnelle et la définition d'un cadre approprié au projet *Sign@tor*.