Signatures électroniques multiples

Proposition de formalisation de signatures électroniques multiples et processus associés.

Damien Rieupet - Maxime Wack - Nathanael Cottin – Didier Assossou

Laboratoire Systèmes Et Transports
Université de Technologie de Belfort-Montbéliard
90010 BELFORT Cedex
{damien.rieupet,, maxime.wack, nathanael.cottin, didier.assossou}@utbm.fr

RÉSUMÉ. *La certification et l'archivage légal des données, alliée à la signature électronique des documents, ouvrent de nouvelles perspectives à la sécurisation des documents. Ainsi, ces technologies offrent des capacités : d'identification, d'authentification, de certification qui concourent à la capacité globale d'archivage sécurisé des données numériques.*
Cependant, il apparaît qu'une signature électronique simple ne répond pas complètement aux besoins des entreprises. En effet, des types de signatures plus évolués (co-signature, sur-signature…) peuvent se révéler nécessaires pour signer et valider un document.
Dans la suite de cet article, nous proposons une description, une modélisation ainsi que des procédures pour réaliser ces signatures.
ABSTRACT. *Certification and legal data storage, bound to data electronic signature open new perspectives to data security. Indeed, these technologies offer capabilities such as: identification, authentication, certification, aiming at increasing the global capacity of secured data storage.*
However a simple electronic signature does not completely answer to enterprise needs.
To perform previously paper-based signed transactions electronically, multiple or special signature are necessary for business and government agencies.
In this paper, we suggest a description, a model, and the related procedures to realize these types of signature.

MOTS-CLÉS : *document, légal, archivage, sécurité, certification, autorités, signature, multi signature, clés.*

KEYWORDS: *document, legal, storage, security, certification, authorities, signature, multiple signature, keys.*

1. Introduction

Depuis la loi n°2000-230 du 13 mars 2000 relative à la signature électronique (JO du 14 mars 2000, p.3968), le support d'archivage de la preuve n'est plus obligatoirement un support papier mais aussi un support électronique. Ce support répond aux caractères de fidélité et de pérennité énoncés par le Code civil ainsi qu'aux exigences futures d'intégrité et d'imputabilité de la preuve.

Dans le cadre de la signature électronique, afin de satisfaire aux exigences des professionnels, il faut être en mesure de proposer un modèle électronique couvrant tous les besoins des utilisateurs en terme de signature multiples et hiérarchiques. Pour répondre à ces besoins, nous allons proposer une modélisation et des procédures de signature multiple conformes aux recommandations précédemment citées, après avoir décrit les principes de signature numérique.

2. Signature numérique et bases de la signature du message

La signature numérique est le moyen courant d'authentification d'une donnée électronique. Elle est le résultat de nombreuses recherches sur la cryptographie asymétrique et le code de hachage.

2.1. *Concepts de cryptographie asymétrique*

Quand une entité émettrice (une personne, un serveur ou un programme) doivent envoyer un message sécurisé à une entité réceptrice, elle crypte le message en utilisant la clé publique du récepteur. Cette clé est diffusée de telle sorte que tout émetteur puisse utiliser la clé publique du récepteur pour crypter la donnée. Le message crypté est ainsi illisible et ne peut être décrypté sans la clé privée correspondante. La clé privée doit être conservée de manière sécurisée par le récepteur, qui ne doit pas la publier. Seul le récepteur doit être capable de décrypter le message codé. Le cryptage asymétrique assure le caractère privé et la confidentialité.

Les algorithmes asymétriques les plus largement utilisés sont RSA (RSA, 1993) et triple-DES (NIST, 1999).

2.2. *Le code de hachage*

Le code de hachage (Menezes, 2001) a pour but la création d'un message de longueur fixe pour tout ensemble de données de taille variable. Ce code est indépendant de la taille des données sources. Considérons h(), une fonction de hachage à sens unique utilisée pour calculer un code sur un ensemble de données s. La plus importante propriété de cette fonction est de permettre la reconstruction de

l'ensemble de données seulement si le code calculé est connu. Bien que la reconstruction des données d'origine s à partir d'un code donné d soit être théoriquement possible, elle apparaît comme informatiquement infaisable :

$(h(s)=d) \Rightarrow (p(h^{-1}(d) = s) \rightarrow 0)$

De plus, la probabilité p que deux différents ensembles de données s1 et s2 obtiennent le même code avec un algorithme de hachage donné ha tend vers 0. La fonction de hachage est ainsi dite résistante aux collisions :

$(s1 \neq 2) \Rightarrow (p(h(s1,ha) = h(s2,ha)) \rightarrow 0$

De nombreux algorithmes de hachage tels que MD2 (Kaliski, 1992), MD4 (Rivest, 1992) et RIPEMD (Dobbertin, 1996) (Preneel, 1997) ont été développés. Les algorithmes très répandus SHA-1 (NIST, 1995), et MD5 (Rivest, 1992) sont spécifiquement conçus pour le calcul des signatures numériques.

2. 3. *Signature numérique*

Les signatures numériques reproduisent les sceaux de cire utilisés dans l'antiquité pour cacheter les lettres (NIST, 2000).

Le sceau peut être comparé à une clé de signature secrète qui ne doit être en possession que du signataire, c'est à dire l'entité qui signe le message. Bien que le sceau reste indépendant de l'information de la lettre, la signature numérique est dépendante du message. Cette manière d'appliquer une clé de signature (la clé privée du signataire) à deux différents messages va résulter en deux signatures numériques différentes. Au contraire, le même message va toujours générer la même signature dans le cas où un algorithme de signature donné est utilisé. Cependant, la clé de vérification (la clé publique) unique correspondant au signataire doit être utilisée pour être certain que la signature a été générée en utilisant sa clé de signature.

La génération des signatures numériques est la simple application du cryptage asymétrique sur les données des codes de hachage. Contrairement au cryptage de données, la signature numérique ne préserve pas la confidentialité des données, mais assure plutôt (Kaeo, 1999) :

- l'intégrité des données : les signatures numériques permettent de détecter les sources de modification des données, c'est à dire les modifications non autorisées des données

- l'authentification : comme la clé de signature est (théoriquement) détenue seulement par le signataire, il est impossible à toute autre personne de générer la signature de l'émetteur sur un ensemble de données. La donnée est authentifiée en comparant la signature avec la clé de vérification correspondante du signataire.

- la non-répudiation : ce service basé sur l'authentification est une preuve effective de la transaction. L'entité de la signature ne peut nier l'auteur de la signature parce que personne d'autre n'a pu créer une telle signature sur un ensemble de données particulières.

La signature numérique est généralement calculée sur les codes de hachage plutôt que directement sur les données.

Bien que la signature numérique permette l'authentification de la donnée reçue, elle n'identifie pas le signataire. Ainsi, aucun lien irréfutable n'existe entre le signataire et sa clé de signature. Une telle identification est permise par le certificat électronique.

3. Certificat électronique (qualifié)

3.1. *Présentation*

Un certificat électronique qualifié (nommé « certificat » par la suite) est une preuve électronique d'identité (Figure 1.) délivrée sous certaines conditions. Il est destiné à permettre l'identification de l'émetteur par les récepteurs de messages signés. La confiance dans les signatures dépend de la confiance que les destinataires attribuent aux fournisseurs des certificats. Seules les autorités de certification (ACs) accréditées par les gouvernements ou leurs représentants sont aptes à délivrer des certificats électroniques à valeur juridique.

Figure 1. *Description d'un certificat*

En sus d'un rôle de protection des données (par chiffremnt), les certificats peuvent être utilisés pour :

- signer les emails : les certificats peuvent être intégrés à l'intérieur de standards d'emails sécurisés tels que PGP (Garfinkel, 1994) (Callas, 1998) (Elkins, 2001) et S/MIME (Ramsdel, 1999)

- signer du code : les Archives Java (Sun, 2001) (Farley, 1998) et Authenticode Microsoft (Garfinkel, 1997) réalisent la plupart des certifications de code

- signer des documents : le document est signé à l'aide de la clé de signature (clé privée) conservée secrète par le signataire. Le certificat contient la clé de vérification qui sera confrontée à la signature afin de prouver son authenticité.

3.2. *Services associés :*

Les certificats sont valides jusqu'à ce qu'ils soient révoqués ou jusqu'à leur expiration (Figure 2.). Dans les deux cas, un nouveau certificat peut être ré-émis par l'AC. Une révocation de certificat intervient quand son propriétaire est informé que son certificat est corrompu, ou qu'une entité non autorisée ait pu l'utiliser. Il est aussi possible pour un gouvernement ou l'AC de révoquer un certificat dans le cas où son propriétaire en a fait une utilisation frauduleuse.

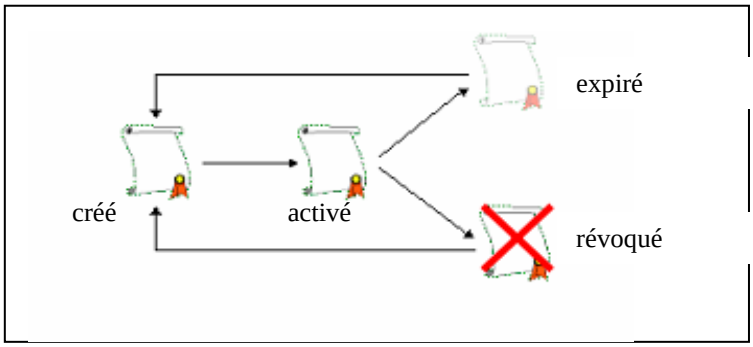

Figure 2. *Cycle de vie d'un certificat*

Ainsi l'AC propose un service de révocation qui se traduit par la publication d'une Liste de Certificats Révoqués (LCR) (Housley, 1999) consultable en ligne ou localement après téléchargement.

4. Les différents types de signature

Le National Institute of Standards and Technology (NIST, 2000) énumère un certain nombre de signatures multiples utilisé actuellement sous forme manuscrite. Afin de reproduire les schémas de signature traditionnels, la signature électronique doit fournir un équivalent de tous les types de signatures multiples. Dans la suite de ce document, nous proposons une formalisation et des procédures afin de réaliser électroniquement ces différentes signatures multiples.
La formalisation proposée utilise une loi que l'on pourra qualifier de loi d'apposition qui sera représentée par le symbole $\oplus$. Des exemples pour chaque type de signature permettront de comprendre cette modélisation.

Les processus permettant de signer et vérifier la signature d'un document, seront représentés par Réseaux De Pétri (RdP) (Murata, 1989)
Le terme document est ici utilisé au sens large, et peut donc désigner n'importe quel document numérique (texte, vidéo, son, image…)

4.1. *Signature unique simple*

C'est le cas de signature le plus simple où un individu unique signe un document (Figure3.).

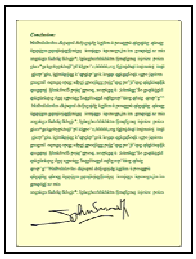

Figure 3. *Signature unique simple*

Le Document Signé (DS) résulte de l'apposition $\oplus$ d'une signature S sur un Document D.
On obtient ainsi la représentation suivante :

$$DS = D \oplus S \qquad [1]$$

Une fois cette définition formelle effectuée nous pouvons nous intéresser à la modélisation des processus. La signature d'un document comporte deux problématiques différentes.
La première appelée « processus de signature » concerne le ou les utilisateurs (les signataires) qui désirent signer un document, la deuxième appelée « processus de vérification » concerne l'utilisateur (le destinataire) qui désire vérifier l'authenticité d'un document et des signatures qu'il comporte.

4.1.1. *Processus de signature*

Pour signer un document, le signataire doit disposer d'une clé de signature verrouillée et stockée sur un support personnel et confidentiel (Cdcarte, carte à puce, token, clé USB…). La sécurité est en général assurée par un code secret mais dont moyens peuvent être utilisés en fonction du niveau de sécurité désiré (identification rétinienne, digitale, thermique…)
Le document à signer est « haché » afin d'obtenir une empreinte unique qui lui est propre. C'est cette empreinte qui est « signée » par la clé privée de l'utilisateur. On

obtient ainsi une signature qui se rapporte à un document unique. Cette signature est ensuite soumise à un ou plusieurs tiers horodateurs qui apposent leurs signatures et conservent une trace de la transaction ainsi que l'empreinte du document (Figure 4.).

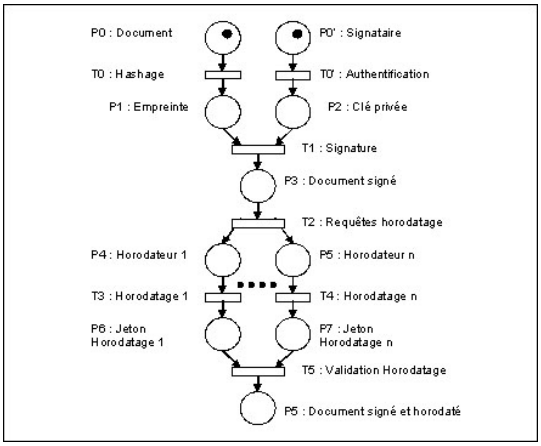

Figure 4. *Signature unique simple : processus de signature*

Lorsque l'utilisateur n'est pas en ligne, l'horodatage ne peut avoir lieu. Rien n'empêche l'utilisateur d'apposer sa signature, mais celle-ci ne sera valide qu'au moment où les horodateurs seront disponibles et auront délivré un horodatage.

Les heures délivrées par les différents horodateurs peuvent différer de quelques instants (temps de réponse du réseau et des horodateurs, files d'attentes, panne…) d'où la nécessité de définir un protocole de génération et de validation de l'horodatage faisant appel à plusieurs horodateurs (Cottin, 2003).
Une fois l'horodatage validé, le document peut être transmis au destinataire.

4.1.2. *Processus de vérification*

Le destinataire doit ensuite s'assurer que le document qu'il a reçu est bien celui qui a été signé et que la (ou les) signatures sont valides.
L'authentification du document est obtenue par une comparaison des empreintes reçues et calculées. Si ces empreintes sont concordantes, le document signé est bien celui reçu. Parallèlement, il faut s'assurer que la signature est bien celle attendue. Le certificat qui permet le décodage de l'empreinte par l'intermédiaire de la clé publique qu'il contient, doit être vérifié.
Une première étape consiste à vérifier que sa période de validité couvre bien la date de signature. Si ce n'est pas le cas, la signature n'a aucune valeur.
Ensuite, comme le certificat peut avoir été révoqué, il faut vérifier son statut. Ce contrôle peut se faire par l'intermédiaire des listes de certificats révoqués (LCR) ou

de l'OSCP (Myers, 1999) (Myers, 2001). Si il a effectivement été révoqué, il faut s'assurer que la date de révocation est postérieure à la valeur de l'horodatage.
La validation de ces étapes permet d'assurer que le document est bien celui signé, et la signature est valide. Dans le cas contraire, la signature est informatiquement rejetée (Figure 5.).

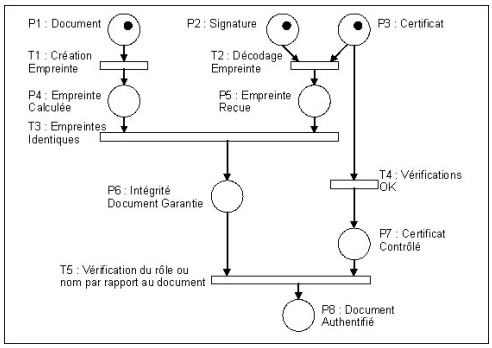

Figure 5. *Signature unique simple : processus d'authentification*

4.2. *Signature multiple simple ou Co-signature*

Ce type de signature se présente lorsque plusieurs individus signent un document (Figure 6.). Dans ce cas, les individus signent exactement le même document et toutes les modifications de ce document sont faites avant toute signature. Cela implique que le signataire sait qu'il voit seulement le document signé ou à signer, et non le fait que d'autres signatures soient présentes.

Figure 6. *Signature multiple simple ou Co-signature*

Le Document Signé (DS) est donc ici un unique document sur lequel est apposé une union de n signatures. Ce Document signé peut donc se représenter de la manière suivante.

$$DS = D \oplus \bigcup_{i=1}^{n} S_i \quad [2]$$

L'exemple à trois signatures présenté précédemment peut se représenter de la manière suivante :

$$DS = D \oplus (S_1 \cup S_2 \cup S_3) \quad [3]$$

Dans un souci de lisibilité, nous allons remplacer le processus d'horodatage par une seule étape (Figure 7.)

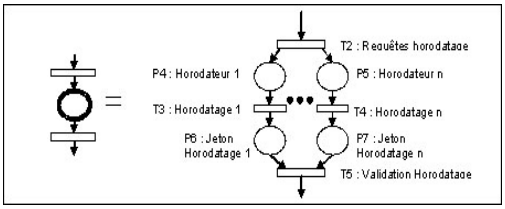

Figure 7. *Simplification du processus d'horodatage*

On peut imaginer deux processus de signature différents, suivants les besoins des utilisateurs. Le premier possède uniquement un horodatage global. Chacun appose sa signature (a tour de rôle par exemple) sur un document et l'horodatage fige les signatures (Figure 8.). Ce type de signature peut être utilisé pour co-signer un document à la fin d'une réunion par exemple.
Techniquement, on obtient le même processus que pour une sur-signature avec horodatage global uniquement. C'est la nature du document qui détermine le type de signature.

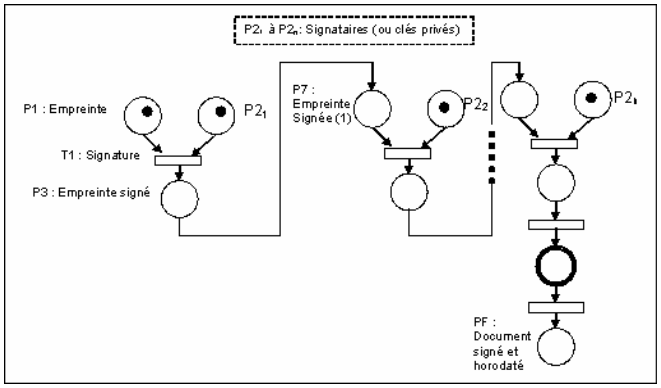

Figure 8. *Co-Signature avec horodatage global uniquement*

Le processus de vérification, s'effectue récursivement et la vérification de validité se fait par rapport à l'horodatage global (Figure 9.).

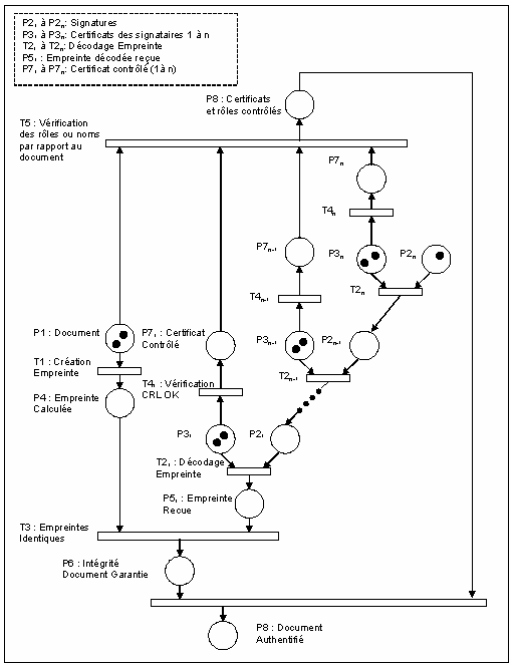

Figure 9. *Co-Signature : processus de vérification (horodatage global)*

La deuxième méthode fait intervenir un horodatage pour chaque signature (Figure10.).

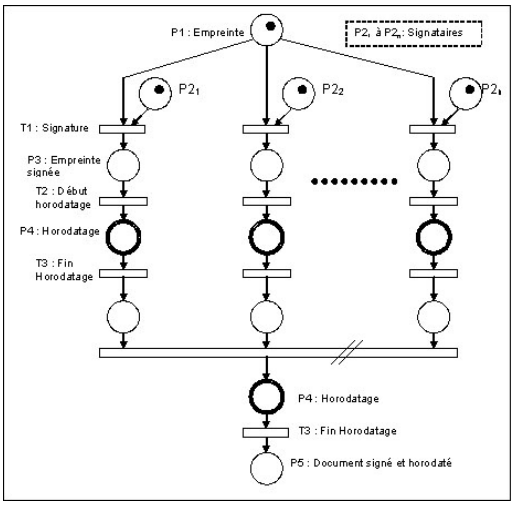

Figure 10. *Co-Signature avec horodatage global et de chaque signature*

Le processus de vérification est similaire à celui d'une signature simple, hormis le fait que plusieurs signatures sont à vérifier. Le document est authentifié uniquement quand toutes les vérifications on été faites (Figure 11.).

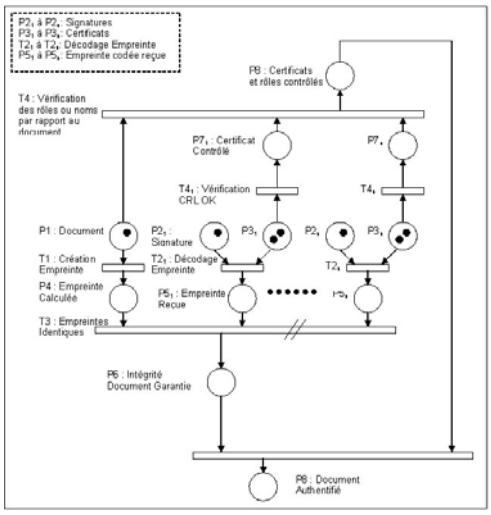

Figure 11. *Co-Signature : processus de vérification (horodatage multiple)*

4.3 *Signature multiple enveloppante ou Sur-signature*

Ce format s'utilise lorsque plusieurs individus représentant une structure hiérarchique donnée signent un document (Figure12.). Dans ce cas, les individus signent tous le même document mais les dernières signatures signent également les signatures précédentes. Le premier signataire signe le document, le second signe le document et la première signature, le troisième signe le document et les deux signatures précédentes et ainsi de suite. Si le document doit être modifié, le processus de signature recommence depuis le début avec le document corrigé.

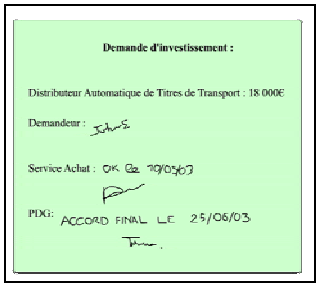

Figure 12. *Sur-Signature : exemple*

La représentation formelle de ce type de signature est la suivante :

$$\begin{aligned} &\mathrm{DS} = X_n = X_{n-1} \oplus S_n \qquad \forall n \in N^* \\ &\text{avec} \quad X_0 = D_0 \oplus S_0 \end{aligned} \qquad [4]$$

n et l'ordre des signatures étant défini par la structure hiérarchique, le niveau de signature S_i étant inférieur au niveau de signature S_{i+1}.
Sur l'exemple précédant cette représentation donne la formule suivante :

$$\mathrm{DS} = \left(\left(\mathrm{D} \oplus S_D\right) \oplus S_{SA}\right) \oplus S_{PDG} \qquad [5]$$

avec : S_D : Signature Demandeur
S_{SA} : Signature Service Achat
S_{PDG} : Signature PDG

Pour réaliser ce type de signature, deux possibilités sont offertes à l'utilisateur.
Un horodatage global (Figure 13.) peut en effet être utilisé lorsque tous les signataires apposent leur signature au même moment, lors d'une réunion par exemple, tandis qu'un horodatage individuel (Figure 14.) peut s'avérer préférable lorsque les signatures se font séparément, par exemple lorsque le document est transmis aux signataires par l'intermédiaire d'une messagerie électronique.

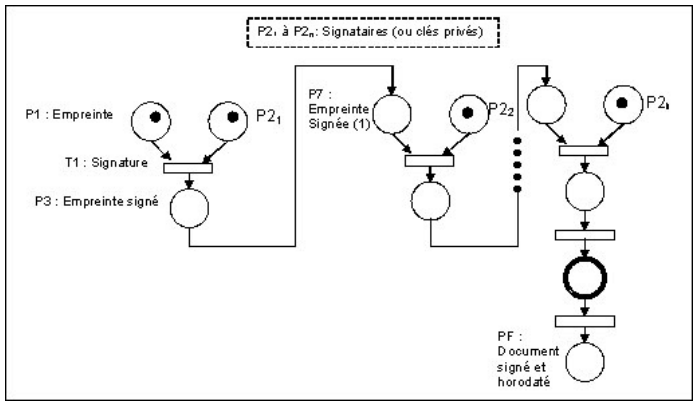

Figure 13. *Sur-Signature : horodatage global uniquement*

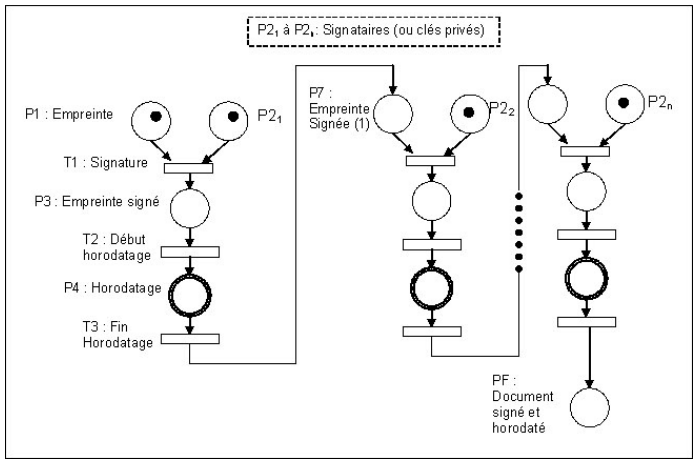

Figure 14. *Sur-Signature : horodatage individuel*

Pour les deux schémas de sur-signature, le processus de vérification est le même, seule diffère la date servant de référence pour vérifier la validité de la ou des signatures (Figure9.).

4.4 *Contre-signature*

Ce type de signature s'utilise lorsque plusieurs individus représentant une structure hiérarchique donnée signent un document. Dans ce cas, les individus signent tous le

même document mais les dernières signatures signent uniquement les signatures précédentes, et non pas le document.
Ce type de signature est utilisé par exemple pour les actes notariés. Le fonctionnement de cette signature est très voisin de la sur-signature présenté précédemment.

La représentation formelle de ce type de signature est la suivante :

$$\mathrm{DS} = \mathrm{D} \bigoplus_{i=1}^{n} S_i \qquad [6]$$

Si l'on applique ce modèle à l'exemple précédent on obtient :

$$\mathrm{DS} = \mathrm{D} \oplus S_D \oplus S_{SA} \oplus S_{PDG} \qquad [7]$$

avec : S_D : Signature Demandeur
S_{SA} : Signature Service Achat
S_{PDG} : Signature PDG

Techniquement les processus de signatures et d'authentification sont identiques à ceux de la sur-signature. La différence réside au niveau de document. Sa nature détermine la responsabilité des signataires.

4.5 *Signature simple par parties de document*

Ce format s'utilise lorsque plusieurs signatures apparaissent sur un document, mais que chaque signature s'applique à une partie différente du document (Figure 15.).

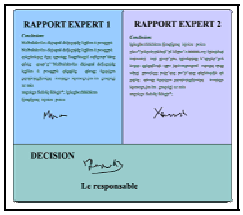

Figure 15. *Signature simple par parties de document*

Le Document Signé est dans ce cas une union de plusieurs documents signés (volets).

$$\mathrm{DS} = \bigcup_{i=1}^{n} \left(D_i \oplus S_i \right) \qquad [8]$$

La formalisation de l'exemple se traduit ainsi :

$$DS = (D_{E1} \oplus S_{E1}) \cup (D_{E2} \oplus S_{E2}) \cup (D_R \oplus S_R) \qquad [9]$$

avec : S_{E1} : Signature Expert 1
S_{E2} : Signature Expert 2
S_R : Signature Responsable
D_{E1} : Volet Expert 1
D_{E2} : Volet Expert 2
D_R : Volet Responsable

Cette catégorie de signature a un fonctionnement identique à celui d'une co-signature. En effet, si l'on considère un document finalisé, rédigé ou non par les signataires, le document peut préciser intrinsèquement quelles parties sont signées par quels signataires. Les responsabilités sont définies par le document, dont on assure l'intégrité par les différentes signatures apposées.
Dans ce cas, le document doit être est finalisé avant d'être transmis pour signature. Le nombre et la qualité des signataires sont connus. Ils peuvent également être identifiés nommément sur le document.

Il peut aussi se présenter le cas suivant. Chaque signataire rédige la partie du document le concernant sans visibilité sur que sera le document final.
Le document final sera donc constitué par un agrégat de documents simples signés. Pour figer cet agrégat il est nécessaire de le signer. Cette signature peut être celle d'un horodateur (Figure 16.).

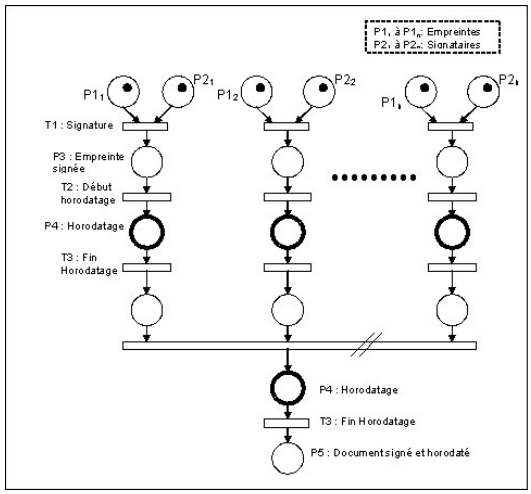

Figure 16. *Signature simple par parties de document*

Cependant, les rédacteurs pourraient tout aussi bien assembler leurs différentes parties afin de constituer un document commun finalisé, et ainsi revenir à un

processus identique à celui de la co-signature décrit précédemment (Figure 10), le document faisant le lien entre les parties et les signataires.

4.6 *Signature multiple enveloppante avec ajout d'information seulement*

Ce format s'utilise lorsque une personne désire ajouter des informations à un document et les signer. Sa signature couvre toutes les informations et les signatures antérieures (Figure 17.).

Figure 17. *Signature multiple enveloppante avec ajout d'information seulement*

Il est possible de modéliser ce type de signature par une suite :

$$\begin{aligned} &\mathrm{DS} = X_n = (X_{n-1} \cup D_n) \oplus S_n \qquad \forall n \in N \\ &\text{avec} \quad X_0 = D_0 \oplus S_0 \end{aligned} \qquad [10]$$

En appliquant cette méthode sur l'exemple précédent on obtient :

$$\begin{aligned} X_0 &= D_0 \oplus S_0 \\ X_1 &= (X_0 \cup D_1) \oplus S_1 = ((D_0 \oplus S_0) \cup D_1) \oplus S_1 \\ \mathrm{DS} = X_2 &= \left(\left(\left(\left(D_0 \oplus S_0\right) \cup D_1\right) \oplus S_1\right) \cup D_2\right) \oplus S_2 \end{aligned} \qquad [11]$$

Le processus de signature fait appel à des versions d'un document. Le n^{ième} signataire signe sa version du document et toutes les signatures précédentes. Sa version du document ne fait qu'ajouter des informations au document précédent.

Techniquement, ce type de signature modifie le document initial, puisqu'il y ajoute des informations. Hors une modification du document invalide l'empreinte initiale. Il est donc nécessaire de gérer ici plusieurs versions du document. En fait cette signature peut être envisagée comme une sur-signature de tous les documents précédents associée à une signature simple de son ajout.

Au final, le document obtenu sera en fait le document initial sur-signé n fois, plus l'ajout numéro 1 sur-signé n-1 fois, plus l'ajout numéro 2 sur-signé n-2 fois, et ainsi de suite jusqu'à l'ajout numéro n signé 1 fois.
La vérification de ce type de signature est alors celle d'une sur-signature (Figure 9.) répétée n fois. Ce qui produit au total ***n!*** signatures à réaliser et vérifier.
Une deuxième approche consiste à signer un document composé du document précédemment signé et de l'ajout effectué. Techniquement, on crée un package contenant le document précédemment signé, la signature précédente et le document ajouté (Figure18.). Ce package peut également contenir le certificat du signataire précédent.
Le processus de vérification de la signature ressemble à celui de la sur signature, avec une étape de d'extraction des packages supplémentaire (Figure 19.).

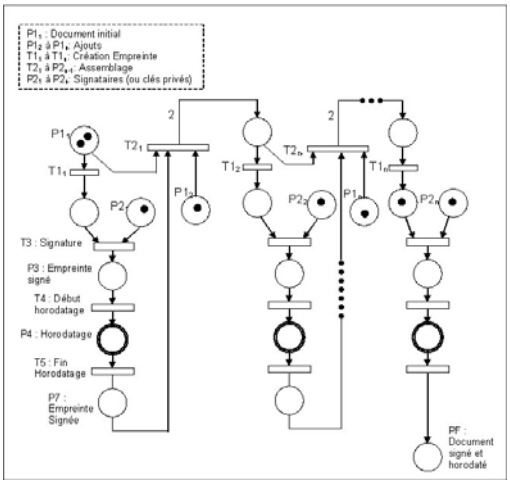

Figure 18. *Signature multiple enveloppante avec ajout d'information seulement : Processus de signature*

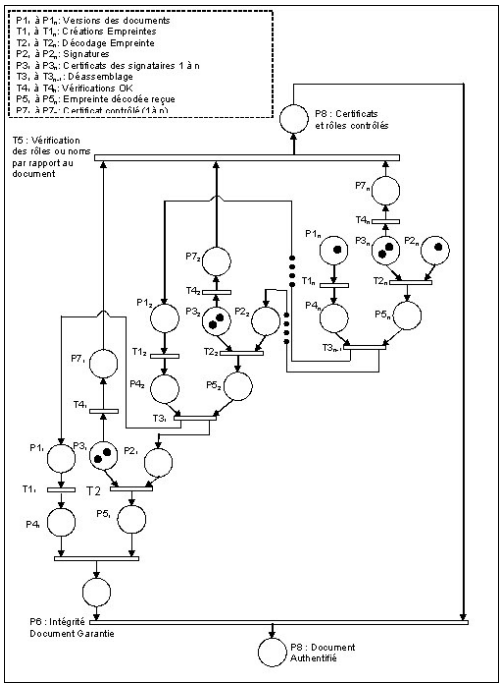

Figure 19. *Signature multiple enveloppante avec ajout d'information seulement : Processus d'authentification*

4. 7. *Signature multiple enveloppante avec ajout et suppression d'information*

Ce format s'utilise lorsqu'un signataire peut modifier unilatéralement un document qui contient de multiples signatures antérieures à sa signature.

Une modélisation simple de cette signature peut être la suivante :

$$\mathrm{DS} = D_{\mathrm{n}} \oplus S_n \qquad [12]$$

S_n : Dernière signature
D_n : Document tel que modifié par le dernier signataire

Ce type de signature peut être employé par exemple pour signer différentes versions d'un plan dans un bureau d'étude. Le dernier signataire valide le plan et les dernières modifications effectuées (Figure 20.).

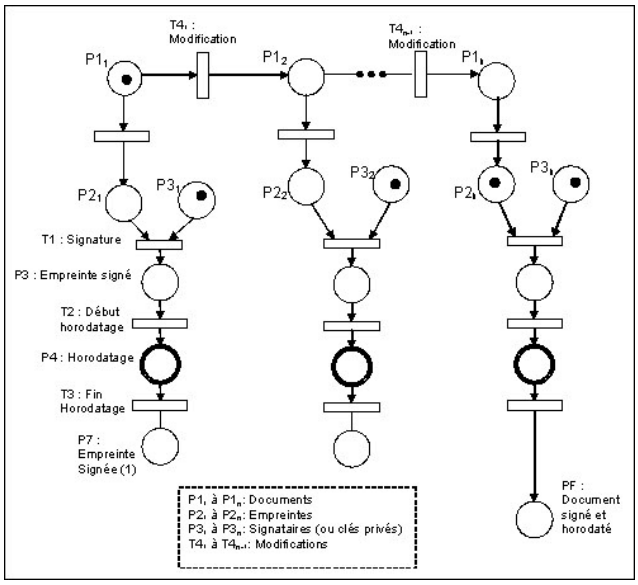

Figure 20. *Signature multiple enveloppante avec ajout et suppression d'information : Processus de signature*

La vérification de validité de ce document se déroule de la même manière que pour une signature simple (Figure 5.).

5. Conclusion

Après avoir décrit une signature électronique simple, nous avons explicité les signatures multiples que sont les co-signatures, sur-signatures, contre-signature, signature par parties et signatures enveloppantes avec ajout et/ou suppression d'information.
Nous avons également décris les processus associés à leur réalisation, que nous avons modélisés sous forme de Réseaux de Petri. Le formalisme des Réseaux de Petri utilisé dans cette étude nous permettra de faire ressortir les blocages, les étreintes fatales et d'une manière générale les performances du système de signature d'une organisation complexe, industrielle ou étatique. Cette étape pourra être la prochaine de notre étude.
Enfin, nous étudions les différents standards en vigueur (PKCS, signature au format XML) afin de vérifier les éventuelles restrictions qu'ils imposent à la mis en pratique de nos processus modélisés dans cet article.

6. Bibliographie

Callas J., Donnerhacke L., Finney H., Thayer R., “RFC 2440: OpenPGP Message Format”, Network Associates, IN-Root-CA Individual Network e.V., EIS Corporation, Novembre 1998

Cottin N., Sehili A., Wack M., “Time-stamping Electronic Documents and Signatures”, AICCSA’03: ACS/IEEE International Conference on Computer Systems and Applications, Tunis, Juillet 2003

Dobbertin H., Bosselaers A., Preneel B., “RIPEMD-160, a strengthened version of RIPEMD”, Fast Software Encryption, LNCS vol. 1039, D. Gollmann Ed., pp. 71-82, 1996

Elkins M., Del Torto D., Levien R., Roessler T., “Draft: MIME Security with OpenPGP”, Network Presence LLC., CryptoRights Foundation, University of California at Berkeley, Avril 2001

Farley J., *JAVA Distributed Computing*, O'Reilly and Associates, USA, ISBN 1-56592-206-9, Janvier 1998

Garfinkel S., *PGP: Pretty Good Privacy*, First Edition, O'Reilly, ISBN 1-56592-098-8, Décembre 1994

Garfinkel S., Spafford E. H., *Web Security & Commerce*, First Edition, O'Reilly, ISBN 1-56592-269-7, Juillet 1997

Housley R., Ford W., Polk W., Solo D., ‘”RFC 2459: Internet X.509 Public Key Infrastructure, Certificate and CRL Profile”, Spyrus, VeriSign and Citicorp, Janvier 1999

Kaeo M., *Designing Network Security*, Macmillan Technical Publishing, USA, ISBN 1-57870-043-4, 1999

Kaliski Jr B. S., “RFC 1319: The MD2 Message-Digest Algorithm”, RSA Laboratories, Janvier 1992

Menezes A. J., Van Oorschot P. C., Vanstone S. A., *Handbook of Applied Cryptography*, CRC Press, USA, ISBN 0-8493-8523-7, Février 2001

Murata T., *Petri nets: Properties, analysis and applications*, proc. of the IEEE, vol. 77, no 4, pp. 541-580, Avril 1989

Myers M., Ankney R., Malpani A., Galperin S., Adams C., “RFC 2560: X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol – OCSP”, VeriSign, CertCo, ValiCert, My CFO, Entrust Technologies, Juin 1999

Myers M., Ankney R., Adams C., Farrell S., Covey C., “Online Certificate Status Protocol, version 2”, *draft-ietf-pkix-ocspv2-02*, Mars 2001

NIST - National Institute of Standards and Technology, “Secure Hash Standard (SHS)”, Federal Information Processing Standards Publication, FIPS PUB 180-1, Avril 1995

NIST - National Institute of Standards and Technology, “Data Encryption Standard (DES)”, Federal Information Processing Standards Publication, FIPS PUB 46-3, Octobre 1999

NIST - National Institute of Standards and Technology, “Digital Signature Standard (DSS)”, Federal Information Processing Standards Publication, FIPS PUB 186-2, Janvier 2000

NIST - National Institute of Standards and Technology, “Common Format for Information that is Digitally Signed”, http://csrc.nist.gov/pki/signed_info_format/welcome.htm, Décembre 2000

Preneel B., Bosselaers A., Dobbertin H., “The cryptographic hash function RIPEMD-160”, *CryptoBytes*, vol. 3, No. 2, pp. 9-14, 1997

Ramsdell B., “RFC 2632: S/MIME Version 3 Certificate Handling”, Worldtalk, Juin 1999

Ramsdell B., “RFC 2633: S/MIME Version 3 Message Specification”, Worldtalk, Juin 1999

Rivest R. L., “RFC 1320: The MD4 Message-Digest Algorithm”, MIT Laboratory for Computer Science and RSA Data Security, Avril 1992

Rivest R.L., “RFC 1321: The MD5 Message-Digest Algorithm”, MIT Laboratory for Computer Science and RSA Data Security Inc., Avril 1992

RSA Data Security Inc., “Public Key Cryptography Standards, PKCS 1-12”, disponible en ligne à ftp://ftp.rsa.com/pub/pkcs, 1993

Shamir A., “How to share a secret”, Communications of the ACM 22 (11), pp 612-614, Novembre 1979

Sun Microsystems, “’Lesson: Signing and Verifying JAR Files”, available on-line at http://java.sun.com/docs/tutorial/jar/sign/index.html, 2001