Statut

Cette note définit un protocole expérimental pour la communauté d'Internet. Elle ne spécifie en aucune manière un standard Internet. Les discussions et les suggestions pour son amélioration sont les bien venues. La diffusion de cette note est libre.

sujet

Ce document décrit comment le protocole de S/MIME (Secure/Multipurpose Internet Mail Extensions) peut être traité et produit par un certain nombre de composants d'un système de messagerie, tels que des agents des transferts de message et des passerelles pour fournir des services de sécurité. Ces services désigné collectivement sous le nom de "services de sécurité de domaine" ("Domain Security Services" - DSS).

1. Introduction

Les séries de standards sur S/MIME définissent un format d'encapsulation des données fournissant d'un certain nombre de services de sécurité comprenant l'intégrité, la confidentialité, et l'authentification de données. S/MIME est conçu à l'usage des clients utilisant une messagerie fournissant des services de sécurité aux applications réparties de messagerie.
Les mécanismes décrits dans ce document sont conçus pour résoudre un certain nombre de problèmes d'interopérabilité et de limitations techniques qui surgissent quand les différents domaines de sécurité souhaitent communiquer de manière sécurisée, par exemple quand deux domaines emploient des technologies incompatibles de messageries telles que la série X.400 et le SMTP/mime, ou lorsqu'un simple domaine souhaite communiquer de manière sécurisée avec un de ses membres appartenant à un domaine qui n'est pas de confiance. Les scénarios de communications couverts par ce document sont : de domaine-à-domaine, d'individu-à-domaine et de domaine-à-individu. Ce document est également applicable pour les organismes et les entreprises qui ont leurs propres IGCs non accessibles par le monde extérieur, mais qui souhaitent être interopérable de manière sécurisée en utilisant le protocole de S/MIME.

La plus part du temps, il n'est pas souhaitable ou pratique de fournir des services de sécurité de bout en bout (poste à poste), en particulier entre différents domaines de sécurité. Une organisation qui se dit fournir des services de sécurité de bout en bout devra typiquement régler toute ou partie des problèmes suivants:

1) méthodes d'accès à des messageries hétérogènes : Les utilisateurs accèdent à leur courrier en utilisant les mécanismes (comme les browsers sur Internet) qui restructurent les messages. Le reformatage de message dans la mémoire de message rend le chiffrement de bout en bout et la validation de signature impossibles.

2) tri et audit des messages : les mécanismes, comme ceux des serveurs, tels que la recherche les mots interdits ou de tout autre contenu, la détection de virus, et l'audit, sont incompatibles avec le chiffrement de bout en bout.

3) problème du déploiement des IGCs : Il se peut qu'il n'y ai aucun chemin de certification entre deux organismes. Ou alors, une organisation peut être sensible à la préservation de son IGC et est peu disposée à les exposer au monde extérieur. En outre, le déploiement complet d'une IGC à tous les employés, peut s'avérer être coûteux, non nécessaire ou même ingérable pour de grands organismes. Pour n'importe laquelle de ces raisons, la validation de signature ou le chiffrement directs de bout en bout sont impossibles.

4) formats de messages hétérogènes : une organisation utilisant une messagerie de type X.400 souhaite communiquer avec une autre organisation qui utilise le SMTP. Le reformatage du message par la passerelle rend impossibles le chiffrement et la validation de signature de bout en bout.

Ce document décrit une approche pour résoudre ces problèmes en fournissant des services de sécurité de messagerie au niveau d'un domaine ou d'une organisation. Ce document spécifie comment ces 'services de sécurité de domaines' peuvent être fournis en utilisant le protocole de S/MIME. Les services de sécurité de domaines peuvent remplacer ou compléter des mécanismes au niveau du poste de travail. Par exemple, un domaine peut décider de fournir un service de signatures de poste à poste, mais un service de chiffrement de domaine-à-domaine. Ou bien, il peut permettre des services poste à poste pour un usage intra-domaine, et imposer des services au niveau du domaine pour la communication avec d'autres domaines.

Des services de domaines peuvent également être employés par les différents membres d'une société qui sont géographiquement distants et qui souhaitent échanger des messages chiffrés et/ou signés avec leur siège. Si bien même, un service du niveau d'un domaine est en soi meilleur ou plus mauvais que celui du niveau du poste de travail, ceci reste une question ouverte. Certains experts pensent que seule la solution de bout en bout reste la solution la plus sécurisée, alors que d'autres pensent que les avantages offert par des mécanismes comme la vérification du contenu du message aux frontières du domaine offre une augmentation considérable de sécurité pratique pour beaucoup de systèmes existants. Ce service supplémentaire permettant de vérifier la signature en plusieurs endroits sur un chemin de communications est également un avantage non négligeable dans beaucoup de situations. Cette discussion est en dehors du champ d'action de ce document. Ce qui est proposé ici est un ensemble d'outils que les administrateurs peuvent intégrer de différentes manières afin de satisfaire différents besoins dans des circonstances diverses.

Les agents des transferts de messages (MTAs), les gardes (filtres), les pare-feux et les passerelles multi protocoles fournissent des services de sécurité au niveau des domaines. Comme avec les solutions de sécurité basées au niveau des postes de sécurité, ces composants doivent être résistants contre tous types d'attaques prévues pouvant contrecarrer les services de sécurité. Par conséquent, une attention toute particulière doit être accordée à la sécurité de ces composants, pour s'assurer que leur mise en place et configuration réduit au minimum la possibilité d'une attaque.

2. Vue d'ensemble des services de sécurité de domaine

Cette section donne une vue d'ensemble informelle des services de sécurité qui sont fournis par S/MIME entre différents domaines de sécurité. Ces services sont fournis par une combinaison des mécanismes dans les domaines de l'expéditeur et du destinataire.

Les sections suivantes décrivent en définitif comment ces services s'intègrent dans des éléments du protocole de S/MIME.

Les mécanismes suivants de sécurité sont spécifiés dans ce document:

1.signature de domaine ("domain signature");
2.signature d'approbation("review signature");
3.Signature d'attributs ajoutés ("Additional Attributes Signature");
4.chiffrement et déchiffrement de domaine.

Les types de signatures, définis dans ce document, sont référencée comme signatures définies "DOMSEC".

Le terme 'domaine de sécurité ', utilisé dans ce document, est défini comme étant un ensemble de systèmes d'information et d'individus agissant pour le compte d'une seule autorité de sécurité et exécutant un travail commun. Les membres d'un domaine de sécurité doivent partager si nécessaire un degré élevé de confiance mutuelle, dû à leurs buts et objectifs partagés.

Un domaine de sécurité est typiquement protégé contre toute attaque directe (moyens physiques) ou indirecte (électronique - informatique) venant de l'extérieur, en combinant pare feux et filtres aux frontières de leur réseau. L'interface entre deux domaines de sécurité se nomme une 'frontière de sécurité '. Un exemple de domaine de sécurité peut être un réseau d'organisation type "Intranet".

2.1. signature de domaine

Une signature de domaine ("Domain Signature") est une signature de type S/MIME produite au nom d'un ensemble d'utilisateurs au sein d'un même domaine. Une signature de domaine peut être employée pour authentifier l'information envoyée entre domaines ou entre un domaine déterminé et un de ses membres, par exemples, deux 'Intranets' connectés entre eux via Internet, ou bien un Intranet connecté à un utilisateur distant (poste nomade) au sein d'Internet. Cette signature de domaine peut être employée lorsque deux domaines utilisent des schémas internes de signatures incompatibles ou lorsqu'il n'y a aucun lien de certification entre leur IGCs. Dans les deux cas, le message, venant du domaine de l'expéditeur, est signé par-dessus le message d'origine et la signature (si elle est présente) qui, elle, utilise un algorithme, une clef, et un certificat pouvant être traités par le destinataire(s) ou le domaine du destinataire(s). Une signature de domaine est parfois désignée sous le nom de "signature d'organisation".

2.2. signature d'approbation

Un tiers peut approuver des messages avant qu'ils soient envoyés au destinataire(s) final qui peut être ou non dans le même domaine de sécurité. Une politique d'organisation et la bonne pratique en matière de sécurité exigent souvent que les messages soient approuvés avant d'être envoyés aux destinataires externes. Après avoir approuvé un message, une signature de type S/MIME lui est ajoutée - une "signature d'approbation" ("Review Signature"). Un agent peut vérifier la signature d'approbation à la frontière du domaine, pour s'assurer que seuls les messages approuvés soient transférés.

2.3. signature d'attributs ajoutés

Un tiers peut ajouter des attributs supplémentaires à un message signé. Une signature de type S/MIME est employée à cet effet - une "signature d'attributs ajoutés" ("Additional Attributes Signature"). Un exemple d'attributs ajoutés peut être l'"étiquette d'équivalence" ("Equivalent Label"), défini dans l'ESS.

2.4. chiffrement et déchiffrement de domaine

Le chiffrement de domaine est celui de type S/MIME effectué au nom d'un ensemble d'utilisateurs d'un même domaine. Le chiffrement de domaine peut être employé pour protéger l'information entre les domaines, par exemple, quand deux "Intranets" sont reliés entre eux via Internet. Il peut également être employé lorsque les utilisateurs n'ont pas de capacité de chiffrement/IGC sur leur poste de travail, ou lorsque deux domaines utilisent des moyens de chiffrement internes incompatibles. Dans ce dernier cas, les messages venant du domaine de l'expéditeur sont chiffrés (ou sur-chiffrés) en utilisant un algorithme, une clef, et un certificat puis sont déchiffrés par le destinataire(s) ou par une entité de leur domaine. Cet processus s'applique également pour protéger l'information entre un domaine particulier et un de ses membres quand tous les deux sont connectés via un réseaux dit "non de confiance", comme Internet.

3. Traçabilité des services de signature du protocole S/MIME

Cette section décrit les éléments du protocole S/MIME qui sont employés pour fournir les services de sécurité décrits ci-dessus. ESS [Enhanced Security Services RFC 2634 - Service de sécurité réhaussés] présente le concept des messages triple-enveloppés qui sont d'abord signés, chiffrés, puis signé à nouveau. Ce document utilise également ce concept de la triple-enveloppe, mais également celui de l'"encapsulation de signature". L'"encapsulation de signature" dénote un message signé ou non qui est enveloppé dans une signature, cette signature recouvre à la fois le contenu et la première signature (intérieure), si elle est présente.

L'encapsulation de signature peut être exécutée sur la signature intérieure et/ou externe d'un message triple-enveloppé.

Par exemple, l'expéditeur signe un message qui est alors encapsulé avec une signature "attributs ajoutés". Il est ensuite chiffré. Un "approuveur" signe alors ces données chiffrées, qui sont alors encapsulées par une signature de domaine.

Il est possible que certaines politiques exigent que l'ajout de signatures se fassent dans un ordre spécifique. Par la simple autorisation d'ajout de signatures par encapsulation, il est possible de déterminer l'ordre dans lequel ces signatures ont été ajoutées.

Une signature définie par "DOMSEC" - "signature DOMSEC" peut encapsuler un message dans l'une des possibilités suivantes:

1) A un message non signé lui est ajouté une couche vide de signature (c.-à-d., le message est enveloppé dans un signedData qui a des signerInfos ne contenant pas d'élément). Ceci doit permettre la compatibilité en retour avec un logiciel de S/MIME ne possédant pas les capacité de "DOMSEC". Si les signerInfos ne contiennent pas d'élément sur le signataire, l'eContentType, dans l'EncapsulatedContentInfo, doit être l'id-data comme décrit dans la CMS [Cryptographic Message Syntax]. Cependant, le champ eContent contiendra le message non signé au lieu d'être laissé vide comme suggéré dans la section 5.2 de la CMS. Ainsi, lorsque la "signature DOMSEC" est ajoutée, comme définie dans la méthode 2) ci-dessous, la signature recouvrira le message non signé.

2) l'encapsulation de signature est employée pour envelopper le message original et signé avec une "signature DOMSEC". Ainsi, la "signature DOMSEC" recouvre le message et toutes ses signatures précédemment ajoutées. En outre, il est possible de déterminer si la "signature DOMSEC" a bien été ajoutée après les signatures déjà présentes.

3.1. conventions d'appellation (de nommage) et types de signature

Une entité recevant un message signé par S/MIME s'attendrait à ce que normalement la signature soit celle du créateur du message. Cependant, les services de sécurité de messages définis dans ce document exigent du destinataire de pouvoir accepter des messages signés par d'autre entités et/ou du créateur. Lorsque d'autres entités signent le message, le nom inscrit dans le certificat ne correspondra pas à celui de l'expéditeur du message. Une exécution conforme de S/MIME marquerait normalement un avertissement s'il y avait une disparité entre le nom dans le certificat et le nom de l'expéditeur du message. Ce contrôle empêche un certain nombre de types d'attaques par masquerade.

Dans le cas des services de sécurité de domaine, cette condition d'avertissement devrait être supprimée dans certaines circonstances. Ces circonstances sont définies par une convention d'appellation qui indique la forme à la laquelle le nom du signataires devrait adhérer. L'adhésion à cette convention d'appellation évite les problèmes d'appellation non contrôlée et les attaques possibles par masquerade que celle-ci produirait.

En tant qu'aide à l'exécution, un attribut signé est défini pour être inclus dans la signature S/MIME - l'attribut "type de signature". À la réception d'un message contenant cet attribut, les contrôles sur la convention d'appellation sont effectués.

Les implémentations se conformant à ce standart doivent:

- prendre en charge la convention d'appellation pour la génération et la vérification de signature;
- identifier l'attribut du type de signature pour la vérification de signature;
- prendre en charge l'attribut du type de signature pour la génération de signature.

3.1.1. Conventions d'appellation

Les conventions d'appellation ("Naming Conventions") suivantes sont spécifiques pour des agents produisant des signatures indiquées dans ce document:

* pour une signature de domaine, un agent produisant cette signature doit être appelé "domaine-signer-autorité";
* pour une signature d'approbation, un agent produisant cette signature doit être appelé "review-authority";
* Pour une signature d'attributs ajoutés, un agent produisant cette signature doit être appelé "attribute-authority".

Ce nom devra apparaître dans le composant "common name (CN)" dans le champ "subject" du certificat X.509. Il ne doit y avoir qu'un seul CN présent. En plus, si le certificat contient une adresse de type RFC 822, ce nom doit apparaître dans l'extrémité gauche de l'adresse (avant le symbole "@").

Dans le cas d'une signature de domaine, une règle supplémentaire d'appellation est définie: la "name mapping rule". Cette règle de mappage du nom stipule que pour une autorité de signature de domaine, la composante de son nom où est indiqué son nom de domaine - "domain part" doit être identique, ou ascendant, à celui du nom de domaine du créateur du message, qu'il représente.
Le "domain part" est définie comme suit:

* dans le cas d'un nom distingué au format X.500 d'un certificat X.509, le "domain part" correspond aux composants "pays", "organisation", "unité de l'organisation", "état", et "localité" du nom distingué;
* dans le cas d'un nom distingué au format RFC 2247, le "domain part" sera la composante "domain" du nom distingué.
* si le certificat contient une adresse au format RFC 822, le "domain part" correspond à la composante à droite du symbole "@" dans l'adresse RFC 822.

Par exemple, une autorité de signature de domaine agissant pour le compte de:
John Doe de la société "Acme corporation", dont le nom distingué est:
"cn=John Doe, ou=marketing, o=acme, c=us"
et dont l'adresse E-mail est:
"John.Doe@marketing.acme.com"
pourrait avoir un certificat contenant un nom distingué de la forme:
"cn=domain-signing-authority, o=acme, c=us"
et d'une adresse de type RFC 822
"domain-signing-authority@acme.com".
Si John Doe a une adresse de type RFC 2247 de la forme:
"cn=John Doe, dc=marketing, dc=acme, dc=us"
ainsi l'adresse suivante:

"cn=domain-signing-authority, dc=acme, dc=us"
pourrait être employée pour représenter l'autorité de signature de domaine.

Lorsqu'un nom distingué, au format X.500 possède des champs consécutifs de plusieurs unités organisationnelles et/ou de localités, il est important de prendre en compte l'ordre de ces valeurs afin de déterminer si le "domain part" de la signature de domaine est ascendante. Dans ce cas, en analysant le nom distingué à partir du composant le plus significatif (c.-à-d., pays, localité ou organisation) l'unité organisationnelle ou la localité analysée est considérée comme étant l'ascendant (non analysé) des unités organisationnelle ou des localités consécutives.

En analysant un nom de type RFC 2247 depuis le composant le plus significatif (c.-à-d., le "dc" entré qui représente le pays, la localité ou l'organisation) l'entrée "dc" analysée est considérée pour être l'ascendant des entrées "dc" (non analysée) consécutives.

Par exemple, une autorité de signature de domaine agissant pour le compte de:
John Doe de la société "Acme corporation", dont le nom distingué est:
"cn=John Doe, ou=marketing, ou=defence, o=acme, c=us"
et dont l'adresse de E-mail est:
"John.Doe@marketing.defence.acme.com",
pourrait avoir un certificat contenant un nom distingué de la forme
"cn=domain-signing-authority, ou=defence, o=acme, c=us"
et d'une adresse de type RFC 822 de la forme:
"domain-signing-authority@defence.acme.com".
Si John Doe a une adresse de type RFC 2247 de la forme:
"cn=John Doe, dc=marketing, dc=defense, dc=acme, dc=us"
alors l'autorité de signature du domaine pourrait avoir un nom distingué de la forme:
"cn=domain-signing-authority, dc=defence, dc=acme, dc=us".

N'importe quel message reçu doit être marqué ("flag") dans le cas où le "domain part" du nom de l'agent (autorité de signature du domaine) ne correspondrait pas ou n'e serait pas l'ascendant de celui du domaine du créateur du message.

Cette règle d'appellation empêche la masquerade pour des agents d'une organisation qui veulent se faire passer pour autorités de signature de domaine à la place d'un autre. Pour les autres types de signature définies dans ce document, aucune règle, concernant le mappage du nom, n'est défini.

Les implémentations se conformant à ce standard doivent prendre en compte au minimum cette convention de mappage du nom. Les implémentations peuvent compléter cette convention avec d'autres conventions définies localement. Cependant, elles doivent être convenus entre les domaines de l'expéditeur et du destinataire avant de sécuriser l'échange de messages.

À la vérification de la signature, un agent de "réception" doit s'assurer que la convention d'appellation a été respectée. N'importe quel message qui viole la convention doit être marqué.

3.1.2. Attribut De Type De Signature

Un attribut signé de type S/MIME ("Signature Type Attribute") est utilisé pour indiquer le type de signature. Cette utilisation devrait être employé en même temps que celle des conventions d'appellation indiquées dans la section précédente. Lorsqu'un message signé de type S/MIME contenant le type d'attribut de signature est reçu, il déclenche l'application de vérification de la convention d'appellation pour voir si elle est correctement employée.

La notations ASN.1 de cet attribut est:

```
SignatureType ::= SEQUENCE OF OBJECT IDENTIFIER

id-sti  OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840)
          rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) 9 }

-- signature type identifier
```

Si il est présent, l'attribut "SignatureType" doit être un attribut signé, comme il doit être défini dans la syntaxe CMS ("Cryptographic Message Syntax", RFC 2630). Si l'attribut "SignatureType" est absent et s'il n'y plus aucune signatures encapsulées, alors le destinataire devrait supposer que la signature est bien celle du créateur de message.

Toutes les signatures définies ici sont produites et traitées comme décrit dans la CMS. Elles sont distinguées par la présence des valeurs suivantes dans l'attribut signé "SignatureType":

```
id-sti-domainSig OBJECT IDENTIFIER ::= { id-sti 2 }
-- domain signature.

id-sti-addAttribSig OBJECT IDENTIFIER ::= { id-sti 3 }
-- additional attributes signature.

id-sti-reviewSig OBJECT IDENTIFIER ::= { id-sti 4 }
-- review signature.
```

En complément, un type d'attribut est également spécifié pour une signature du créateur de message. Cependant, ce type de signature est facultatif. Il est défini comme suit:

```
id-sti-originatorSig OBJECT IDENTIFIER ::= { id-sti 1 }
-- originator's signature.
```

Tous les types de signature, à l'exception de celui du créateur du message, doivent encapsuler les autres signatures.
Notez qu'une "signature DOMSEC" pourrait encapsuler une signature vide comme cela est défini dans la section 3.

Un "SignerInfo" NE doit PAS inclure des exemples multiples de "SignatureType". Un attribut signé représentant un "SignatureType" peut inclure des exemples multiples de différentes valeurs de "SignatureType" comme "AttributeValue" de "attrValues" [ref CMS], ceci tant que le "SignatureType" correspondant à "additional attributes" n'est pas présent.

S'il y a plus d'un "SignerInfo" dans un "signerInfos" (c.-à-d., lorsque différents algorithmes sont employés), alors, l'attribut "SignatureType" dans tout les "SignerInfos" doit contenir le même contenu.

Les sections suivantes décrivent les conditions dans lesquelles chacun de ces types de signature peut être généré, et comment il est traité.

3.2. Génération et vérification de signature de domaine

Une "signature de domaine" ("Domain Signature") est une "proxy" signature générée par une entité particulière au nom de tous les utilisateurs du domaine. La signature doit adhérer à la convention d'appellation (paragraphe 3.1.1) incluant la convention de mappage du nom. Une "signature de domaine" d'un message autentifie le fait que ce message a bien été traité par ce domaine. Avant la signature, un processus, générant un "domaine de signature", doit avant tout, se satisfaire de l'autenticité du message de l'expéditeur. Ceci s'éffectue par une des deux méthodes suivantes:
- soit la signature du créateur du message est vérifiée, si les signature S/MIME sont utilisées au sein du domaine;
- sinon, des mécanismes externes à S/MIME sont utilisés, comme l'adresse physique du client d'origine ou bien un lien IP autentifié.

Si l'authenticité du créateur du message est vérifiée avec succès, par l'une des deux méthodes ci-dessus, et si toutes les autres signatures présentes sont valides, incluant celles qui ont été chiffrées, alors un "domaine de signature" peut être ajouté au message.

Si un domaine de signature est ajouté au message et si ce nouveau message est reçu par un agent de traitement de liste (MLA - Mail List Agent), il se peut que le "domaine de signature" soit retiré. Pour empêcher que le 'domaine de signature" soit retiré l'étape mentionnée dans la section 5 doit être appliquée.

Une entité générant un "domaine de signature" doit le faire en utilisant un certificat dont le sujet du nom doit suivre la convention d'appellation spécifiée dans le paragraphe 3.1.1.

Si l'authenticité du crateur du message n'est pas vérifiée avec succès, ou bien si toutes les signatures présentes ne sont pas valides, un "domaine de signature ne doit pas être généré.

A la réception, le "domaine de signature" devrait être utilisé pour vérifier l'authentification du message. Une vérification doit être faite pour s'assurer qu'aussi bien la convention d'appellation que celle du mappage du nom, ont bien été utilisées, comme spécifiées dans ce standard.

Un destinataire peut assumer le fait que la vérification avec succès du "domaine de signature" permet également d'authentifier le message original.

Si la signature du créateur du message est présente, le nom dans ce certificat pourait être utilisé pour identifier l'expéditeur, créateur du message. Cette information peut alors être diffusée au destinataire.

Si la signature du créateur du message n'est pas présente, la seule supposition qui peut être faite est de savoir de quel domaine le message provient.

Un signataire d'un domaine peut supposer avoir vérifié n'importe quelles signatures qu'il encapsule. Donc, il n'est pas nécessaire de vérifier ces signatures avant de considérer le message comme authentique. Cependant, ce standard ne doit empêcher un destinataire à pouvoir vérifier n'importe quelles autres signatures présentes.

La "signature de domaine" est indiquée par la présence de la valeur "id-sti-domainSig" dans l'attribut signé "signature type".

Il peut y avoir une ou plusieurs signatures du "domaine de signature" dans un encodage S/MIME.

3.3. Génération et vérification de signatures d'attributs ajoutés

Le type de signature "attributs ajoutés" ("Additional Attributes Signature") indique que "SignerInfo" contient des attributs ajoutés qui sont associés au message.

Tous les attributs dans le "SignerInfo" applicable doivent être traités comme des attributs ajoutés. La vérification avec succès d'une signature "attributs ajoutés" signifie seulement que les attributs sont authentiquement lié au message. Un destinataire ne doit pas supposer que cette vérification avec succès permet d'authentifier également le message original.

Une entité générant une signature "attributs ajoutés" doit le faire en utilisant un certificat dont le sujet du nom doit suivre la convention d'appellation spécifiée dans le paragraphe 3.1.1. A la réception, une vérification doit être faite pour s'assurer que la convention d'appellation a bien été utilisée.

Au moment de la génération d'une signature "attributs ajoutés", un signataire peut inclure n'importe quel attribut listé dans l'ESS ("Enhanced Security Services for S/MIME" - Services de Sécurité Renforcés, RFC 2634) ou listé dans ce document. Les attributs suivants ont une signification particulière, lorsqu'ils sont présents dans une signature "attributs ajoutés":

1) Etiquette Equivalente ("Equivalent Label"): les valeurs de cette étiquette dans cet attribut, doivent être considérées comme équivalentes à celles de l'étiquette de sécurité contenues dans le "SignerInfo" encapsulé, si il est présent.

2) Etiquette de Sécurité ("Security Label"): la valeur de cette étiquette indique la confidentialité globale du contenu du message interne (original) avec en plus des "signedData" et container "envelopedData", encapsulés. L'étiquette sur les données originales est indiquée par la valeur de la signature du créateur du message, si elle est présente.

Une signature "attributs ajoutés" est indiquée par la présence de la valeur "id-sti-addAttribSig" dans l'attribut signé "signature type". Les autres idantifiants d'objets (Object Identifiers - OIDs) ne doivent pas être inclus dans la séquence du OIDs si cette valeur est présente.

Il peut y avoir de multiples signatures "attributs ajoutés" dans un encodage S/MIME.

3.4. Génération et vérification de signature d'approbation

La signature d'approbation ("review signature") indique que le signataire a viser le message. Une vérification avec succès d'une signature d'approbation signifie seulement que le signataire confirme que le message peut être envoyé désormais à son destinataire.
Si le destinataire est d'un autre domaine, un mécanisme au niveau de la frontière du domaine, comme un pare-feux peut être configuré pour vérifier les signatures d'approbation. Un destinataire ne doit pas supposer que la vérification avec succès de cette signature permette d'authentifier également le message original.

Une entité générant une signature d'approbation doit le faire en utilisant un certificat dont le sujet du nom doit suivre la convention d'appellation spécifiée dans le paragraphe 3.1.1. A la réception, une vérificationdoit être faite pour s'assurer que la convention d'appellation a bien été utilisée.

Une signature d'approbation est indiquée par la présence de la valeur "id-sti-reviewSig" dans l'attribut signé "signature type".

Il peut y avoir de multiples signatures d'approbation dans un encodage S/MIME.

3.5. Signature du créateur du message

La signature du créateur du message ("Originator signature") est utilisée pour indiquer que le signataire est bien le créateur du message et de ses contenus. Elle est incluse dans ce document en complément seulement. Une signature du créateur du message est indiquée aussi par l'absence de "signature type attribute", ou par la présence de la valeur "id-sti-originatorSig" dans l'attribut signé "signature type".

4. Chiffrement et déchiffrement

Le chiffrement de message peut être effectué par un tiers au nom d'un ensemble d'utilisateurs au sein d'un même domaine. Ce processus fait référence au domaine de chiffrement ("domain encryption"). Le déchiffrement de message peut être exécuté par un tiers au nom d'un ensemble de destinataires d'un même domaine. Ce processus fait référence au domaine de déchiffrement ("domain decryption"). Le tiers qui exécute ces processus est désigné dans cette section sous le nom d'une "autorité de confience de domaine" (ACDomom) ("Domain Confidentiality Authority" - DCA). Ces deux processus sont décrits dans cette section.

Des messages peuvent être chiffrés pour le déchiffrement par le destinataire final et/ou par un ACDomom du domaine du destinataire. Le message peut également être chiffré pour le déchiffrement par un ACDomom du domaine du créateur du message (par exemple, pour

l'analyse du contenu, l'audit, la recherche de mot clé, etc.). Le choix de la mise en place d'un de ces processus dépend de la politique de sécurité du système à mettre en oeuvre. Il est donc en dehors de la portée de ce document. Ces processus de chiffrement et de déchiffrement sont montrés dans le tableau suivant:

	Déchiffrement par le destinataire	Déchiffrement par l'ACDom
Chiffrement par le créateur	Cas (a)	Cas (b)
Chiffrement par l'ACDom	Cas (c)	Cas (d)

Le cas (a), chiffrement des messages par le créateur pour le déchiffrement par le destinataire(s) final, est décrit dans la CMS.
Dans les cas (c) et (d), le chiffrement n'est pas effectué par le créateur mais par l'ACDom du domaine du créateur. Dans les cas (b) et (d), le déchiffrement n'est pas exécuté par le destinataire(s) mais par l'ACDom du domaine du destinataire.

Une implémentation client qui se conforme à ce standard doit soutenir le cas (b) pour l'envoi, le cas (c) pour la réception et le cas (a) pour l'envoi et la réception.

Une implémentation ACDom qui se conforme à ce standard doit soutenir le cas (c) et (d), pour l'envoi, et les cas (b) et (d) pour la réception. Dans les cas (c) et (d) la "signature de domaine" devrait être appliquée avant le chiffrement. Dans les cas (b) et (d) le message devrait être déchiffré avant l'obtention et la vérification de la "signature de domaine".

Le processus du chiffrement et du déchiffrement est documenté dans la CMS. La seule condition supplémentaire introduite par le chiffrement et déchiffrement de domaine est d'avoir une plus grande flexibilité dans la gestion des clefs (voir sous-sections suivantes). Comme pour les signatures, une convention d'appellation et une convention de mappage du nom sont utilisées pour localiser la clef publique adéquate.

Les mécanismes décrits ci-dessous sont applicables à la fois pour les systèmes d'accord de clefs et de transport de clefs (documentation dans la CMS). L'expression "clef de chiffrement" est employée comme terme générique pour couvrir la gestion des clefs utilisée par les deux techniques.

Les mécanismes ci-dessous sont également applicables pour les utilisateurs nomades qui souhaitent chiffrer les messages qui sont envoyés vers leur base.

4.1. Conventions d'appellation des domaines de confidentialités

Un ACDom doit être appelé "domain-confidentiality-authority". Ce nom devra apparaître dans le composant "common name (CN)" dans le champ "subject" du certificat X.509. Il ne doit y avoir qu'un seul CN présent. En plus, si le certificat contient une adresse de type RFC 822, ce nom doit apparaître dans l'extrémité gauche de l'adresse (avant le symbole "@").

Dans le cas d'une convention d'appellation, une règle supplémentaire d'appellation est définie: la "name mapping rule". Cette règle de mappage du nom stipule que pour une ACDom, son "domain part" doit être identique, ou ascendant (comme défini dans la section 3.1.1), à celui du nom de domaine de l'ensemble des entités qu'il représente. Le "domain part" est définie comme suit:

* dans le cas d'un nom distingué au format X.500 d'un certificat X.509, le "domain part" correspond aux composants "pays", "organisation", "unité de l'organisation", "état", et "localité" du nom distingué;

* dans le cas d'un nom distingué au format RFC 2247, le "domain part" sera la composante "domain" du nom distingué.

* si le certificat contient une adresse au format RFC 822, le "domain part" correspond à la composante à droite du symbole "@" dans l'adresse RFC 822.

Par exemple, une ACDom agissant pour le compte de:
John Doe de la société "Acme corporation", dont le nom distingué est:
"cn=John Doe, ou=marketing, o=acme, c=us"
et dont l'adresse E-mail est:
"John.Doe@marketing.acme.com"
pourrait avoir un certificat contenant un nom distingué de la forme:
"cn=domain-confidentiality-authority, o=acme, c=us"
et d'une adresse de type RFC 822
"domain-confidentiality-authority@acme.com".
Si John Doe a une adresse de type RFC 2247 de la forme:
"cn=John Doe, dc=marketing, dc=defense, dc=acme, dc=us"
ainsi l'adresse suivante:
"cn=domain-signing-authority, dc=defense, dc=acme, dc=us"
pourrait être employée pour représenter l'autorité de signature de domaine.
La clef associée à ce certificat pourrait être utilisé pour chiffrer les messages pour John Doe.

N'importe quel message reçu doit être marqué ("flag") dans le cas où le "domain part" du nom de l'agent de chiffrement du domaine ne correspondrait pas ou ne serait pas l'ascendant de celui de l'entité qu'il représente.

Cette règle d'appellation empêche les messages à être chiffrés par un mauvais agent de chiffrement de domaine.

Les implémentations se conformant à ce standard doivent prendre en compte au minimum cette convention de mappage du nom. Les implémentations peuvent compléter cette convention avec d'autres conventions définies localement. Cependant, elles doivent être convenus par un accord entre les domaines de l'expéditeur et du destinataire avant l'échange de messages.

4.2. Gestion des clefs pour le chiffrement par l'ACDom

Au niveau du domaine de l'expéditeur, le chiffrement par l'ACDom est réalisé en utilisant soit le certificat de l'ACDom destinataire, soit le certificat du destinataire final. Pour réaliser ceci, le processus de chiffrement doit pouvoir localiser correctement soit le certificat de l'ACDom correspondant au domaine auquel le destinataire appartient, soit celui du destinataire final. Après avoir localisé le certificat adéquat, le procédé de chiffrement est alors exécuté et l'information supplémentaire exigée pour le déchiffrement est envoyée au destinataire dans le champ "recipientInfo" comme indiqué dans la CMS. La dénomination d'un agent de chiffrement ACDom doit suivre la convention d'appellation spécifée dans la section 4.1., de sorte que le certificat correspondant puisse être trouvé.

Aucune méthode spécifique n'est exigée dans ce document pour localiser soit le certificat de l'ACDom correspondant au domaine dans lequel le destinataire appartient, soit celui du destinataire final. Une implémentation peut choisir d'accéder à un certificat local enregistré pour localiser le certificat adéquat. Alternativement, un annuaire X.500 ou LDAP peut être employé dans un des cas suivants:

1. L'annuaire peut stocker le certificat de ACDom dans le répertoire du destinataire. Lorsque le type de certificat à utiliser est demandé, ce certificat est retourné.

2. L'agent de chiffrement retranscrit le nom du destinataire vers le nom de l'ACDom de la façon indiquée dans la section 4.1. Le type de certificat à utiliser, associé à ce répertoire est alors obtenu.

Ce document ne relève pas non plus de ces processus. Quelque soit l'un des certificats employé, la convention de mappage du nom doit être respectée, afin de maintenir la confidentialité.

Après avoir localisé le certificat adéquat, le procédé de chiffrement est alors exécuté. Comme il l'est décrit dans la CMS, un "recipientInfo" est alors généré pour l'ACDom ou le destinataire final.

Le chiffrement par l'ACDom peut être effectué pour le déchiffrement par un destinataire final et/ou une ACDom. Le déchiffrement par le destinataire final est décrit dans la CMS. Le déchiffrement par l'ACDom est décrit dans la section 4.3.

4.3. Gestion des clefs pour le déchiffrement par l'ACDom

Le déchiffrement par l'ACDom s'effectue à partir d'une clef-privée appartenant à l'ACDom et à partir de l'information nécessaire transmise par le champ "recipientInfo" de l'ACDom.

Il convient noter que le déchiffrement de domaine peut être exécuté sur des messages chiffrés par le créateur et/ou par une ACDom du domaine du créateur. Dans le premier cas, le procédé de chiffrement est décrit dans la CMS; dans le deuxième cas, le procédé de chiffrement est décrit dans la section 4.2.

5. Mise en oeuvre d'une signature de domaine lorsque les agents de liste de courrier sont présents.

Il est possible qu'un message partant d'un domaine "DOMSEC" puisse rencontrer un agent de liste de courrier ("Mail List Agent" - MLA) avant qu'il atteigne le destinataire final. Dans ce cas, il se peut que la "signature de domaine" soit retirée du message. Ce que nous ne voulons pas. Par conséquent, la "signature de domaine" doit être appliquée au message de telle manière qu'elle empêche un MLA de l'enlever.

Un MLA recherchera, dans un message, la couche 'externe' de "signedData", comme elle est définie dans la section 4.2 de l'ESS, et "décolle" (désencapsule) toutes les couches de "signedData" qui encapsulent cette couche externe de "signedData". La détection de cette couche externe de "signedData" dépendra de la manière où le message contient un attribut "mlExpansionHistory" ou une couche "envelopedData".

Il se peut qu'un message partant d'un domaine "DOMSEC" ait été déjà traité par un MLA, dans ce cas un attribut "mlExpansionHistory" sera présent dans le message.

Il se peut que le message contienne une couche "envelopedData". C'est le cas lorsque le message a été chiffré dans le domaine par l'ACDom du domaine concerné, voir section 4,0, et, probablement, par le destinataire final.

La manière dont la "signature de domaine" est appliquée dépendra de ce qui est déjà présent dans le message. Avant que la "signature de domaine" puisse être appliquée, le message doit être recherché à partir de la couche externe de "signedData", cette recherche est complète lorsqu'un des éléments suivant est trouvé:
- la couche externe de "signedData" qui inclut un attribut "mlExpansionHistory" ou qui encapsule un objet "envelopedData"; - une couche "envelopedData";
- le contenu original (c'est-à-dire, une couche qui n'est ni "envelopedData" ni "signedData").

Si une couche de "signedData" contenant un attribut "mlExpansionHistory" a été trouvée alors:

1) décollage (désencapsulation) la couche de "signedData" (après s'être rappelé les "signedAttributes" inclus).

2) recherche du reste du message jusqu'à qu'une couche "envelopedData" ou que le contenu original soit trouvé.

3) a) si une couche "envelopedData" a été trouvée alors:

- décollage (désencapsulation), vers le bas, de toutes les couches "signedData" jusqu'à la couche "envelopedData".
- localisation du "RecipientInfo" pour l'ACDom local et utilisation de l'information qu'elle contient pour obtenir la clef de message.
- déchiffrement du "encryptedContent" en utilisant la clef de message.
- encapsulation du message déchiffré avec une "signature de domaine"
- si la politique locale exige du message d'être chiffré en utilisant le chiffrement S/MIME avant de laisser le domaine alors encapsuler la "signature de domaine" avec une couche

"envelopedData" contenant des structures "RecipientInfo" pour chacun des destinataires et une valeur "originatorInfo" construite à partir de l'information décrivant cet ACDom.

Si la politique locale n'exige pas du message d'être chiffré en utilisant le chiffrement S/MIME, il y a, tout de même, un "envelopedData" à un niveau plus bas dans le message où la "signature de domaine" doit être encapsulée par un "envelopedData" comme cela est décrit ci-dessus.

Un exemple, lorsque ce n'est pas une politique locale qui exige le chiffrement S/MIME, c'est qu'un lien "cryto" est présent.

b) Si une couche "envelopedData" n'a pas été trouvée alors:

- encapsulation du nouveau message avec une "signature de domaine".

4) encapsulation du nouveau message dans une couche "signedData", ajoutant les "signedAttributes" (à partir) de la couche de "signedData" qui contient l'attribut "mlExpansionHistory".

Si aucune couche "signedData" contenant un attribut "mlExpansionHistory" n'a été trouvée mais si un "envelopedData" a été trouvé alors:

1) décollement, vers le bas, de toutes les couches "signedData" jusqu'à la couche "envelopedData";

2) localisation du "RecipientInfo" pour l'ACDom local et utilisation de l'information qu'il contient pour obtenir la clef de message;

3) déchiffrement du "encryptedContent" en utilisant la clef de message;

4) encapsulation du message déchiffré avec une "signature de domaine";

5) si la politique locale exige du message d'être chiffré avant de quitter le domaine alors encapsulation de la "signature de domaine" avec une couche "envelopedData" contenant les structures "RecipientInfo" pour chacun des destinataires et une valeur "originatorInfo" construite à partir de l'information décrivant cet ACDom.

Si la politique locale n'exige pas du message d'être chiffré en utilisant le chiffrement de S/MIME, il y a, tout de même, un "envelopedData" à un niveau plus bas dans le message où la "signature de domaine" doit être encapsulée par un "envelopedData" comme cela est décrit ci-dessus.

Si aucune couche "signedData" contenant un attribut "mlExpansionHistory" n'a été trouvée et aucun "envelopedData" n'a été trouvé alors:

1) encapsulation du message dans une "signature de domaine".

5.1. Exemples de traitement des règles

Les exemples suivants aident à expliquer les règles ci-dessus. Tous les objets "signedData" sont valides et aucun d'entre eux ne sont des "signatures de domaine". Si un objet "signedData" était une "signature de domaine" alors il ne serait plus nécessaire de valider tout autre objet "signedData".

1) Application d'une "signature de domaine" à un message signé (S1(contenu_original)) (où S = "signedData")
Le "signedData", S1, n'inclut pas d'attribut "mlExpansionHistory".
- vérification du "signedData", S1;
- Après recherche comme définie ci-dessus, aucun "signedData" externe n'est trouvé;
- puisque le contenu original est trouvé, ni "envelopedData" ni un attribut "mlExpansionHistory" n'est trouvé.
- création d'une nouvelle couche de "signedData", S2, qui contient une "signature de domaine", ayant pour résultat le message (S2(S1(contenu_original))) envoyé par le domaine.

2) Application d'une "signature de domaine" à un message signé à plusieurs niveaux (S3(S2(S1(contenu_original)))
Aucun des "signedData", S1, S2 ou S3, n'inclut d'attribut "mlExpansionHistory".
- vérification des objets "signedData" S1, S2 et S3.
- la couche "signedData" externe n'est pas celle de l'origine puisque jusqu'à la découverte du contenu original, ni "envelopedData" ni attribut "mlExpansionHistory" n'a été trouvé.
- création d'une nouvelle couche de "signedData", S4, qui contient une "signature de domaine", ayant pour résultat le message (S4(S3(S2(S1(contenu_original))) envoyé par le domaine.

3) Application d'une "signature de domaine" à un message signé et chiffré (E1(S1(contenu_original))) (où E = "envelopedData")
Le "signedData", S1, n'inclut pas d'attribut "mlExpansionHistory".
- la couche "signedData" externe n'est pas celle de l'origine puisque l'"envelopedData", E1, est détectée à la couche externe;
- déchiffrement du "encryptedContent";
- vérification du "signedData", S1;
- enveloppement du contenu déchiffré dans une nouvelle couche "signedData", S2, qui contient une "signature de domaine".
--Si la politique locale exige du message d'être chiffré en utilisant le chiffrement S/MIME, avant de quitter le domaine, alors ce nouveau message est enveloppé dans une couche "envelopedData", E2, ayant pour résultat le message (E2(S2(S1(contenu_original)))) envoyé par le domaine;
--sinon le message n'est pas enveloppé dans une couche "envelopedData" ayant pour résultat le message (S2(S1(contenu_original))) envoyé par le domaine.

4) Application d'une "signature de domaine" à un message signé, chiffré et resigné (S2(E1(S1(contenu_original))))
Le "signedData", S2, inclut un attribut "mlExpansionHistory".
- vérification du "signedData", S2;
- détection de l'attribut "mlExpansionHistory" dans S2, ainsi S2 est le "signedData" externe.
- mise en mémoire des attributs signés de S2 pour une inclusion postérieure dans le nouveau "signedData" externe à appliquer au nouveau message;
- décollement de S2;

- déchiffrement du message;
- vérification de l'objet "signedData", S1;
- enveloppement du message déchiffré dans une couche "signedData", S3, qui contient une "signature de domaine";
--Si la politique locale exige du message d'être chiffré en utilisant le chiffrement S/MIME, avant de quitter le domaine, alors ce nouveau message est enveloppé dans une couche "envelopedData", E2. Une nouvelle couche de "signedData", S4, est alors enroulée autour de l'"envelopedData", E2, ayant pour résultat le message (S4(E2(S3(S1(contenu_original))))) envoyé par le domaine.
--Si la politique locale n'exige pas du message d'être chiffré, en utilisant le chiffrement de S/MIME, avant de quitter le domaine alors le message n'est pas enveloppé dans une couche "envelopedData" mais est enveloppé dans une nouvelle couche "signedData", S4, ayant pour résultat le message (S4(S3(S1(contenu_original) envoyé par le domaine.
Le "signedData" S4, dans les deux cas, contient les attributs signés de S2.

5) Application d'une "signature de domaine" à un message signé, chiffré et resigné 2 fois (S3(S2(E1(S1(contenu_original)))))
Aucune des couches "signedData" n'inclue d'attribut "mlExpansionHistory".
- vérification des objets "signedData", S3 et S2;
- détection de l'"envelopedData" E1;- décollement des objets "signedData", S3 et S2;
- déchiffrement de l'"encryptedContent";
- vérification de l'objet "signedData", S1;
- enveloppement du contenu déchiffré dans une nouvelle couche "signedData", S4, qui contient une "signature de domaine".
--Si la politique locale exige du message d'être chiffré en utilisant le chiffrement S/MIME, avant de quitter le domaine, alors ce nouveau message est enveloppé dans une couche "envelopedData", E2, ayant pour résultat le message (E2(S4(S1(contenu_original)))) envoyé par le domaine;
--sinon le message n'est pas enveloppé dans une couche "envelopedData" ayant pour résultat le message (S4(S1(contenu_original))) envoyé par le domaine.

6) Application d'une "signature de domaine" à un message signé, chiffré et resigné 2 fois (S3(S2(E1(S1(contenu_original)))))
La couche "signedData", S3 inclue un attribut "mlExpansionHistory".
- vérification des objets "signedData", S3 et S2;
- détection de l'attribut "mlExpansionHistory" dans S3, ainsi S3 est le "signedData" externe;
- mise en mémoire des attributs signés de S3 pour une inclusion postérieure dans le nouveau "signedData" externe à appliquer au nouveau message;
- décollement de l'objets "signedData", S3;
- détection de l'"envelopedData" E1;- décollement de l'objets "signedData", S2;
- déchiffrement du message;
- vérification de l'objet "signedData", S1;
- enveloppement du contenu déchiffré dans une nouvelle couche "signedData", S4, qui contient une "signature de domaine".
--Si la politique locale exige du message d'être chiffré en utilisant le chiffrement S/MIME, avant de quitter le domaine, alors ce nouveau message est enveloppé dans une couche "envelopedData", E2. Une nouvelle couche de "signedData", S5, est alors enroulée autour de l'"envelopedData", E2, ayant pour résultat le message (S5(E2(S4(S1(contenu_original))))) envoyé par le domaine.

--Si la politique locale n'exige pas du message d'être chiffré, en utilisant le chiffrement de S/MIME, avant de quitter le domaine alors le message n'est pas enveloppé dans une couche "envelopedData" mais est enveloppé dans une nouvelle couche "signedData", S5, ayant pour résultat le message (S5(S4(S1(contenu_original) envoyé par le domaine.
Le "signedData" S5, dans les deux cas, contient les attributs signés de S3.

7) Application d'une "signature de domaine" à un message (signé-chiffré)x2 et resigné (S3(E2(S2(E1(S1(contenu_original))))))
Le "signedData", S3, n'inclut pas d'attribut "mlExpansionHistory".
- vérification de l'objet "signedData", S3;
- détection de l'"envelopedData" E2;- décollement de l'objets "signedData", S3;
- déchiffrement de l'"encryptedContent";
- vérification de l'objet "signedData", S2;
- déchiffrement de l'"envelopedData", E1;
- vérification de l'objet "signedData", S1;
- enveloppement de l'objet "signedData", S2 dans une nouvelle couche "signedData", S4,qui contient une "signature de domaine".
- puisqu'il y a une "envelopedData", E1, inférieure dans le message, le nouveau message est enveloppé dans une couche "envelopedData", E3, ayant pour résultat le message (E3(S4(S2(E1(S1(contenu_original)))))) envoyé par le domaine.

6. Considérations De Sécurité

Ces spécifications se fondent sur l'existence de plusieurs noms caractéristiques, tels que le "domain-confidentiality-authority" ou le "domain-signing-authority". Les organismes doivent prendre en considération ces noms, même si ils ne soutiennent pas de "DOMSEC", de sorte que les certificats délivrés par ces noms soient uniquement délivrés aux entités légitimes. Si ce n'est pas le cas, alors un individu pourrait obtenir un certificat lié à domain-confidentiality-authority@acme.com et en conséquence serait capable de lire des messages d'un client "DOMSEC", destinés à d'autres.

Les implémentations doivent protéger toutes les clefs privées. La compromition de la clef privée du signataire permet la masquerade.

De même, la compromition de la clef "content-encryption" peut avoir comme conséquence la révélation du contenu chiffré.

La compromition de la clef matérielle est considéré comme un problème bien plus sérieux pour des services de sécurité de domaine que pour un client de S/MIME. C'est parce que la compromition de la clef privée peut compromettre, à tour de rôle, la sécurité d'un domaine tout entier. Par conséquent, le grand soin devrait être porté à la considération de sa protection.

Seul, le chiffrement de domaine n'est pas sécurisé et devrait être employé en même temps qu'une signature de domaine pour éviter une attaque par masquerade, où un attaquant qui a obtenu un certificat de l'ACDom peut modifier un message de ce domaine en se faisant passer pour un autre domaine.

Lorsqu'un message chiffré de "DOMSEC" est envoyé à un utilisateur final de telle manière que le message soit déchiffré par les utilisateurs finaux de l'ACDom, le message sera en texte plein et donc la confidentialité pourrait être compromise.

Si l'ACDom du destinataire est compromis alors le destinataire ne peut pas garantir l'intégrité du message. En outre, même si l'ACDom du destinataire vérifie correctement les signatures d'un message qui a été modifié sans que sa modification soit détectable, alors le destinataire ne peut vérifier aucune signature sur ce message.

7. Module de "DOMSEC" ASN.1

```
DOMSECSyntax
  { iso(1) member-body(2) us(840) rsadsi(113549)
       pkcs(1) pkcs-9(9) smime(16) modules(0) domsec(10) }

  DEFINITIONS IMPLICIT TAGS ::=
  BEGIN

  -- TOUTES LES EXPORTATIONS
  -- les types et des valeurs définies dans ce module sont exportées pour
  -- l'usage dans les autres modules ASN.1.
  -- D'autres applications peuvent les employer pour leurs propres comptes.

  SignatureType ::= SEQUENCE OF OBJECT IDENTIFIER

  id-smime OBJECT IDENTIFIER ::= { iso(1) member-body(2)
         us(840) rsadsi(113549) pkcs(1) pkcs-9(9) 16 }

  id-sti  OBJECT IDENTIFIER ::= { id-smime 9 }   -- signature type
  identifier

  -- Signature Type Identifiers / identifiants de type de signature

  id-sti-originatorSig      OBJECT IDENTIFIER ::= { id-sti 1 }
  id-sti-domainSig          OBJECT IDENTIFIER ::= { id-sti 2 }
  id-sti-addAttribSig       OBJECT IDENTIFIER ::= { id-sti 3 }
  id-sti-reviewSig          OBJECT IDENTIFIER ::= { id-sti 4 }

  END -- fin de DOMSECSyntax
```