Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 5698
Catégorie : Sur la voie de la normalisation
ISSN: 2070-1721
Traduction Claude Brière de L'Isle

T. Kunz, Fraunhofer SIT
S. Okunick, pawisda systems GmbH
U. Pordesch, Fraunhofer Gesellschaft
novembre 2009

Structure des données pour l'aptitude à la sécurité (DSSC) des algorithmes de chiffrement

Résumé
Comme les algorithmes de chiffrement peuvent devenir faibles au fil des ans, il est nécessaire d'évaluer leur adaptabilité à la sécurité. Quand on signe ou vérifie des données, ou quand on chiffre ou déchiffre des données, ces évaluations doivent être prises en compte. Le présent document spécifie une structure de données qui permet une analyse automatisée de l'aptitude à la sécurité d'un algorithme de chiffrement donné à un certain moment, qui peut être dans le passé, le présent, ou l'avenir.

Statut de ce mémoire
Ceci est un document de l'Internet sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc5698

Notice de droits de reproduction
Copyright (c) 2009 IETF Trust et les personnes identifiées comme auteurs du document. Tous droits réservés.

Le présent document est soumis au BCP 78 et aux dispositions légales de l'IETF Trust qui se rapportent aux documents de l'IETF (http://trustee.ietf.org/license-info) en vigueur à la date de publication de ce document. Prière de revoir ces documents avec attention, car ils décrivent vos droits et obligations par rapport à ce document. Les composants de code extraits du présent document doivent inclure le texte de licence simplifié de BSD comme décrit au paragraphe 4.e des dispositions légales du Trust et sont fournis sans garantie comme décrit dans la licence de BSD simplifiée.

Le présent document peut contenir des matériaux provenant de documents de l'IETF ou de contributions à l'IETF publiées ou rendues disponibles au public avant le 10 novembre 2008. La ou les personnes qui ont le contrôle des droits de reproduction sur tout ou partie de ces matériaux peuvent n'avoir pas accordé à l'IETF Trust le droit de permettre des modifications de ces matériaux en dehors du processus de normalisation de l'IETF. Sans l'obtention d'une licence adéquate de la part de la ou des personnes qui ont le contrôle des droits de reproduction de ces matériaux, le présent document ne peut pas être modifié en dehors du processus de normalisation de l'IETF, et des travaux dérivés ne peuvent pas être créés en dehors du processus de normalisation de l'IETF, excepté pour le formater en vue de sa publication comme RFC ou pour le traduire dans une autre langue que l'anglais.

Table des matières

1. Introduction

1.1 Motivation

Les signatures numériques peuvent fournir l'intégrité des données et l'authentification. Elles sont fondées sur des algorithmes de chiffrement dont il est exigé qu'ils aient certaines propriétés de sécurité. Par exemple, les algorithmes de hachage doivent être résistants aux collisions, et dans le cas d'algorithmes de clé publique, le calcul de la clé privée qui correspond à une certaine clé publique doit être infaisable. Si des algorithmes n'ont pas les propriétés exigées, les signatures pourraient être falsifiées, sauf si elles sont protégées par un fort algorithme de chiffrement.

Les algorithmes cryptographiques qui sont utilisés dans les signatures devront être choisis pour résister à de telles attaques durant leur période d'utilisation. Pour les clés de signature incluses dans les certificats de clé publique, cette période d'utilisation est la période de validité du certificat. Les algorithmes de chiffrement qui sont utilisés pour le chiffrement devront résister à de telles attaques durant la période prévue pour garder les informations confidentielles.

Seuls très peu d'algorithmes satisfont les exigences de sécurité. Par ailleurs, à cause des performances croissantes des ordinateurs et des progrès de la cryptographie, les algorithmes ou leurs paramètres deviennent non sûrs au fil des ans. L'algorithme de hachage MD5, par exemple, ne convient plus aujourd'hui pour de nombreux objets. Une signature numérique utilisant un algorithme "faible" n'a pas de valeur probatoire, sauf si l'algorithme "faible" a été protégé par un algorithme fort avant le moment où il a été considéré comme faible. De nombreuses sortes de données signées numériquement (y compris des documents signés, des horodatages, des certificats, et des listes de révocation) sont affectés, en particulier dans le cas d'archivage à long terme. Sur de longues périodes, il est supposé que les algorithmes utilisés dans les signatures deviennent non sûrs.

Pour cette raison, il est important d'évaluer périodiquement l'adéquation d'un algorithme et de considérer les résultats de ces évaluations quand on crée et vérifie des signatures, ou quand on maintient la validité de signatures faites dans le passé. Un résultat est une période de validité projetée pour l'algorithme, c'est-à-dire, une prédiction de la période durant laquelle il convient d'utiliser l'algorithme. Cette prédiction peut aider à détecter si un algorithme faible est utilisé dans une signature et si cette signature a été protégée de façon appropriée en temps utile par une autre signature faite en utilisant un algorithme

convenable à l'instant présent. Les évaluations d'algorithme sont faites par des comités d'experts. En Allemagne, l'Agence Fédérale des réseaux publie annuellement les évaluations des algorithmes de chiffrement [BNetzAg.2008]. Des exemples d'autres évaluations Européennes et internationales sont dans [TS102176-1] et [NIST.800-57].

Ces évaluations sont publiées dans des documents destinés à être lus par des humains. Donc, pour permettre un traitement automatique, il est nécessaire de définir une structure de données qui exprime le contenu des évaluations. Cette structure de données normalisée peut être utilisée pour la publication et peut être interprétée par des outils de génération et de vérification de signature. Les évaluations d'algorithme sont regroupées dans une politique d'aptitude de sécurité. Dans ce document est spécifiée une structure pour une politique d'aptitude de sécurité. Donc, le document fournit un cadre pour exprimer les évaluations des algorithmes de chiffrement. Le présent document ne tente pas de cataloguer les propriétés de sécurité des algorithmes de chiffrement. De plus, aucune ligne directrice n'est donnée sur la sorte d'algorithme qui devra être évalué, par exemple, des politiques d'aptitude de sécurité peuvent être utilisées pour évaluer des algorithmes de clé publique et de hachage, des schémas de signature et des schémas de chiffrement.

1.2 Terminologie

Algorithme : un algorithme de chiffrement, c'est-à-dire, un algorithme de clé publique ou de hachage. Pour les algorithmes de clé publique, c'est l'algorithme avec ses paramètres, si il y en a. De plus, le terme "algorithme" est utilisé pour les schémas de chiffrement et pour les fonctions de bourrage.

Opérateur : instance qui utilise et interprète une politique, par exemple, un composant de vérification de signature.

Politique : abréviation pour politique d'aptitude de sécurité.

Éditeur : instance qui publie la politique contenant l'évaluation des algorithmes.

Politique d'aptitude de sécurité : évaluation des algorithmes de chiffrement à l'égard de leur sécurité dans un domaine d'application spécifique, par exemple, signer ou vérifier les données. L'évaluation est publiée dans un format électronique.

Algorithme convenable : algorithme évalué par rapport à une politique et déterminé comme valide, c'est-à-dire, résistant aux attaques, à un instant particulier.

1.2.1 Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

1.3 Cas d'utilisation

Des cas d'utilisation d'une politique d'aptitude de sécurité sont présentés ici.

Archivage à long terme : le cas d'utilisation de plus important est l'archivage à long terme des données signées. Les algorithmes ou leurs paramètres deviennent non sûrs sur de longues périodes. Donc, les signatures de données archivées et des horodatages doivent être périodiquement renouvelées. Une politique fournit des informations sur les algorithmes convenables et menacés. De plus, la politique aide à vérifier les documents archivés ainsi que re-signés.

Services : les services peuvent fournir des informations sur les algorithmes de chiffrement. Sur la base d'une politique, un service est capable de fournir la date à laquelle un algorithme est devenu non sûr ou va probablement le devenir, ainsi que des informations concernant les algorithmes actuellement valides. Des outils de vérification ou des systèmes d'archivage à long terme peuvent demander de tels services et donc n'ont pas besoin de traiter par eux-même de la sécurité de l'algorithme.

Services d'archive à long terme (LTA, *Long-term Archive Services*) : comme défini dans la [RFC4810], peuvent utiliser la politique pour le renouvellement de signature.

Signature et vérification : quand on signe des documents ou des certificats, on doit être assuré que les algorithmes utilisés pour signer ou vérifier sont convenables. En conséquence, quand on vérifie la syntaxe de message cryptographique

(CMS, *Cryptographic Message Syntax*) [RFC5652] ou des signatures XML [RFC3275], [TS101903], non seulement la validité des certificats mais aussi la validité de tous les algorithmes impliqués peut être vérifiée.

Re-chiffrement : une politique d'aptitude de sécurité peut aussi être utilisée pour décider si des documents chiffrés doivent être re-chiffrés parce que l'algorithme de chiffrement n'est plus sûr.

2. Exigences et hypothèses

Le paragraphe 2.1 décrit les exigences générales pour une structure de données contenant l'aptitude à la sécurité des algorithmes. Au paragraphe 2.2, des hypothèses sont spécifiées concernant la conception et l'usage de la structure des données.

Une politique contient une liste d'algorithmes qui ont été évalués par un éditeur. Une évaluation d'algorithme est décrite par un identifiant, des contraintes de sécurité, et une période de validité. Par ces contraintes, les exigences pour les propriétés d'un algorithme doivent être définies ; par exemple, un algorithme de clé publique est évalué sur la base de ses paramètres.

2.1 Exigences

Interprétation automatique : la structure des données de la politique doit permettre une évaluation automatique de l'aptitude à la sécurité d'un algorithme.

Souplesse : la structure de données doit être assez souple pour prendre en charge de nouveaux algorithmes. Les futures publications de politique peuvent inclure des évaluations d'algorithmes qui sont inconnus actuellement. Il doit être possible d'ajouter de nouveaux algorithmes avec les contraintes de sécurité correspondantes dans la structure des données. De plus, la structure de données doit être indépendante de l'utilisation prévue, par exemple, chiffrement, signature, vérification, et renouvellement de signature. Donc, la structure de données est utilisable dans chaque cas d'utilisation.

Authentification de source : les politiques peuvent être publiées par différentes institutions, par exemple, au niveau national ou au niveau de l'Union Européenne (EU) tandis qu'une politique n'a pas besoin d'être en accord avec d'autres. De plus, des organisations peuvent entreprendre leurs propres évaluations pour des besoins internes. Pour cette raison, une politique doit être attribuable à son éditeur.

Intégrité et authenticité : il doit être possible d'assurer l'intégrité et l'authenticité d'une politique d'aptitude à la sécurité publiée. De plus, la date de production doit être identifiable.

2.2 Hypothèses

Il est supposé qu'une politique contient les évaluations de tous les algorithmes actuellement connus, y compris expirés.

Un algorithme est convenable à un instant intéressant si il est contenu dans la politique courante et si l'instant intéressant est dans la période de validité. De plus, si l'algorithme a des paramètres, ces paramètres doivent satisfaire les exigences définies dans les contraintes de sécurité.

Si un algorithme apparaît dans une politique pour la première fois, on peut supposer que l'algorithme a déjà été convenable dans le passé. Généralement, les algorithmes sont utilisés en pratique avant l'évaluation.

Pour éviter des incohérences, plusieurs instances du même algorithme sont interdites. L'éditeur doit veiller à empêcher des conflits au sein d'une politique.

Les assertions faites dans la politique sont convenables au moins jusqu'à la publication de la prochaine politique.

Les éditeurs peuvent étendre la durée de vie d'un algorithme avant d'atteindre la fin de la période de validité de l'algorithme en publiant une politique révisée. Les éditeurs ne devraient pas faire revivre des algorithmes qui ont expiré au moment où une politique révisée est publiée.

3. Structures de données

Cette Section décrit la syntaxe d'une politique d'aptitude à la sécurité définie par un schéma XML [XML-SCHEMA]. Les modules ASN.1 sont définis dans les appendices C et D. Le schéma utilise l'espace de noms XML suivant [XML-NAMES] : urn:ietf:params:xml:ns:dssc

Dans ce document, le préfixe "dssc" est utilisé pour cet espace de noms. Le schéma commence par la définition de schéma suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:dssc="urn:ietf:params:xml:ns:dssc"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        targetNamespace="urn:ietf:params:xml:ns:dssc"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="xmldsig-core-schema.xsd"/>
```

3.1 SecuritySuitabilityPolicy

L'élément SecuritySuitabilityPolicy est l'élément racine d'une politique. Il a un attribut Identifiant facultatif, qui DOIT être utilisé comme référence quand on signe la politique (paragraphe 3.13). L'attribut facultatif lang définit le langage conformément à la [RFC5646]. Le langage est appliqué à tout texte lisible par l'homme au sein de la politique. Si l'attribut lang est omis, le langage par défaut est l'anglais ("en"). L'élément est défini par le schéma suivant :

```
<xs:element name="SecuritySuitabilityPolicy" type="dssc:SecuritySuitabilityPolicyType"/>
  <xs:complexType name="SecuritySuitabilityPolicyType">
    <xs:sequence>
      <xs:element ref="dssc:PolicyName"/>
      <xs:element ref="dssc:Publisher"/>
      <xs:element name="PolicyIssueDate" type="xs:dateTime"/>
      <xs:element name="NextUpdate" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="Usage" type="xs:string" minOccurs="0"/>
      <xs:element ref="dssc:Algorithm" maxOccurs="unbounded"/>
      <xs:element ref="ds:Signature" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" default="1"/>
    <xs:attribute name="lang" default="en"/>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
```

3.2 PolicyName

L'élément PolicyName contient un nom arbitraire pour la politique. Les éléments facultatifs Object Identifier (OID) et Uniform Resource Identifier (URI) PEUVENT être utilisés pour l'identification de la politique. Les OID DOIVENT être exprimés dans la notation à points.

```
<xs:element name="PolicyName" type="dssc:PolicyNameType"/>
  <xs:complexType name="PolicyNameType">
    <xs:sequence>
      <xs:element ref="dssc:Name"/>
      <xs:element ref="dssc:ObjectIdentifier" minOccurs="0"/>
      <xs:element ref="dssc:URI" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

```
<xs:element name="Name" type="xs:string"/>
<xs:element name="ObjectIdentifier">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:pattern value="(\d+\.)+\d+"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
<xs:element name="URI" type="xs:anyURI"/>
```

3.3 Publisher

L'élément Publisher contient des informations sur l'éditeur de la politique. Il est composé du nom (par exemple, nom de l'institution) d'une adresse facultative, et d'un URI facultatif. L'élément Address contient un texte arbitraire de format libre non destiné à un traitement automatique.

```
<xs:element name="Publisher" type="dssc:PublisherType"/>
 <xs:complexType name="PublisherType">
  <xs:sequence>
   <xs:element ref="dssc:Name"/>
   <xs:element name="Address" type="xs:string" minOccurs="0"/>
   <xs:element ref="dssc:URI" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
```

3.4 PolicyIssueDate

L'élément PolicyIssueDate indique l'instant où la politique a été produite.

3.5 NextUpdate

L'élément facultatif NextUpdate PEUT être utilisé pour indiquer quand la prochaine politique sera produite.

3.6 Usage

L'élément facultatif Usage détermine l'utilisation prévue de la politique (par exemple, validation de certificat, signature et vérification de documents). L'élément contient un texte de format libre destiné seulement à la lecture par l'homme.

3.7 Algorithm

Une politique d'aptitude à la sécurité DOIT contenir au moins un élément Algorithm. Un algorithme est identifié par un élément AlgorithmIdentifier. De plus, l'élément Algorithm contient toutes les évaluations de l'algorithme de chiffrement spécifique. Plus d'une évaluation peut être nécessaire si l'évaluation dépend des contraintes de paramètre. L'élément facultatif Information PEUT être utilisé pour fournir des informations supplémentaires comme des références aux spécifications d'algorithme. Afin de donner l'option d'étendre l'élément Algorithm, il contient de plus un caractère générique. L'élément Algorithm est défini par le schéma suivant :

```
<xs:element name="Algorithm" type="dssc:AlgorithmType"/>
 <xs:complexType name="AlgorithmType">
  <xs:sequence>
   <xs:element ref="dssc:AlgorithmIdentifier"/>
   <xs:element ref="dssc:Evaluation" maxOccurs="unbounded"/>
   <xs:element ref="dssc:Information" minOccurs="0"/>
   <xs:any namespace="##other" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
```

3.8 AlgorithmIdentifier

L'élément AlgorithmIdentifier est utilisé pour identifier un algorithme de chiffrement. Il consiste en le nom de l'algorithme, au moins un OID, et des URI facultatifs. Le nom de l'algorithme n'est pas destiné à être analysé par des processus automatiques. Il est seulement destiné à être lu par l'homme. L'OID DOIT être exprimé dans la notation à points (par exemple, "1.3.14.3.2.26"). L'élément est défini comme suit :

```
<xs:element name="AlgorithmIdentifier" type="dssc:AlgorithmIdentifierType"/>
<xs:complexType name="AlgorithmIdentifierType">
  <xs:sequence>
    <xs:element ref="dssc:Name"/>
    <xs:element ref="dssc:ObjectIdentifier" maxOccurs="unbounded"/>
    <xs:element ref="dssc:URI" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

3.9 Evaluation

L'élément Evaluation contient l'évaluation d'un algorithme de chiffrement par rapport à ses contraintes de paramètres. Par exemple, l'aptitude de l'algorithme RSA dépend de la longueur du module (RSA avec une longueur de module de 1024 peut avoir une autre période d'aptitude que RSA avec une longueur de module de 2048). Les algorithmes de hachage courants comme SHA-1 ou RIPEMD-160 n'ont pas de paramètres. Donc, l'élément Parameter est facultatif. L'aptitude de l'algorithme est exprimée par une période de validité, qui est définie par l'élément Validity. Un caractère générique facultatif PEUT être utilisé pour étendre l'élément Evaluation.

```
<xs:element name="Evaluation" type="dssc:EvaluationType"/>
<xs:complexType name="EvaluationType">
  <xs:sequence>
    <xs:element ref="dssc:Parameter" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="dssc:Validity"/>
    <xs:any namespace="##other" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

3.10 Parameter

L'élément Parameter est utilisé pour exprimer des contraintes sur des paramètres spécifiques de l'algorithme.

L'élément Parameter a un attribut de nom, qui contient le nom du paramètre (par exemple, "moduluslength" pour RSA [RFC3447]). La Section 5 définit les noms de paramètres pour les algorithmes de clé publique connus actuellement ; ces noms de paramètres DEVRAIENT être utilisés. Pour le paramètre actuel, une gamme de valeurs ou une valeur exacte peut être définie. Ces contraintes sont exprimées par les éléments suivants :

Min : l'élément Min définit la valeur minimum du paramètre. Cela signifie que des valeurs égales ou supérieures à la valeur donnée satisfont l'exigence.

Max : l'élément Max définit la valeur maximum que peut prendre le paramètre.

Au moins un des deux éléments DOIT être établi pour définir une gamme de valeurs. Une gamme PEUT aussi être spécifiée par une combinaison des deux éléments, tandis que la valeur de l'élément Min DOIT être inférieure ou égale à la valeur de l'élément Max . Le paramètre peut avoir toute valeur dans la gamme définie, incluant les valeurs minimum et maximum. Une valeur exacte est exprimée en utilisant la même valeur dans les deux éléments Min et Max.

Ces contraintes sont suffisantes pour tous les algorithmes courants. Si de futurs algorithmes ont besoin de contraintes qui ne peuvent pas être exprimées par les éléments ci-dessus, une structure XML arbitraire PEUT être insérée qui satisfasse les nouvelles contraintes. Pour cette raison, l'élément Parameter contient un caractère générique. Un paramètre DOIT contenir au moins une contrainte. Le schéma pour l'élément Parameter est comme suit :

```
<xs:element name="Parameter" type="dssc:ParameterType"/>
<xs:complexType name="ParameterType">
```

```
  <xs:sequence>
    <xs:element name="Min" type="xs:int" minOccurs="0"/>
    <xs:element name="Max" type="xs:int" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

3.11 Validity

L'élément Validity est utilisé pour définir la période d'aptitude (prévue) de l'algorithme. Il est composé d'une date de début facultative et d'une date de fin facultative. Ne pas définir de date de fin signifie que l'algorithme a une validité ouverte. Bien sûr, cela peut être restreint par une future politique qui établit une date de fin pour l'algorithme. Si la fin de la période de validité est dans le passé, l'algorithme a été convenable jusqu'à cette date de fin. L'élément est défini par le schéma suivant :

```
<xs:element name="Validity" type="dssc:ValidityType"/>
<xs:complexType name="ValidityType">
  <xs:sequence>
    <xs:element name="Start" type="xs:date" minOccurs="0"/>
    <xs:element name="End" type="xs:date" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

3.12 Information

L'élément Information PEUT être utilisé pour donner des informations textuelle supplémentaires sur l'algorithme ou l'évaluation, par exemple, des références sur les spécifications de l'algorithme. L'élément est défini comme suit :

```
<xs:element name="Information" type="dssc:InformationType"/>
<xs:complexType name="InformationType">
  <xs:sequence>
    <xs:element name="Text" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

3.13 Signature

L'élément facultatif Signature PEUT être utilisé pour garantir l'intégrité et l'authenticité de la politique. C'est une signature XML spécifiée dans la [RFC3275]. La signature DOIT se rapporter à l'élément SecuritySuitabilityPolicy. Si l'élément Signature est établi, l'élément SecuritySuitabilityPolicy DOIT avoir l'attribut facultatif id. Cet attribut DOIT être utilisé pour faire référence à l'élément SecuritySuitabilityPolicy au sein de l'élément Signature. Comme c'est une signature enveloppée, la signature DOIT utiliser l'algorithme de signature identifié par l'URI suivant :

http://www.w3.org/2000/09/xmldsig#enveloped-signature

4. Politiques de DSSC

Les politiques de DSSC DOIVENT être exprimées soit en XML soit en ASN.1. Cependant, afin d'atteindre l'interopérabilité, les politiques de DSSC DEVRAIENT être publiées à la fois en XML et en ASN.1.

Dans le cas de XML, une politique de DSSC est un document XML qui DOIT être bien formé et DEVRAIT être valide. Les politiques de DSSC codées en XML DOIVENT être fondées sur [XML 1.0] et DOIVENT être codées en utilisant UTF-8 [RFC3629]. La présente spécification utilise les espaces de noms XML [XML-NAMES] pour identifier les politiques de DSSC. L'URI d'espace de noms pour les éléments définis par la présente spécification est un URN [RFC2141] utilisant le préfixe d'espace de noms "dssc". Cet URN est : urn:ietf:params:xml:ns:dssc.

Les politiques de DSSC codées en XML sont identifiées avec le type MIME "application/dssc+xml" et sont des instances

du schéma XML [XMLSchema] défini dans l'Appendice B.

Un fichier contenant une politique de DSSC en représentation ASN.1 (pour la spécification de l'ASN.1, se référer à [X.208], [X.209], [X.680] et [X.690]) DOIT contenir seulement le codage DER d'une politique de DSSC, c'est-à-dire, il NE DOIT PAS y avoir d'en-tête étranger ni d'informations en queue dans le fichier. Les politiques de DSSC fondées sur l'ASN.1 sont identifiées avec le type MIME "application/dssc+der". Les modules ASN.1 appropriés sont définis dans les appendices C (syntaxe ASN.1 de 1988) et D (syntaxe ASN.1 de 1997).

5. Définition des paramètres

Cette Section définit les noms de paramètres pour les algorithmes de clé publique connus actuellement. Les paramètres suivants se réfèrent aussi aux schémas de chiffrement fondés sur ces algorithmes de clé publique (par exemple, le schéma de signature PKCS n° 1 v1.5 SHA-256 avec RSA [RFC3447]).

Le paramètre de RSA [RFC3447] DEVRAIT être nommé "moduluslength".

Les paramètres pour l'algorithme de signature numérique (DSA, *Digital Signature Algorithm*) [FIPS186-2] DEVRAIENT être "plength" et "qlength".

Ces noms de paramètres ont été enregistrés par l'IANA (voir la Section 8). Il peut être nécessaire d'enregistrer plus d'algorithmes non donnés dans cette section (en particulier, des algorithmes futurs). Le processus pour enregistrer des noms de paramètres d'autres algorithmes est décrit dans la Section 8. Les éditeurs de politiques DEVRAIENT utiliser ces noms de paramètres afin de garantir une interprétation correcte.

6. Traitement

L'évaluation de l'aptitude à la sécurité d'un algorithme est décrite en trois parties : vérification de la politique, détermination de la validité de l'algorithme, et évaluation des paramètres de l'algorithme, si il en est.

Dans les paragraphes suivants, un processus est décrit
- o pour déterminer si un algorithme était convenable à un moment particulier, et
- o pour déterminer jusqu'à quand un algorithme a été ou va être convenable.

6.1 Entrées

Pour déterminer l'aptitude à la sécurité d'un algorithme, les informations suivantes sont exigées :
- o Politique
- o Heure actuelle
- o Identifiant d'algorithme et contraintes de paramètre (si il en est d'associées)
- o Période d'intérêt (facultatif). Ne pas fournir de période d'intérêt signifie la détermination de la date de fin de validité de l'algorithme.

6.2 Vérification de politique

La signature sur la politique DEVRAIT être vérifiée et un chemin de certification depuis le certificat du signataire de la politique jusqu'à une ancre de confiance courante DEVRAIT être construit et validé [RFC5280]. Les algorithmes utilisés pour vérifier la signature numérique et valider le chemin de certification DOIVENT être convenables par rapport au contenu de la politique vérifiée. Si la vérification de signature échoue, la validation du chemin de certification échoue ou un algorithme inadéquat est exigé pour effectuer ces vérifications, alors la politique DOIT être rejetée.

L'heure de nextUpdate dans la politique DOIT être supérieure à l'heure courante ou absente. Si l'heure de nextUpdate est avant l'heure courante, la politique DOIT être rejetée.

6.3 Évaluation d'algorithme

Pour déterminer la période de validité d'un algorithme, on localise dans la politique l'élément Algorithm qui correspond à l'identifiant d'algorithme fourni en entrée. L'élément Algorithm est localisé en comparant l'OID dans l'élément à l'OID inclus dans l'identifiant d'algorithme fourni en entrée.

Si aucun élément Algorithm correspondant n'est trouvé, l'algorithme est inconnu.

Si l'instant intéressant était fourni en entrée, la validité de chaque élément Evaluation DOIT être vérifiée afin de déterminer si l'algorithme était convenable à l'instant intéressant. Pour chaque élément Evaluation :

- Confirmer que l'instant de début est avant l'instant intéressant ou absent. Éliminer l'entrée si l'instant de début est présent et supérieur à l'instant intéressant.
- Confirmer que l'instant de fin est supérieur à l'instant intéressant ou absent. Éliminer l'entrée si l'instant de fin est présent et inférieur à l'instant intéressant.

Si tous les éléments Evaluation ont été rejetés, l'algorithme n'est pas convenable conformément à la politique.

Toutes les entrées non rejetées vont être utilisées pour l'évaluation des paramètres, si il en est.

6.4 Évaluation des paramètres

Tout paramètre nécessaire aux entrées non rejeté DOIT être évalué dans le contexte du type et de l'usage de l'algorithme. Les détails de l'évaluation du paramètre sont définis par algorithme.

Pour évaluer les paramètres, les éléments Parameter de chaque élément Evaluation qui n'a pas été rejeté dans le processus décrit au paragraphe 6.3 DOIVENT être vérifiés. Pour chaque élément Parameter :

- Confirmer que le paramètre a été fourni en entrée. Éliminer l'élément Evaluation si le paramètre ne correspond pas à un des paramètres fournis en entrée.
- Si l'élément Parameter a un élément Min, confirmer que la valeur de paramètre est inférieure ou égale au paramètre correspondant fourni en entrée. Éliminer l'élément Evaluation si la valeur de paramètre ne satisfait pas la contrainte.
- Si l'élément Parameter a un élément Max, confirmer que la valeur de paramètre est supérieure ou égale au paramètre correspondant fourni en entrée. Éliminer l'élément Evaluation si la valeur de paramètre ne satisfait pas la contrainte.
- Si le paramètre a une autre contrainte, confirmer que la valeur du paramètre correspondant fourni comme entrée satisfait cette contrainte. Sinon, ou si la contrainte n'est pas reconnue, éliminer l'élément Evaluation.

Si tous les éléments Evaluation ont été rejetés, l'algorithme n'est pas convenable conformément à la politique.

Toutes les entrées non rejetées vont être fournies comme résultat.

6.5 Résultat

Si l'algorithme n'est pas dans la politique, retourner une erreur "algorithme inconnu".

Si aucun instant intéressant n'a été fourni en entrée, retourner l'instant de fin maximum des éléments Evaluation qui n'ont pas été éliminés. Si au moins un instant de fin de ces éléments Evaluation est absent, retourner "l'algorithme a un instant de fin indéfini".

Autrement, si l'algorithme n'est pas convenable par rapport à l'instant intéressant, retourner une erreur "algorithme non approprié".

Si l'algorithme convient par rapport à l'instant intéressant, retourner les éléments Evaluation qui n'ont pas été éliminés.

7. Considérations sur la sécurité

La politique pour l'aptitude à la sécurité d'un algorithme a un grand impact sur la qualité des résultats des opérations de génération et de vérification de signature. Si un algorithme est incorrectement évalué par rapport à une politique, des signatures avec une faible force probatoire pourraient être créées, ou les résultats de vérification pourraient être incorrects. Les considérations sur la sécurité suivantes ont été identifiées :

1. Les éditeurs DOIVENT s'assurer qu'une manipulation non autorisée de toute aptitude à la sécurité n'est pas possible avant qu'une politique soit signée et publiée. Il n'y a pas de mécanisme fourni pour révoquer une politique après sa publication. Comme les évaluations d'algorithme changent peu fréquemment, la durée de vie d'une politique devrait être considérée avec attention avant sa publication.

2. Les opérateurs DEVRAIENT seulement accepter des politiques provenant d'un éditeur de confiance. De plus, la validité du certificat utilisé pour signer la politique DEVRAIT être vérifiable par une liste de révocation de certificat (CRL, *Certificate Revocation List*) [RFC5280] ou le protocole d'état de certificat en ligne (OCSP, *Online Certificate Status Protocol*) [RFC2560]. Le certificat utilisé pour signer la politique DEVRAIT être révoqué si les algorithmes utilisés dans ce certificat ne sont plus convenables. Il NE DOIT PAS être possible d'altérer ou remplacer une politique une fois acceptée par un opérateur.

3. Les opérateurs DEVRAIENT vérifier périodiquement si une nouvelle politique a été publiée pour éviter d'utiliser des informations de politique obsolètes. Pour les éditeurs, il est suggéré de ne pas omettre l'élément NextUpdate afin de donner aux opérateurs une indication sur le moment où la prochaine politique sera publiée.

4. Quand on signe une politique, les algorithmes qui sont convenables conformément à cette politique DEVRAIENT être utilisés.

5. Les règles de traitement décrites à la Section 6 sont sur un algorithme de chiffrement indépendamment du cas d'utilisation. Selon le cas d'utilisation, un algorithme qui n'est plus convenable à l'instant intéressant, ne signifie pas nécessairement que la structure de données où il est utilisé n'est plus sûre. Par exemple, une signature a été faite avec une clé de signataire RSA de 1024 bits. Cette signature est horodatée avec un jeton d'horodatage qui utilise une clé RSA de 2048 bits, avant qu'une taille de clé RSA de 1024 bits ne soit cassée. Le fait que la clé de signature de 1024 bits ne soit plus convenable à l'instant intéressant ne signifie pas que toute la structure de données n'est plus sûre, si une taille de clé RSA de 2048 bits est toujours convenable à l'instant intéressant.

6. En plus des considérations de taille de clé, d'autres considérations doivent être appliquées, comme si un jeton d'horodatage a été fourni par une autorité de confiance. Cela signifie que la simple utilisation d'une politique d'aptitude n'est pas le seul élément à considérer quand on évalue la sécurité d'une structure de données complexe qui utilise plusieurs algorithmes de chiffrement.

7. Les politiques décrites dans ce document sont convenables pour évaluer les algorithmes de chiffrement de base, comme les algorithmes de clé publique ou de hachage, ainsi que les schémas de chiffrement (par exemple, les schémas de signature PKCS n°1 v 1.5 [RFC3447]). Mais on DOIT se souvenir qu'un algorithme de chiffrement de base qui est convenable conformément à la politique ne signifie pas nécessairement que tous les schémas de chiffrement fondés sur cet algorithme sont aussi sûrs. Par exemple, un schéma de signature fondé sur RSA ne doit pas nécessairement être sûr si RSA est convenable. Dans le cas d'une vérification de signature complète, incluant la validation du chemin de certification, divers algorithmes doivent être confrontés à la politique (c'est-à-dire, les schémas de signature des objets de données signés et les informations de révocation, les algorithmes de clé publique des certificats impliqués, etc.). Donc, une politique DEVRAIT contenir des évaluations des algorithmes de clé publique et de hachage ainsi que des schémas de signature.

8. Les documents de re-chiffrement qui étaient originellement chiffrés en utilisant un algorithme qui n'est plus convenable ne vont pas protéger la sémantique du document si le document a été intercepté. Cependant, pour les documents mémorisés sous forme chiffrée, le re-chiffrement doit être envisagé, sauf si le document a perdu sa valeur originale.

8. Considérations relatives à l'IANA

Le présent document définit l'espace de noms XML "urn:ietf:params:xml:ns:dssc" conformément aux lignes directrices de la [RFC3688]. Cet espace de noms a été enregistré dans le registre XML de l'IANA.

Le présent document définit un schéma XML (voir l'Appendice B) conformément aux lignes directrices de la [RFC3688]. Ce schéma XML a été enregistré dans le registre XML de l'IANA et peut être identifié avec l'URN "urn:ietf:params:xml:schema:dssc".

Le présent document définit le type MIME "application/dssc+xml". Ce type MIME a été enregistré par l'IANA sous

"MIME Media Types" conformément aux procédures de la [RFC4288].

Nom de type : application
Nom de sous type : dssc+xml
Paramètres exigés : aucun
Paramètres facultatifs : "charset" comme spécifié pour "application/xml" dans la [RFC3023].
Considérations de codage : les mêmes que spécifiées pour "application/xml" dans la [RFC3023].
Considérations de sécurité : les mêmes que spécifiées pour "application/xml" à la Section 10 de la [RFC3023]. Pour plus de considérations sur la sécurité, voir la Section 7 du présent document.
Considérations d'interopérabilité : les mêmes que spécifiées pour "application/xml" dans la [RFC3023].
Spécification publiée : le présent document.
Applications qui utilisent ce support : applications d'archivage à long terme de données signées, applications pour signer/vérifier des données signées, et applications pour chiffrer/déchiffrer des données.
Informations supplémentaires :
Numéro magique : aucun
Extension de fichier : .xdssc
Code de type de fichier Macintosh : "TEXT"
Identifiant d'objet : aucun
Personne à contacter pour plus d'informations : Thomas Kunz (thomas.kunz@sit.fraunhofer.de)
Usage précu : COMMUN
Restrictions d'usage : aucune
Auteur/contrôleur des changements : IETF

Le présent document définit le type MIME "application/dssc+der". Ce type MIME a été enregistré par l'IANA sous "MIME Media Types" conformément aux procédures de la [RFC4288].

Nom de type : application
Nom de sous type : dssc+der
Paramètres exigés : aucun
Paramètres facultatifs : aucun
Considérations de codage : binary
Considérations de sécurité : voir la Section 7 du présent document.
Considérations d'interopérabilité : aucune
Spécification publiée : le présent document.
Applications qui utilisent ce support : applications d'archivage à long terme de données signées, applications pour signer/vérifier des données signées, et applications pour chiffrer/déchiffrer des données.
Informations supplémentaires :
Numéro magique : aucun
Extension de fichier : .dssc
Code de type de fichier Macintosh : : aucun
Identifiant d'objet : aucun
Personne à contacter pour plus d'informations : Thomas Kunz (thomas.kunz@sit.fraunhofer.de)
Usage précu : COMMUN
Restrictions d'usage : aucune
Auteur/contrôleur des changements : IETF

La présente spécification crée un nouveau registre IANA intitulé "Structure de données pour l'aptitude à la sécurité des algorithmes de chiffrement (DSSC)". Ce registre contient deux sous-registres intitulés"Définitions de paramètres" et "Algorithmes de chiffrement". La politique pour les futures allocations au sous registre "Définitions de paramètres" est "RFC exigée".

Les valeurs initiales pour le sous registre "Définitions de paramètres" sont :

Valeur	**Description**	**Référence**
moduluslength	Paramètre pour RSA (valeur d'entier)	RFC 5698
plength	Paramètre pour DSA (valeur d'entier, utilisé avec le paramètre "qlength")	RFC 5698
qlength	Paramètre pour DSA (valeur d'entier, utilisé avec le paramètre "plength")	RFC 5698

Le sous registre "Algorithmes de chiffrement" contient les noms textuels ainsi que les identifiants d'objet (OID) et les identifiants de ressource universels (URI, *Uniform Resource Identifier*) des algorithmes de chiffrement. Il sert d'assistance à la création d'une nouvelle politique. La politique pour les futures allocations est "premier arrivé, premier servi". Quand on

enregistre un nouvel algorithme, les informations suivantes DOIVENT être fournies :
- o Le nom textuel de l'algorithme.
- o L'OID de l'algorithme.
- o Une référence à une spécification publiquement disponible qui définit l'algorithme et ses identifiants.

Facultativement, un URI PEUT être fourni si possible.

Les valeurs initiales pour le sous registre "Algorithmes de chiffrement" sont :

Nom	**OID / URI**	**Référence**
rsaEncryption	1.2.840.113549.1.1.1	[RFC3447]
dsa	1.2.840.10040.4.1	[RFC3279]
md2	1.2.840.113549.2.2	[RFC3279]
md5	1.2.840.113549.2.5	[RFC3279]
	http://www.w3.org/2001/04/xmldsig-more#md5	[RFC4051]
sha-1	1.3.14.3.2.26	[RFC3279]
	http://www.w3.org/2000/09/xmldsig#sha1	[RFC3275]
sha-224	2.16.840.1.101.3.4.2.4	[RFC4055]
	http://www.w3.org/2001/04/xmldsig-more#sha224	[RFC4051]
sha-256	2.16.840.1.101.3.4.2.1	[RFC4055]
sha-384	2.16.840.1.101.3.4.2.2	[RFC4055]
	http://www.w3.org/2001/04/xmldsig-more#sha384	[RFC4051]
sha-512	2.16.840.1.101.3.4.2.3	[RFC4055]
md2WithRSAEncryption	1.2.840.113549.1.1.2	[RFC3443]
md5WithRSAEncryption	1.2.840.113549.1.1.4	[RFC3443]
	http://www.w3.org/2001/04/xmldsig-more#rsa-md5	[RFC4051]
sha1WithRSAEncryption	1.2.840.113549.1.1.5	[RFC3443]
	http://www.w3.org/2000/09/xmldsig#rsa-sha1	[RFC3275]
sha256WithRSAEncryption	1.2.840.113549.1.1.11	[RFC3443]
	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	[RFC4051]
sha384WithRSAEncryption	1.2.840.113549.1.1.12	[RFC3443]
	http://www.w3.org/2001/04/xmldsig-more#rsa-sha384	[RFC4051]
sha512WithRSAEncryption	1.2.840.113549.1.1.13	[RFC3443]
	http://www.w3.org/2001/04/xmldsig-more#rsa-sha512	[RFC4051]
sha1WithDSA	1.2.840.10040.4.3	[RFC3279]
	http://www.w3.org/2000/09/xmldsig#dsa-sha1	[RFC3275]

9. Références

9.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. DOI 10.17487/RFC2119, *(MàJ par* RFC8174)

[RFC2141] R. Moats, "Syntaxe des URN", mai 1997. *(Obsolète, voir* RFC8141)

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin et C. Adams, "Protocole d'état de certificat en ligne d'infrastructure de clé publique X.509 pour l'Internet - OCSP", juin 1999. *(P.S.) (Remplacée par* RFC6960)

[RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de supports XML", janvier 2001. (*Obsolète, voir* RFC7303)

[RFC3275] D. Eastlake 3rd , J. Reagle, D. Solo, "Syntaxe et traitement de signature en langage de balisage extensible (XML)", mars 2002. *(D.S.)*

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003, DOI 10.17487/RFC3629.

[RFC3688] M. Mealling, "Registre XML de l'IETF", BCP 81, janvier 2004.

[RFC4288] N. Freed et J. Klensin, "Spécifications du type de support et procédures d'enregistrement", BCP 13, décembre 2005.

[RFC4998] T. Gondrom et autres, "Syntaxe d'enregistrement de preuve (ERS)", août 2007. *(P.S.)*

[RFC5280] D. Cooper et autres, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", DOI 10.17487/RFC5280, mai 2008. (*Remplace les* RFC3280, RFC4325, RFC4630) *(P.S. ; MàJ par* RFC8398, 8399)

[RFC5646] A. Phillips, M. Davis, "Étiquettes d'identification des langages", BCP0047, septembre 2009. *(Remplace* RFC4646)

[RFC5652] R. Housley, "Syntaxe de message cryptographique (CMS)", STD70, septembre 2009. *(Remplace* RFC3852)

[XML1.0] Yergeau, F., Maler, E., Paoli, J., Sperberg-McQueen, C., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fifth ed)", W3C Recommendation xml, novembre 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

[XMLNAMES] Bray, T., Hollander, D., and A. Layman, "Namespaces in XML", W3C Recommendation xml-names, août 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816>.

[XMLSchema] World Wide Web Consortium, "XML Schema Part 1: Structures", W3C XML Schema, octobre 2004, <http://www.w3.org/TR/xmlschema-1/>.

[X.680] Recommandation UIT-T X.680 | ISO/CEI 8824-1:2002. "Technologie de l'information - Notation de syntaxe abstraite numéro un (ASN.1) : spécification de la notation de base". <https://www.itu.int/rec/T-REC-X.680>

[X.690] Recommandation UIT-T X.690 | ISO/CEI 8825-1:2002, "Technologies de l'information - Règles de codage de l'ASN.1 : Spécification des règles de codage de base (BER), règles de codage canoniques (CER) et règles de codage distinctives (DER)", juillet 2002. <https://www.itu.int/rec/T-REC-X.690>

9.2 Références pour information

[BNetzAg] Federal Network Agency for Electricity, Gas, Telecommunications, Post and Railway, "Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Uebersicht ueber geeignete Algorithmen)", décembre 2007, <http://www.bundesnetzagentur.de/media/archive/12198.pdf>.

[FIPS186-2] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS PUB 186-2 with Change Notice, janvier 2000.

[NIST.800-576] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1: General (Revised)", NIST 800-57 Part 1, mai 2006.

[RFC3279] L. Bassham, W. Polk et R. Housley, "Algorithmes et identifiants pour le profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002.

[RFC3443] P. Agarwal, B. Akyol, "Traitement de la durée de vie (TTL) dans les réseaux à commutation d'étiquettes multi-protocoles (MPLS)", janvier 2003. *(P.S.)*

[RFC4051] D. Eastlake 3rd, "Identifiants de ressource universels (URI) de sécurité supplémentaires en XML ", avril 2005. *(P.S.) (Remplacée par* RFC6931)

[RFC4055] J. Schaad et autres, "Algorithmes et identifiants supplémentaires pour la cryptographie RSA à utiliser dans le profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", juin 2005.

[RFC4810] C. Wallace et autres, "Exigences pour un service d'archive à long terme", mars 2007. *(Information)*

[TS101903] European Telecommunication Standards Institute (ETSI), "XML Advanced Electronic Signatures (XAdES)",

ETSI TS 101 903 V1.3.2, mars 2006.

[TS102176-1] European Telecommunication Standards Institute (ETSI), "Electronic Signatures and Infrastructures (ESI); "Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms"", ETSI TS 102 176-1 V2.0.0, novembre 2007.

[X.208] Recommandation UIT-T X.208, "Spécification de la notation numéro un de syntaxe abstraite (ASN.1)", Genève, novembre 1988.

[X.209] Recommandation UIT-T X.209, "Spécification des règles de codage de base pour la notation numéro un de syntaxe abstraite (ASN.1)", Genève, 1988.

Appendice A. DSSC et ERS

A.1 Vérification des enregistrements Evidence avec DSSC (pour information)

Cette section décrit la vérification d'un enregistrement de preuve *(Evidence Record)* conformément à la syntaxe d'enregistrement de preuve (ERS, *Evidence Record Syntax*) [RFC4998], en utilisant la structure de données présentée.

Un enregistrement de preuve contient une séquence de ArchiveTimeStampChains *(chaînes d'horodatages d'archive)* qui consiste en ArchiveTimeStamp. Pour chaque ArchiveTimeStamp, l'algorithme de hachage utilisé pour l'arborescence de hachages (digestAlgorithm) ainsi que l'algorithme de clé publique et l'algorithme de hachage dans la signature d'horodatage doivent être examinés. La date pertinente est l'information de temps dans l'horodatage (date de production). En commençant par le premier ArchiveTimeStamp, on doit s'assurer que :

1. L'horodatage utilise les algorithmes de clé publique et de hachage qui étaient convenables à la date de production.

2. L'arborescence de hachage a été construite avec un algorithme de hachage qui était aussi convenable à la date de production.

3. Les algorithmes pour l'horodatage et l'arborescence de hachage dans le ArchiveTimeStamp précédant doivent avoir été convenables à la date de production du ArchiveTimeStamp considéré.

4. Les algorithmes dans le dernier ArchiveTimeStamp doivent être convenables maintenant.

Si la vérification d'un de ces éléments échoue, cela va conduire à l'échec de la vérification.

A.2 Mémorisation des politiques de DSSC dans les enregistrements de preuve (normative)

Cette section décrit comment mémoriser une politique dans un enregistrement de preuve. L'ERS fournit le champ cryptoInfos pour la mémorisation de données de vérification supplémentaires. Pour l'intégration d'une politique d'aptitude à la sécurité dans un enregistrement de preuve, les types de contenu suivants sont définis pour la représentation en ASN.1 et en XML :

DSSC_ASN1 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) ltans(11) id-ct(1) id-ct-dssc-asn1(2) }

DSSC_XML {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) ltans(11) id-ct(1) id-ct-dssc-xml(3) }

Appendice B. Schéma XML (normatif)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:dssc="urn:ietf:params:xml:ns:dssc"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      targetNamespace="urn:ietf:params:xml:ns:dssc"
```

```
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
         schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
         schemaLocation="xmldsig-core-schema.xsd"/>
<xs:element name="SecuritySuitabilityPolicy"
          type="dssc:SecuritySuitabilityPolicyType"/>
<xs:complexType name="SecuritySuitabilityPolicyType">
  <xs:sequence>
    <xs:element ref="dssc:PolicyName"/>
    <xs:element ref="dssc:Publisher"/>
    <xs:element name="PolicyIssueDate" type="xs:dateTime"/>
    <xs:element name="NextUpdate" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="Usage" type="xs:string" minOccurs="0"/>
    <xs:element ref="dssc:Algorithm" maxOccurs="unbounded"/>
    <xs:element ref="ds:Signature" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" default="1"/>
  <xs:attribute name="lang" default="en"/>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
<xs:element name="PolicyName" type="dssc:PolicyNameType"/>
<xs:complexType name="PolicyNameType">
  <xs:sequence>
    <xs:element ref="dssc:Name"/>
    <xs:element ref="dssc:ObjectIdentifier" minOccurs="0"/>
    <xs:element ref="dssc:URI" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Publisher" type="dssc:PublisherType"/>
<xs:complexType name="PublisherType">
  <xs:sequence>
    <xs:element ref="dssc:Name"/>
    <xs:element name="Address" type="xs:string" minOccurs="0"/>
    <xs:element ref="dssc:URI" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Name" type="xs:string"/>
<xs:element name="ObjectIdentifier">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="(\d+\.)+\d+"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="URI" type="xs:anyURI"/>
<xs:element name="Algorithm" type="dssc:AlgorithmType"/>
<xs:complexType name="AlgorithmType">
  <xs:sequence>
    <xs:element ref="dssc:AlgorithmIdentifier"/>
    <xs:element ref="dssc:Evaluation" maxOccurs="unbounded"/>
    <xs:element ref="dssc:Information" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="AlgorithmIdentifier" type="dssc:AlgorithmIdentifierType"/>
<xs:complexType name="AlgorithmIdentifierType">
  <xs:sequence>
    <xs:element ref="dssc:Name"/>
    <xs:element ref="dssc:ObjectIdentifier" maxOccurs="unbounded"/>
```

```
      <xs:element ref="dssc:URI" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Validity" type="dssc:ValidityType"/>
  <xs:complexType name="ValidityType">
    <xs:sequence>
      <xs:element name="Start" type="xs:date" minOccurs="0"/>
      <xs:element name="End" type="xs:date" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Information" type="dssc:InformationType"/>
  <xs:complexType name="InformationType">
    <xs:sequence>
      <xs:element name="Text" type="xs:string" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Evaluation" type="dssc:EvaluationType"/>
  <xs:complexType name="EvaluationType">
    <xs:sequence>
      <xs:element ref="dssc:Parameter" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="dssc:Validity"/>
      <xs:any namespace="##other" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Parameter" type="dssc:ParameterType"/>
  <xs:complexType name="ParameterType">
    <xs:sequence>
      <xs:element name="Min" type="xs:int" minOccurs="0"/>
      <xs:element name="Max" type="xs:int" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

Appendice C. Module ASN.1 en syntaxe 1988 (pour information)

Module ASN.1

DSSC {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) ltans(11) id-mod(0) id-mod-dssc88(6) id-mod-dssc88-v1(1) }

```
  DEFINITIONS IMPLICIT TAGS ::=
  BEGIN

  -- EXPORT ALL --

  IMPORTS

  -- Import from RFC 5280 [RFC5280]
  -- Delete following import statement
  -- if "new" types are supported

  UTF8String FROM PKIX1Explicit88
          { iso(1) identified-organization(3) dod(6)
          internet(1) security(5) mechanisms(5) pkix(7)
          mod(0) pkix1-explicit(18) }

  -- Import from RFC 5652 [RFC5652]
```

```
ContentInfo FROM CryptographicMessageSyntax2004
        { iso(1) member-body(2) us(840)
        rsadsi(113549) pkcs(1) pkcs-9(9)
        smime(16) modules(0) cms-2004(24)}

;

SecuritySuitabilityPolicy ::= ContentInfo

-- contentType is id-signedData as defined in [RFC5652]
-- content is SignedData as defined in [RFC5652]
-- eContentType within SignedData is id-ct-dssc
-- eContent within SignedData is TBSPolicy

id-ct-dssc  OBJECT IDENTIFIER ::= {
        iso(1) identified-organization(3) dod(6)
        internet(1) security(5) mechanisms(5)
        ltans(11) id-ct(1) id-ct-dssc-tbsPolicy(6) }

TBSPolicy ::= SEQUENCE {
    version         INTEGER             DEFAULT   {v1(1)},
    language        UTF8String          DEFAULT   "en",
    policyName      PolicyName,
    publisher       Publisher,
    policyIssueDate  GeneralizedTime,
    nextUpdate      GeneralizedTime      OPTIONAL,
    usage           UTF8String          OPTIONAL,
    algorithms      SEQUENCE OF Algorithm
}

PolicyName ::= SEQUENCE {
    name  UTF8String,
    oid   OBJECT IDENTIFIER OPTIONAL,
    uri   IA5String        OPTIONAL
}

Publisher ::= SEQUENCE {
    name       UTF8String,
    address [0] UTF8String  OPTIONAL,
    uri     [1] IA5String   OPTIONAL
}

Algorithm ::= SEQUENCE {
    algorithmIdentifier     AlgID,
    evaluations             SEQUENCE OF Evaluation,
    information         [0] SEQUENCE OF UTF8String  OPTIONAL,
    other               [1] Extension               OPTIONAL
}

Extension ::= SEQUENCE {
    extensionType           OBJECT IDENTIFIER,
    extension               ANY DEFINED BY extensionType
}

AlgID ::= SEQUENCE {
    name      UTF8String,
    oid   [0] SEQUENCE OF OBJECT IDENTIFIER,
    uri   [1] SEQUENCE OF IA5String         OPTIONAL
}
```

```
Evaluation ::= SEQUENCE {
    parameters      [0] SEQUENCE OF Parameter OPTIONAL,
    validity         [1] Validity,
    other            [2] Extension          OPTIONAL
}

Parameter ::= SEQUENCE {
    name      UTF8String,
    min    [0] INTEGER    OPTIONAL,
    max    [1] INTEGER    OPTIONAL,
    other  [2] Extension      OPTIONAL
}

Validity ::= SEQUENCE {
    start  [0] GeneralizedTime OPTIONAL,
    end    [1] GeneralizedTime  OPTIONAL
}

END
```

Appendice D. Module ASN.1 en syntaxe 1997 (normatif)

Module ASN.1

DSSC {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) ltans(11) id-mod(0) id-mod-dssc(7) id-mod-dssc-v1(1) }

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORT ALL --

IMPORTS

-- Import from RFC 5280 [RFC5280]
-- Delete following import statement
-- if "new" types are supported

UTF8String FROM PKIX1Explicit88
        { iso(1) identified-organization(3) dod(6)
        internet(1) security(5) mechanisms(5) pkix(7)
        mod(0) pkix1-explicit(18) }

-- Import from RFC 5652 [RFC5652]

ContentInfo FROM CryptographicMessageSyntax2004
        { iso(1) member-body(2) us(840)
        rsadsi(113549) pkcs(1) pkcs-9(9)
        smime(16) modules(0) cms-2004(24)}
;

SecuritySuitabilityPolicy ::= ContentInfo

-- contentType is id-signedData as defined in [RFC5652]
-- content is SignedData as defined in [RFC5652]
-- eContentType within SignedData is id-ct-dssc
-- eContent within SignedData is TBSPolicy

id-ct-dssc  OBJECT IDENTIFIER ::= {
```

```
        iso(1) identified-organization(3) dod(6)
        internet(1) security(5) mechanisms(5)
        ltans(11) id-ct(1) id-ct-dssc-tbsPolicy(6) }

TBSPolicy ::= SEQUENCE {
    version         INTEGER             DEFAULT   {v1(1)},
    language        UTF8String          DEFAULT   "en",
    policyName      PolicyName,
    publisher       Publisher,
    policyIssueDate  GeneralizedTime,
    nextUpdate      GeneralizedTime      OPTIONAL,
    usage           UTF8String           OPTIONAL,
    algorithms      SEQUENCE OF Algorithm
}

PolicyName ::= SEQUENCE {
    name  UTF8String,
    oid   OBJECT IDENTIFIER OPTIONAL,
    uri   IA5String         OPTIONAL
}

Publisher ::= SEQUENCE {
    name        UTF8String,
    address  [0] UTF8String  OPTIONAL,
    uri      [1] IA5String   OPTIONAL
}

Algorithm ::= SEQUENCE {
    algorithmIdentifier     AlgID,
    evaluations             SEQUENCE OF Evaluation,
    information         [0] SEQUENCE OF UTF8String  OPTIONAL,
    other               [1] Extension               OPTIONAL
}

Extension ::= SEQUENCE {
    extensionType  EXTENSION-TYPE.&id ({SupportedExtensions}),
    extension      EXTENSION-TYPE.&Type
                   ({SupportedExtensions}{@extensionType})
}

EXTENSION-TYPE ::= TYPE-IDENTIFIER

SupportedExtensions EXTENSION-TYPE ::= {...}

AlgID ::= SEQUENCE {
    name      UTF8String,
    oid   [0] SEQUENCE OF OBJECT IDENTIFIER,
    uri   [1] SEQUENCE OF IA5String         OPTIONAL
}

Evaluation ::= SEQUENCE {
    parameters      [0] SEQUENCE OF Parameter OPTIONAL,
    validity        [1] Validity,
    other           [2] Extension           OPTIONAL
}

Parameter ::= SEQUENCE {
    name       UTF8String,
    min    [0] INTEGER     OPTIONAL,
    max    [1] INTEGER     OPTIONAL,
    other  [2] Extension   OPTIONAL
```

```
}

Validity ::= SEQUENCE {
    start  [0] GeneralizedTime OPTIONAL,
    end    [1] GeneralizedTime OPTIONAL
}

END
```

Appendice E. Exemple

L'exemple suivant montre une politique qui peut être utilisée pour la vérification de signature. Elle contient des algorithmes de hachage, des algorithmes de clé publique, et des schémas de signature. SHA-1 ainsi que RSA avec une longueur de module de 1024 sont des exemples d'algorithmes périmés.

```
<SecuritySuitabilityPolicy xmlns="urn:ietf:params:xml:ns:dssc"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <PolicyName>
  <Name>Evaluation of cryptographic algorithms</Name>
 </PolicyName>
 <Publisher>
  <Name>Some Evaluation Authority</Name>
 </Publisher>
 <PolicyIssueDate>2009-01-01T00:00:00</PolicyIssueDate>
 <Usage>Digital signature verification</Usage>
 <Algorithm>
  <AlgorithmIdentifier>
   <Name>SHA-1</Name>
   <ObjectIdentifier>1.3.14.3.2.26</ObjectIdentifier>
  </AlgorithmIdentifier>
  <Evaluation>
   <Validity>
    <End>2008-06-30</End>
   </Validity>
  </Evaluation>
 </Algorithm>
 <Algorithm>
  <AlgorithmIdentifier>
   <Name>SHA-256</Name>
   <ObjectIdentifier>2.16.840.1.101.3.4.2.1</ObjectIdentifier>
  </AlgorithmIdentifier>
  <Evaluation>
   <Validity>
    <End>2014-12-31</End>
   </Validity>
  </Evaluation>
 </Algorithm>
 <Algorithm>
  <AlgorithmIdentifier>
   <Name>SHA-512</Name>
   <ObjectIdentifier>2.16.840.1.101.3.4.2.3</ObjectIdentifier>
  </AlgorithmIdentifier>
  <Evaluation>
   <Validity>
    <End>2014-12-31</End>
   </Validity>
  </Evaluation>
 </Algorithm>
 <Algorithm>
```

```
 <AlgorithmIdentifier>
  <Name>RSA</Name>
  <ObjectIdentifier>1.2.840.113549.1.1.1</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>1024</Min>
  </Parameter>
  <Validity>
   <End>2008-03-31</End>
  </Validity>
 </Evaluation>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>2048</Min>
  </Parameter>
  <Validity>
   <End>2014-12-31</End>
  </Validity>
 </Evaluation>
</Algorithm>
<Algorithm>
 <AlgorithmIdentifier>
  <Name>DSA</Name>
  <ObjectIdentifier>1.2.840.10040.4.1</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="plength">
   <Min>1024</Min>
  </Parameter>
  <Parameter name="qlength">
   <Min>160</Min>
  </Parameter>
  <Validity>
   <End>2007-12-31</End>
  </Validity>
 </Evaluation>
 <Evaluation>
  <Parameter name="plength">
   <Min>2048</Min>
  </Parameter>
  <Parameter name="qlength">
   <Min>224</Min>
  </Parameter>
  <Validity>
   <End>2014-12-31</End>
  </Validity>
 </Evaluation>
</Algorithm>
<Algorithm>
 <AlgorithmIdentifier>
  <Name>PKCS#1 v1.5 SHA-1 with RSA</Name>
  <ObjectIdentifier>1.2.840.113549.1.1.5</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>1024</Min>
  </Parameter>
  <Validity>
   <End>2008-03-31</End>
  </Validity>
```

```
 </Evaluation>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>2048</Min>
  </Parameter>
  <Validity>
   <End>2008-06-30</End>
  </Validity>
 </Evaluation>
</Algorithm>
<Algorithm>
 <AlgorithmIdentifier>
  <Name>PKCS#1 v1.5 SHA-256 with RSA</Name>
  <ObjectIdentifier>1.2.840.113549.1.1.11</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>1024</Min>
  </Parameter>
  <Validity>
   <End>2008-03-31</End>
  </Validity>
 </Evaluation>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>2048</Min>
  </Parameter>
  <Validity>
   <End>2014-12-31</End>
  </Validity>
 </Evaluation>
</Algorithm>
<Algorithm>
 <AlgorithmIdentifier>
  <Name>PKCS#1 v1.5 SHA-512 with RSA</Name>
  <ObjectIdentifier>1.2.840.113549.1.1.13</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>1024</Min>
  </Parameter>
  <Validity>
   <End>2008-03-31</End>
  </Validity>
 </Evaluation>
 <Evaluation>
  <Parameter name="moduluslength">
   <Min>2048</Min>
  </Parameter>
  <Validity>
   <End>2014-12-31</End>
  </Validity>
 </Evaluation>
</Algorithm>
<Algorithm>
 <AlgorithmIdentifier>
  <Name>SHA-1 with DSA</Name>
  <ObjectIdentifier>1.2.840.10040.4.3</ObjectIdentifier>
 </AlgorithmIdentifier>
 <Evaluation>
  <Parameter name="plength">
```

```
    <Min>1024</Min>
   </Parameter>
   <Parameter name="qlength">
    <Min>160</Min>
   </Parameter>
   <Validity>
    <End>2007-12-31</End>
   </Validity>
  </Evaluation>
  <Evaluation>
   <Parameter name="plength">
    <Min>2048</Min>
   </Parameter>
   <Parameter name="qlength">
    <Min>224</Min>
   </Parameter>
   <Validity>
    <End>2008-06-30</End>
   </Validity>
  </Evaluation>
 </Algorithm>
</SecuritySuitabilityPolicy>
```

Adresse des auteurs

Thomas Kunz
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
Darmstadt D-64295
Germany
mél : thomas.kunz@sit.fraunhofer.de

Susanne Okunick
pawisda systems GmbH
Robert-Koch-Strasse 9
Weiterstadt D-64331
Germany
mél : susanne.okunick@pawisda.de

Ulrich Pordesch
Fraunhofer Gesellschaft
Rheinstrasse 75
Darmstadt D-64295
Germany
mél : ulrich.pordesch@zv.fraunhofer.de