Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 6931
RFC rendue obsolète : 4051
Catégorie : Sur la voie de la normalisation
ISSN : 2070-1721

D. Eastlake 3rd, Huawei

avril 2013
Traduction Claude Brière de L'Isle

Identifiants de ressource universels supplémentaires pour la sécurité de XML

Résumé
Le présent document étend, met à jour, et établit un registre IANA pour la liste des URI destinés à être utilisés avec des signatures numériques XML, du chiffrement de la canonisation, et de la gestion de clé. Ces URI identifient des algorithmes et des types d'informations. Le présent document rend obsolète la RFC 4051.

Statut de ce mémoire
Ceci est un document de l'Internet en cours de 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/rfc6931

Table des matières

1. Introduction

Les signatures numériques XML, leur canonisation, et leur chiffrement ont été normalisés par le W3C et le groupe de travail conjoint IETF/W3C XMLDSIG. Tout cela fait maintenant l'objet de Recommandations du W3C et certaines d'entre elles sont aussi des RFC. Voici celles qui sont disponibles :

Statut de RFC	Recommandation W3C	sujet
[RFC3275] Projet de norme	[XMLDSIG10]	Signatures numériques XML
[RFC3076] pour information	[CANON10]	XML canonique
- - - - - -	[XMLENC10]	Chiffrement XML 1.0
[RFC3741] pour information	[XCANON]	Canonisation XML 1.0 exclusive

Tous ces documents et recommandations utilisent des URI [RFC3986] pour identifier des algorithmes et des types d'informations de chiffrement. Le W3C a ensuite produit des versions mises à jour des spécifications de signature XML 1.1 [XMLDSIG11], XML 1.1 canonique [CANON11], et chiffrement XML 1.1 [XMLENC11], ainsi que une nouvelle spécification des propriétés de signature XML [XMLDSIG-PROP].

Tous les noms d'éléments en notation chameau *(camel-case)* comme DigestValue, sont tirés de ces documents.

Le présent document est une liste de référence pratique mise à jour des URI et des algorithmes correspondants pour lesquels de l'interêt a été exprimé. Comme la liste précédente [RFC4051] a été produite en 2005, des algorithmes de chiffrement significatifs intéressant la sécurité XML, dont pour certains l'URI n'est spécifié que dans le présent document, ont été ajoutés.

Le présent document rend obsolète la [RFC4051]. Tous les URI apparaissent dans les index de la Section 4. Seuls les URI qui ont été ajoutés par la [RFC4051] ou le présent document ont un paragraphe dans les Sections 2 ou 3, à l'exception de la canonisation minimale (paragraphe 2.4), par exemple, l'utilisation de SHA-256 est définie dans [XMLENC11] et donc il n'y a pas de paragraphe sur cet algorithme ici, mais son URI est inclus dans les index de la Section 4.

La spécification dans le présent document de l'URI représentant un algorithme n'implique pas l'homologation de cet algorithme pour un objet particulier. Une spécification de protocole, ce qu'il n'est pas, donne généralement les exigences pour l'algorithme et sa mise en œuvre pour le protocole. Les considérations de sécurité pour les algorithmes sont en évolution constante, et sont documentées ailleurs. La présente spécification donne simplement quelques URI et le formatage pertinent quand ces URI sont utilisés.

Noter que les progrès de la signature numérique XML [RFC3275] sur la voie de la normalisation exigent la suppression de tous les algorithmes de la version originale [RFC3075] pour laquelle l'interopérabilité n'a pas été démontrée. Cela exige la suppression de l'algorithme de canonisation minimale, pour lequel l'interêt paraît se maintenir. L'URI pour la canonisation minimale était inclus dans la [RFC4051] et il est inclus ici.

1.1 Terminologie

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

Le présent document n'est destiné à changer les exigences de mise en œuvre des algorithmes d'aucun document de l'IETF ou du W3C. L'utilisation de la terminologie de la [RFC2119] est seulement destinée à ce qui est déja déclaré ou impliqué par les autres documents d'autorité.

1.2 Acronymes

Les acronymes suivants sont utilisés dans le présent document :
HMAC *(Keyed-Hashing MAC)* code d'authentification de message par hachage de clé [RFC2104]
IETF *(Internet Engineering Task Force)* équipe d'ingénierie de l'Internet < www.ietf.org >
MAC *(Message Authentication Code)* code d'authentification de message
MD *(Message Digest)* résumé de message

NIST *(United States National Institute of Standards et Technology)* institut national américain des normes et technologies < www.nist.gov >
RC *(Rivest Cipher)* chiffrement de Rivest
RSA : Rivest, Shamir,et Adleman
SHA *(Secure Hash Algorithm)* algorithme de chiffrement irréversible
URI *(Uniform Resource Identifier)* identifiant de ressource universel [RFC3986]
W3C *(World Wide Web Consortium)* consortium de la Toile mondiale < www.w3.org >
XML *(eXtensible Markup Language)* langage de balisage extensible

2. Algorithmes

L'URI [RFC3986] qui a été éliminé de la norme de signature numérique XML à cause de la transition du statut de proposition de norme à celui de projet de norme [RFC3275] est inclus au paragraphe 2.4 ci dessous avec son préfixe d'origine, http://www.w3.org/2000/09/xmldsig# de façon à éviter de changer l'espace de noms des normes XMLDSIG.

Des algorithmes supplémentaires étaient donnés dans la [RFC4051] avec des URI qui commençaient par http://www.w3.org/2001/04/xmldsig-more# tandis que d'autres algorithmes ont été ajoutés dans ce document avec des URI qui commencent par http://www.w3.org/2007/05/xmldsig-more#

De plus, pour faciliter les références, le présent document inclut dans les index de la Section 4 de nombreux URI d'algorithmes de chiffrement provenant de plusieurs documents de sécurité XML qui utilisent les espaces de noms qui sont définis dans ces documents. Par exemple, 2000/09/xmldsig# pour certains URI spécifiés dans la [RFC3275] et 2001/04/xmlenc# pour des URI spécifiés dans [XMLENC10]. Voir aussi [XMLSECXREF].

2.1 Algorithmes de méthode de résumé (hachage)

Ces algorithmes sont utilisables chaque fois qu'un élément DigestMethod *(méthode de résumé)* survient.

2.1.1 MD5

Identifiant : http://www.w3.org/2001/04/xmldsig-more#md5

L'algorithme MD5 [RFC1321] ne prend pas de paramètre explicite. Un exemple d'élément DigestAlgorithm MD5 est :

```
<DigestAlgorithm
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#md5"/>
```

Un résumé MD5 est une chaîne de 128 bits. Le contenu de l'élément DigestValue DOIT être le codage base64 [RFC2045] de cette chaîne binaire vue comme un flux de 16 octets. Voir dans la [RFC6151] les considérations de sécurité pour MD5.

2.1.2 SHA-224

Identifiant : http://www.w3.org/2001/04/xmldsig-more#sha224

L'algorithme SHA-224 [FIPS180-4] [RFC6234] ne prend pas de paramètre explicite. Un exemple d'élément DigestAlgorithm SHA-224 est :

```
<DigestAlgorithm
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha224" />
```

Un résumé SHA-224 est une chaîne de 224 bits. Le contenu de l'élément DigestValue DOIT être le codage base64 [RFC2045] de cette chaîne vue comme un flux de 28 octets.

2.1.3 SHA-384

Identifiant : http://www.w3.org/2001/04/xmldsig-more#sha384

L'algorithme SHA-384 [FIPS180-4] ne prend pas de paramètre explicite. Un exemple d'élément DigestAlgorithm SHA-384 est :

```
<DigestAlgorithm
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384" />
```

Un résumé SHA-384 est une chaîne de 384 bits. Le contenu de l'élément DigestValue DOIT être le codage base64 [RFC2045] de cette chaîne vue comme un flux de 48 octets.

2.1.4 Whirlpool

Identifiant : http://www.w3.org/2007/05/xmldsig-more#whirlpool

L'algorithme Whirlpool [10118-3] ne prend pas de paramètre explicite. Un résumé Whirlpool est une chaîne de 512 bits. Le contenu de l'élément DigestValue DOIT être le codage base64 [RFC2045] de cette chaîne vue comme un flux de 64 octets.

2.1.5 Nouvelles fonctions SHA

Identifiants :
http://www.w3.org/2007/05/xmldsig-more#sha3-224
http://www.w3.org/2007/05/xmldsig-more#sha3-256
http://www.w3.org/2007/05/xmldsig-more#sha3-384
http://www.w3.org/2007/05/xmldsig-more#sha3-512

Le NIST a récemment achevé une compétition de fonctions de hachage pour une solution de remplacement à la famille SHA. L'algorihme Keccak-f[1600] a été choisi [Keccak], [SHA-3]. Cette fonction de hachage est couramment appelée "SHA-3", et ce paragraphe contient l'espace de réservation des URI futurs pour les informations sur l'utilisation de Keccak dans la sécurité XML.

Un résumé SHA-3 224, 256, 384, et 512 est une chaîne de respectivement 224, 256, 384, et 512 bits. Le contenu de l'élément DigestValue DOIT être le codage base64 [RFC2045] de cette chaîne vue comme un flux de respectivement 28, 32, 48, et 64 octets.

2.2 Algorithmes de MAC SignatureMethod

Ce paragraphe couvre les algorithmes de code d'authentification de message *(Message Authentication Code)* de méthode de signature.

Note : Du texte de ce paragraphe est dupliqué de la [RFC3275] pour l'agrément du lecteur. La RFC 3275 est normative en cas de conflit.

2.2.1 HMAC-MD5

Identifiant : http://www.w3.org/2001/04/xmldsig-more#hmac-md5

L'algorithme HMAC [RFC2104] prend comme paramètre la longueur de troncature en bits ; si le paramètre n'est pas spécifié, alors tous les bits du hachage sont sortis. Un exemple d'élément SignatureMethod HMAC-MD5 est le suivant :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-md5">
    <HMACOutputLength>112</HMACOutputLength>
  </SignatureMethod>
```

Le résultat de l'algorithme HMAC est finalement le résultat (éventuellement tronqué) de l'algorithme de résumé choisi. Cette valeur DOIT être codée en base64 [RFC2045] de la même façon directe que le résultat des algorithmes de résumé. Exemple : l'élément SignatureValue pour le résumé HMAC-MD5 9294727A 3638BB1C 13F48EF8 158BFC9D provenant des valeurs d'essai de la [RFC2104] serait kpRyejY4uxwT9I74FYv8nQ==

Définition de schéma :

```
<simpleType name="HMACOutputLength">
    <restriction base="entier"/>
  </simpleType>
```

DTD : <!ELEMENT HMACOutputLength (#PCDATA) >

La définition de schéma et le DTD immédiatement ci-dessus sont copiés de la [RFC3275].

Voir dans la [RFC6151] les considérations de sécurité pour HMAC-MD5.

2.2.2 Variantes de HMAC SHA

Identifiants :
http://www.w3.org/2001/04/xmldsig-more#hmac-sha224
http://www.w3.org/2001/04/xmldsig-more#hmac-sha256
http://www.w3.org/2001/04/xmldsig-more#hmac-sha384
http://www.w3.org/2001/04/xmldsig-more#hmac-sha512

SHA-224, SHA-256, SHA-384, et SHA-512 [FIPS180-4], [RFC6234] peuvent aussi être utilisés dans HMAC comme décrit au paragraphe 2.2.1 ci-dessus pour HMAC-MD5.

2.2.3 HMAC-RIPEMD160

Identifiant : http://www.w3.org/2001/04/xmldsig-more#hmac-ripemd160

RIPEMD-160 [10118-3] peut aussi être utilisé dans HMAC comme décrit au paragraphe 2.2.1 ci-dessus pour HMAC-MD5.

2.3 Algorithmes de signature à clé publique SignatureMethod

Ces algorithmes se distinguent de ceux du paragraphe 2.2 ci-dessus en ce qu'ils utilisent des méthodes à clé publique. C'est-à-dire que la clé de vérification est différente de la clé de signature et ne peut en être déduite.

2.3.1 RSA-MD5

Identifiant : http://www.w3.org/2001/04/xmldsig-more#rsa-md5

Cela implique l'algorithme de bourrage PKCS#1 v1.5 décrit dans la [RFC3447]. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-md5" />
```

Le contenu de SignatureValue pour une signature RSA-MD5 est le codage en base64 [RFC2045] de la chaîne d'octets calculée selon le paragraphe 8.2.1 de la [RFC3447], génération de signature pour le schéma de signature RSASSA-PKCS1-v1_5. Comme spécifié dans la fonction EMSA-PKCS1-V1_5-ENCODE au paragraphe 9.2 de la [RFC3447], la valeur entrée à la fonction de signature DOIT contenir un identifiant d'objet d'algorithme ajouté en tête pour la fonction de hachage, mais la disponibilité d'un analyseur ASN.1 et la reconnaissance des OID n'est pas exigée d'un vérificateur de signature. La représentation PKCS#1 v1.5 apparaît comme : CRYPT (PAD (ASN.1 (OID, DIGEST (data))))

Noter que l'ASN.1 avec bourrage sera de la forme suivante : 01 | FF* | 00 | préfixe | hachage

La barre verticale ("|") représente l'enchaînement. "01", "FF", et "00" sont des octets fixes de la valeur hexadécimale correspondante, et l'astérisque ("*") après "FF" indique la répétition. "hachage" est le résumé MD5 des données. "préfixe" est le préfixe de désignation de l'algorithme MD5 en BER ASN.1 exigé dans PKCS n° 1 [RFC3447], c'est-à-dire:

hex 30 20 30 0c 06 08 2a 86 48 86 f7 0d 02 05 05 00 04 10

Ce préfixe est inclus pour faciliter l'utilisation de bibliothèques cryptographiques standard. L'octet FF DOIT être répété un nombre de fois suffisant pour que la valeur de la quantité chiffrée soit exactement d'un octet plus courte que le module RSA. Voir dans la [RFC6151] les considérations de sécurité pour MD5.

2.3.2 RSA-SHA256

Identifiant : http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

Cela implique l'algorithme de bourrage PKCS#1 v1.5 [RFC3447] comme décrit au paragraphe 2.3.1, mais avec le préfixe de désignateur d'algorithme ASN.1 BER SHA-256. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
```

2.3.3 RSA-SHA384

Identifiant : http://www.w3.org/2001/04/xmldsig-more#rsa-sha384

Cela implique l'algorithme de bourrage PKCS#1 v1.5 [RFC3447] comme décrit au paragraphe 2.3.1, mais avec le préfixe de désignateur d'algorithme ASN.1 BER SHA-384. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha384" />
```

Comme cela représente le même effort de calculer un résumé de message SHA-384 que pour SHA-512, il est suggéré d'utiliser de préférence RSA-SHA512 à RSA-SHA384 lorsque possible.

2.3.4 RSA-SHA512

Identifiant : http://www.w3.org/2001/04/xmldsig-more#rsa-sha512

Cela implique l'algorithme de bourrage PKCS#1 v1.5 [RFC3447] comme décrit au paragraphe 2.3.1, mais avec le préfixe de désignateur d'algorithme ASN.1 BER SHA-512. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512" />
```

2.3.5 RSA-RIPEMD160

Identifiant : http://www.w3.org/2001/04/xmldsig-more#rsa-ripemd160

Cela implique l'algorithme de bourrage PKCS#1 v1.5 [RFC3447] comme décrit au paragraphe 2.3.1, mais avec le préfixe de désignateur d'algorithme ASN.1 BER RIPEMD160. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-ripemd160" />
```

2.3.6 ECDSA-SHA*, ECDSA-RIPEMD160, ECDSA-Whirlpool

Identifiants :

http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512
http://www.w3.org/2007/05/xmldsig-more#ecdsa-ripemd160
http://www.w3.org/2007/05/xmldsig-more#ecdsa-whirlpool

L'algorithme de signature numérique à courbe elliptique (ECDSA, *Elliptic Curve Digital Signature Algorithm*) [FIPS180-4] est la courbe éliptique analogue de la méthode de signature de l'algorithme de signature numérique (DSA, *Digital Signature Algorithm*) c'est-à-dire, la norme de signature numérique (DSS, *Digital Signature Standard*). Il ne prend pas de paramètres explicites. Pour les spécifications détaillées de la façon dont il l'utilise avec les fonctions de hachage SHA et la signature numérique XML, voir [X9.62] et la [RFC4050]. Les fragments #ecdsa-ripemd160 et #ecdsa-whirlpool dans le nouvel espace de noms identifient une méthode de signature traitée de la même façon que spécifié par le fragment #ecdsa-sha1 de cet espace de noms, à l'exception que RIPEMD160 ou Whirlpool est utilisé à la place de SHA-1.

Le résultat de l'algorithme ECDSA consiste en une paire d'entiers généralement désignés comme la paire (r, s). La valeur de signature consiste en le codage en base64 de l'enchaînement de flux de deux octets qui résultent respectivement du codage en octets des valeurs r et s dans cet ordre. La conversion de l'entier en flux d'octets doit être faite conformément à

l'opération I2OSP définie dans la [RFC3447] avec le paramètre l égal à la taille de l'ordre du point de base de la courbe en octets (par exemple, 32 pour la courbe P-256 et 66 pour la courbe P-521 [FIPS186-3]).

Pour une introduction aux algorithmes de chiffrement par courbe elliptique, voir la [RFC6090] et noter l'errata (Errata ID 2773-2777).

2.3.7 ESIGN-SHA*

Identifiants :
http://www.w3.org/2001/04/xmldsig-more#esign-sha1
http://www.w3.org/2001/04/xmldsig-more#esign-sha224
http://www.w3.org/2001/04/xmldsig-more#esign-sha256
http://www.w3.org/2001/04/xmldsig-more#esign-sha384
http://www.w3.org/2001/04/xmldsig-more#esign-sha512

L'algorithme ESIGN spécifié dans [IEEEP1363a] est un schéma de signature fondé sur le problème de la factorisation d'entiers. Il est beaucoup plus rapide que les précédents schémas de signature numérique, de sorte que ESIGN peut être mis en œuvre sur des cartes à mémoire dans co-processeurs particuliers.

Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#esign-sha1" />
```

2.3.8 RSA-Whirlpool

Identifiant : http://www.w3.org/2007/05/xmldsig-more#rsa-whirlpool

Comme dans la définition de l'algorithme RSA-SHA1 dans [XMLDSIG11], la désignation "RSA" signifie l'algorithme RSASSA-PKCS1-v1_5 comme défini dans la [RFC3447]. Quand il est identifié par l'identifiant de fragment #rsa-whirlpool, Whirlpool est utilisé plutôt comme l'algorithme de hachage. L'utilisation de la désignation BER ASN.1 d'algorithme Whirlpool est implicire. Cette désignation est : hex 30 4e 30 0a 06 06 28 cf 06 03 00 37 05 00 04 40 comme une séquence d'octets explicite. Cela correspond à l'OID 1.0.10118.3.0.55 défini dans [10118-3].

Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-whirlpool" />
```

2.3.9 RSASSA-PSS avec paramètres

Identifiants :
http://www.w3.org/2007/05/xmldsig-more#rsa-pss
http://www.w3.org/2007/05/xmldsig-more#MGF1

Ces identifiants impliquent l'algorithme de codage PKCS#1 EMSA-PSS [RFC3447]. L'algorithme RSASSA-PSS prend la méthode du résumé (fonction de hachage), une fonction de génération de gabarit, la longueur de sel en octets (SaltLength), et le champ d'en queue, comme paramètres explicites.

Les identifiants d'algorithme pour les fonctions de hachage spécifiées dans le chiffrement XML [XMLENC11] [XMLDSIG11] et au paragraphe 2.1 sont considérés comme des identifiants d'algorithme valides pour les fonctions de hachage. Conformément à la [RFC3447], la valeur par défaut pour la fonction de résumé est SHA-1, mais à cause de la faiblesse découverte dans SHA-1 [RFC6194], il est recommandé que SHA-256 ou une fonction de hachage plus forte soit utilisée. Malgré la [RFC3447], SHA-256 est à utiliser par défaut avec ces identifiants de méthode de signature si aucune fonction de hachage n'a été spécifiée.

La longueur de sel par défaut pour ces identifiants de méthode de signature si la longueur de sel n'est pas spécifiée DEVRA être le nombre d'octets de la valeur hachée de la méthode de résumé, comme recommandé dans la [RFC4055]. Dans une signature RSASSA-PSS paramétrée, les paramètres ds:DigestMethod et SaltLength apparaissent généralement. Si ils ne le font pas, les valeurs par défaut rendent ceci équivalent à http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1 (voir le paragraphe 2.3.10). Le champ d'en queue (TrailerField) prend par défaut 1 (0xBC) quand il est omis.

Définition de schéma (espace de nom cible http://www.w3.org/2007/05/xmldsig-more#) :

```
<xs:element name="RSAPSSParams" type="pss:RSAPSSParamsType">
    <xs:annotation>
      <xs:documentation>
Élément de niveau supérieur qui peut être utilisé dans xs:any namespace="#other" caractère générique de contenu ds:SignatureMethod.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="RSAPSSParamsType">
    <xs:sequence>
      <xs:element ref="ds:DigestMethod" minOccurs="0"/>
      <xs:element name="MaskGenerationFunction"
        type="pss:MaskGenerationFunctionType" minOccurs="0"/>
      <xs:element name="SaltLength" type="xs:int"
        minOccurs="0"/>
      <xs:element name="TrailerField" type="xs:int"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MaskGenerationFunctionType">
    <xs:sequence>
      <xs:element ref="ds:DigestMethod" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Algorithm" type="xs:anyURI"
      default="http://www.w3.org/2007/05/xmldsig-more#MGF1"/>
  </xs:complexType>
```

2.3.10 RSASSA-PSS sans paramètre

La [RFC3447] spécifie actuellement une seule fonction de génération de gabarit MGF1 fondée sur une fonction de hachage. Bien que la [RFC3447] permette des paramètres, on utilise par défaut la même fonction de hachage que la fonction de méthode de résumé. Cette seule approche par défaut est prise en charge dans ce paragraphe ; donc la définition d'un type de fonction de génération de gabarit n'est pas nécessaire pour l'instant. La même chose s'applique au champ d'en queue. Il y a seulement une valeur (0xBC) spécifiée dans la [RFC3447]. Donc, ce paramètre par défaut doit être utilisé pour la génération de la signature. La longueur de sel par défaut est la longueur de la fonction de hachage.

Identifiants :
http://www.w3.org/2007/05/xmldsig-more#sha3-224-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha3-256-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha3-384-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha3-512-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#md2-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#md5-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha1-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha224-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha384-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#ripemd128-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#ripemd160-rsa-MGF1
http://www.w3.org/2007/05/xmldsig-more#whirlpool-rsa-MGF1

Un exemple d'utilisation est :

```
<SignatureMethod
   Algorithm="http://www.w3.org/2007/05/xmldsig-more#SHA3-256-rsa-MGF1"/>
```

2.3.11 RSA-SHA224

Identifiant : http://www.w3.org/2007/05/xmldsig-more#rsa-sha224

Cela implique l'algorithme de bourrage PKCS#1 v1.5 [RFC3447] comme décrit au paragraphe 2.3.1, mais avec le préfixe de désignation d'algorithme ASN.1 BER SHA-224. Un exemple d'utilisation est :

```
<SignatureMethod
    Algorithm="http://www.w3.org/2007/05/xmldsig-more#rsa-sha224" />
```

Parce que c'est à peu près le même effort de calculer un résumé de message SHA-224 et un résumé de message SHA-256, il est suggéré que RSA-SHA256 soit utilisé de préférence à RSA-SHA224 lorsque possible.

2.4 Canonisation minimale

Jusqu'à présent, deux mises en œuvre indépendantes interopérables de canonisation minimale n'ont pas été annoncées. Donc, quand la signature numérique XML a avancé sur la voie de la normalisation de la [RFC3075] à la [RFC3275], la canonisation minimale a été abandonnée. Cependant, il y a toujours de l'intérêt pour le sujet. Pour sa définition, voir le paragraphe 6.5.1 de la [RFC3075].

Pour référence, son identifiant reste : http://www.w3.org/2000/09/xmldsig#minimal

2.5 Algorithmes de transformation

Noter que tous les algorithmes de méthode de canonisation peuvent aussi être utilisés comme algorithmes de transformation.

2.5.1 XPointer

Identifiant : http://www.w3.org/2001/04/xmldsig-more#xptr

Cet algorithme de transformation prend un [XPointer] comme paramètre explicite. Un exemple d'utilisation est :

```
<Transform
  Algorithm="http://www.w3.org/2001/04/xmldsig-more/xptr">
  <XPointer
    xmlns="http://www.w3.org/2001/04/xmldsig-more/xptr">
      xpointer(id("foo")) xmlns(bar=http://foobar.example)
      xpointer(//bar:Zab[@Id="foo"])
  </XPointer>
</Transform>
```

Définition de schéma : <element name="XPointer" type="string"/>

DTD : <!ELEMENT XPointer (#PCDATA) >

L'entrée à cette transformation est une chaîne d'octets (qui est ensuite analysée en XML).

Le résultat de cette transformation est un ensemble de nœuds ; les résultats du XPointer sont traités comme défini dans la spécification XMLDSIG [RFC3275] pour un XPointer same-document.

2.6 Algorithmes EncryptionMethod

Ce paragraphe donne les identifiants et les informations pour plusieurs algorithmes de méthode de chiffrement.

2.6.1 Algorithme de chiffrement ARCFOUR

Identifiant : http://www.w3.org/2001/04/xmldsig-more#arcfour

ARCFOUR est un algorithme de chiffrement de flux rapide et simple qui est compatible avec l'algorithme RCA de RSA Security [RC4]. Un exemple d'élément EncryptionMethod qui utilise ARCFOUR est :

```
<EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmldsig-more#arcfour">
```

```
    <KeySize>40</KeySize>
  </EncryptionMethod>
```

Noter que Arcfour utilise le paramètre générique KeySize spécifié et défini dans [XMLENC11].

2.6.2 Chiffrement de bloc Camellia

Identifiants :
http://www.w3.org/2001/04/xmldsig-more#camellia128-cbc
http://www.w3.org/2001/04/xmldsig-more#camellia192-cbc
http://www.w3.org/2001/04/xmldsig-more#camellia256-cbc

Camellia est un chiffrement de bloc avec la même interface que l'AES [Camellia] [RFC3713]; il a une taille de bloc de 128 bits et des tailles de clé de 128, 192, et 256 bits. Dans le chiffrement XML, Camellia est utilisé de la même façon que l'AES : il est utilisé dans le mode de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) avec une valeur d'initialisation (IV) de 128 bits. Le texte chiffré résultant est précédé par l'IV. Si il est inclus dans un résultat XML, il est alors codé en base64. Un exemple de méthode de chiffrement Camellia est comme suit :

```
<EncryptionMethod
    Algorithm= "http://www.w3.org/2001/04/xmldsig-more#camellia128-cbc"/>
```

2.6.3 Enveloppe de clé Camellia

Identifiants :
http://www.w3.org/2001/04/xmldsig-more#kw-camellia128
http://www.w3.org/2001/04/xmldsig-more#kw-camellia192
http://www.w3.org/2001/04/xmldsig-more#kw-camellia256

L'enveloppe de clé Camellia [Camellia] [RFC3713] est identique à l'algorithme d'enveloppe de clé AES [RFC3394] spécifié dans la norme de chiffrement XML avec "AES" remplacé par "Camellia". Comme avec l'enveloppe de clé AES, la valeur de vérification est 0xA6A6A6A6A6A6A6A6.

L'algorithme est le même quelle que soit la taille de la clé Camellia utilisée dans l'enveloppement, appelée la "clé de chiffrement de clé" ou "KEK". Si Camellia est pris en charge, il est particulièrement suggéré que l'enveloppement des clés de 128 bits avec une KEK de 128 bits, et l'enveloppement des clés de 256 bits avec une KEK de 256 bits, soient pris en charge.

Un exemple d'utilisation est :

```
<EncryptionMethod
    Algorithm= "http://www.w3.org/2001/04/xmldsig-more#kw-camellia128"/>
```

2.6.4 PSEC-KEM

Identifiant : http://www.w3.org/2001/04/xmldsig-more#psec-kem

L'algorithme PSEC-KEM, spécifié dans [18033-2], est un mécanisme d'encapsulation de clé qui utilise le chiffrement par courbe elliptique. Un exemple de son utilisation est :

```
<EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#psec-kem">
    <ECParameters>
      <Version>version</Version>
      <FieldID>id</FieldID>
      <Curve>curve</Curve>
      <Base>base</Base>
      <Order>order</Order>
      <Cofactor>cofactor</Cofactor>
    </ECParameters>
  </EncryptionMethod>
```

Voir dans [18033-2] les informations sur les paramètres utilisés ci-dessus.

2.6.5 Chiffrement de bloc SEED

Identifiant : http://www.w3.org/2007/05/xmldsig-more#seed128-cbc

SEED [RFC4269] a une taille de bloc de 128 bits avec des tailles de clé de 128 bits. En chiffrement XML, SEED peut être utilisé en mode CBC avec une valeur d'initialisation (IV) de 128 bits. Le texte chiffré résultant est précédé de l'IV. Si il est inclus dans un résultat XML, il est alors codé en base64. Un exemple de méthode de chiffrement SEED est comme suit :

```
<EncryptionMethod
    Algorithm="http://www.w3.org/2007/05/xmldsig-more#seed128-cbc" />
```

2.6.6 Enveloppe de clé SEED

Identifiant : http://www.w3.org/2007/05/xmldsig-more#kw-seed128

L'enveloppe de clé avec SEED est identique à ce qui est indiqué au paragraphe 2.2.1 de la [RFC3394] avec "AES" remplacé par "SEED". L'algorithme est spécifié dans la [RFC4010]. La mise en œuvre de SEED est facultative. La valeur initiale par défaut est 0xA6A6A6A6A6A6A6A6. Un exemple d'utilisation est :

```
<EncryptionMethod
    Algorithm= "http://www.w3.org/2007/05/xmldsig-more#kw-seed128"/>
```

3. KeyInfo

On spécifie au paragraphe 3.1 ci-dessous un nouvel élément fils KeyInfo, et au paragraphe 3.2 des valeurs supplémentaires de type KeyInfo à utiliser dans RetrievalMethod *(méthodes de restitution)*.

3.1 Sac de certificats et de CRL PKCS n° 7

Des "signedData" *(données signées)* PKCS n° 7 [RFC2315] peuvent aussi être utilisées comme un sac de certificats et/ou de listes de révocation de certificats (CRL). L'élément PKCS7signedData est défini pour s'accommoder de telles structures au sein de KeyInfo. La structure binaire PKCS n° 7 est codée en base64 [RFC2045]. Toutes les informations de signataire présentes sont ignorées. Voici un exemple de la [RFC3092], sans les données en base64 :

```
<foo:PKCS7signedData
    xmlns:foo="http://www.w3.org/2001/04/xmldsig-more">
    ...
  </foo:PKCS7signedData>
```

3.2 Valeurs de types RetrievalMethod supplémentaires

L'attribut Type de RetrievalMethod est un identifiant facultatif pour le type de données à restituer. Le résultat du déréférencement d'une référence RetrievalMethod pour tous les types de KeyInfo avec une structure XML est un élément XML ou un document avec cet élément comme racine. Les divers types d'informations de clé "brutes" retournent une valeur binaire. Donc, elles exigent un attribut Type parce que elles ne sont pas anlysables sans ambiguïté.

Identifiants :

http://www.w3.org/2001/04/xmldsig-more#KeyName
http://www.w3.org/2001/04/xmldsig-more#KeyValue
http://www.w3.org/2001/04/xmldsig-more#PKCS7signedData
http://www.w3.org/2001/04/xmldsig-more#rawPGPKeyPacket
http://www.w3.org/2001/04/xmldsig-more#rawPKCS7signedData
http://www.w3.org/2001/04/xmldsig-more#rawSPKISexp
http://www.w3.org/2001/04/xmldsig-more#rawX509CRL
http://www.w3.org/2001/04/xmldsig-more#RetrievalMethod

4. Index

Les paragraphes suivants founissent un index par URI et par identifiant de fragment (la portion de l'URI après "#") des algorithmes d'URI et KeyInfo définis dans le présent document et dans les normes (plus le nom d'élément fils KeyInfo défini dans le présent document). La colonne "Paragraphe/document" donne le paragraphe du présent document ou, si ce n'est pas spécifié dans ce document, le document où l'élément est spécifié. Voir aussi [XMLSECXREF].

4.1 Index de fragment

La partie initiale "http://www.w3.org/" de l'URI n'est pas incluse ci-dessous. Les sept premières entrées ont un identifiant de fragment nul ou pas d'identifiant de fragment.

Fragment	URI	Paragraphe/document
	2002/06/xmldsig-filter2	[XPATH]
	2006/12/xmlc12n11#	[CANON11]
	TR/1999/REC-xslt-19991116	[XSLT]
	TR/1999/REC-xpath-19991116	[XPATH]
	TR/2001/06/xml-exc-c14n#	[XCANON]
	TR/2001/REC-xml-c14n-20010315	[CANON10]
	TR/2001/REC-xmlschema-1-20010502	[Schema]
aes128-cbc	2001/04/xmlenc#aes128-cbc	[XMLENC11]
aes128-gcm	2009/xmlenc11#aes128-gcm	[XMLENC11]
aes192-cbc	2001/04/xmlenc#aes192-cbc	[XMLENC11]
aes192-gcm	2009/xmlenc11#aes192-gcm	[XMLENC11]
aes256-cbc	2001/04/xmlenc#aes256-cbc	[XMLENC11]
aes256-gcm	2009/xmlenc11#aes256-gcm	[XMLENC11]
arcfour	2001/04/xmldsig-more#arcfour	2.6.1
base64	2000/09/xmldsig#base64	[RFC3275]
camellia128-cbc	2001/04/xmldsig-more#camellia128-cbc	2.6.2
camellia192-cbc	2001/04/xmldsig-more#camellia192-cbc	2.6.2
camellia256-cbc	2001/04/xmldsig-more#camellia256-cbc	2.6.2
ConcatKDF	2009/xmlenc11#ConcatKDF	[XMLENC11]
decrypt#XML	2002/07/decrypt#XML	[DECRYPT]
decrypt#Binary	2002/07/decrypt#Binary	[DECRYPT]
DEREncodedKeyValue	2009/xmldsig11#DEREncodedKeyValue	[XMLDSIG11]
dh	2001/04/xmlenc#dh	[XMLENC11]
dh-es	2009/xmlenc11#dh-es	[XMLENC11]
dsa-sha1	2000/09/xmldsig#dsa-sha1	[RFC3275]
dsa-sha256	2009/xmldsig11#dsa-sha256	[XMLDSIG11]
DSAKeyValue	2000/09/xmldsig#DSAKeyValue	[XMLDSIG11]
ECDH-ES	2009/xmlenc11#ECDH-ES	[XMLENC11]
ecdsa-ripemd160	2007/05/xmldsig-more#ecdsa-ripemd160	2.3.6
ecdsa-sha1	2001/04/xmldsig-more#ecdsa-sha1	2.3.6
ecdsa-sha224	2001/04/xmldsig-more#ecdsa-sha224	2.3.6
ecdsa-sha256	2001/04/xmldsig-more#ecdsa-sha256	2.3.6
ecdsa-sha384	2001/04/xmldsig-more#ecdsa-sha384	2.3.6
ecdsa-sha512	2001/04/xmldsig-more#ecdsa-sha512	2.3.6
ecdsa-whirlpool	2007/05/xmldsig-more#ecdsa-whirlpool	2.3.5
ecies-kem	2010/xmlsec-ghc#ecies-kem	[GENERIC]
ECKeyValue	2009/xmldsig11#ECKeyValue	[XMLDSIG11]
enveloped-signature	2000/09/xmldsig#enveloped-signature	[RFC3275]
esign-sha1	2001/04/xmldsig-more#esign-sha1	2.3.7
esign-sha224	2001/04/xmldsig-more#esign-sha224	2.3.7
esign-sha256	2001/04/xmldsig-more#esign-sha256	2.3.7
esign-sha384	2001/04/xmldsig-more#esign-sha384	2.3.7
esign-sha512	2001/04/xmldsig-more#esign-sha512	2.3.7

generic-hybrid	2010/xmlsec-ghc#generic-hybrid	[GENERIC]
hmac-md5	2001/04/xmldsig-more#hmac-md5	2.2.1
hmac-ripemd160	2001/04/xmldsig-more#hmac-ripemd160	2.2.3
hmac-sha1	2000/09/xmldsig#hmac-sha1	[RFC3275]
hmac-sha224	2001/04/xmldsig-more#hmac-sha224	2.2.2
hmac-sha256	2001/04/xmldsig-more#hmac-sha256	2.2.2
hmac-sha384	2001/04/xmldsig-more#hmac-sha384	2.2.2
hmac-sha512	2001/04/xmldsig-more#hmac-sha512	2.2.2
KeyName	2001/04/xmldsig-more#KeyName	3.2
KeyValue	2001/04/xmldsig-more#KeyValue	3.2
kw-aes128	2001/04/xmlenc#kw-aes128	[XMLENC11]
kw-aes128-pad	2009/xmlenc11#kw-aes-128-pad	[XMLENC11]
kw-aes192	2001/04/xmlenc#kw-aes192	[XMLENC11]
kw-aes192-pad	2009/xmlenc11#kw-aes-192-pad	[XMLENC11]
kw-aes256	2001/04/xmlenc#kw-aes256	[XMLENC11]
kw-aes256-pad	2009/xmlenc11#kw-aes-256-pad	[XMLENC11]
kw-camellia128	2001/04/xmldsig-more#kw-camellia128	2.6.3
kw-camellia192	2001/04/xmldsig-more#kw-camellia192	2.6.3
kw-camellia256	2001/04/xmldsig-more#kw-camellia256	2.6.3
kw-seed128	2007/05/xmldsig-more#kw-seed128	2.6.6
md2-rsa-MGF1	2007/05/xmldsig-more#md2-rsa-MGF1	2.3.10
md5	2001/04/xmldsig-more#md5	2.1.1
md5-rsa-MGF1	2007/05/xmldsig-more#md5-rsa-MGF1	2.3.10
MGF1	2007/05/xmldsig-more#MGF1	2.3.9
mgf1sha1	2009/xmlenc11#mgf1sha1	[XMLENC11]
mgf1sha224	2009/xmlenc11#mgf1sha224	[XMLENC11]
mgf1sha256	2009/xmlenc11#mgf1sha256	[XMLENC11]
mgf1sha384	2009/xmlenc11#mgf1sha384	[XMLENC11]
mgf1sha512	2009/xmlenc11#mgf1sha512	[XMLENC11]
MgmtData	2000/09/xmldsig#MgmtData	[XMLDSIG11]
minimal	2000/09/xmldsig#minimal	2.4
pbkdf2	2009/xmlenc11#pbkdf2	[XMLENC11]
PGPData	2000/09/xmldsig#PGPData	[XMLDSIG11]
PKCS7signedData	2001/04/xmldsig-more#PKCS7signedData	3.1
PKCS7signedData	2001/04/xmldsig-more#PKCS7signedData	3.2
psec-kem	2001/04/xmldsig-more#psec-kem	2.6.4
rawPGPKeyPacket	2001/04/xmldsig-more#rawPGPKeyPacket	3.2
rawPKCS7signedData	2001/04/xmldsig-more#rawPKCS7signedData	3.2
rawSPKISexp	2001/04/xmldsig-more#rawSPKISexp	3.2
rawX509Certificate	2000/09/xmldsig#rawX509Certificate	[RFC3275]
rawX509CRL	2001/04/xmldsig-more#rawX509CRL	3.2
RetrievalMethod	2001/04/xmldsig-more#RetrievalMethod	3.2
ripemd128-rsa-MGF1	2007/05/xmldsig-more#ripemd128-rsa-MGF1	2.3.10
ripemd160	2001/04/xmlenc#ripemd160	[XMLENC11]
ripemd160-rsa-MGF1	2007/05/xmldsig-more#ripemd160-rsa-MGF1	2.3.10
rsa-1_5	2001/04/xmlenc#rsa-1_5	[XMLENC11]
rsa-md5	2001/04/xmldsig-more#rsa-md5	2.3.1
rsa-oaep	2009/xmlenc11#rsa-oaep	[XMLENC11]
rsa-oaep-mgf1p	2001/04/xmlenc#rsa-oaep-mgf1p	[XMLENC11]
rsa-pss	2007/05/xmldsig-more#rsa-pss	2.3.9
rsa-ripemd160	2001/04/xmldsig-more#rsa-ripemd160	2.3.5
rsa-sha1	2000/09/xmldsig#rsa-sha1	[RFC3275]
rsa-sha224	2007/05/xmldsig-more#rsa-sha224	2.3.11
rsa-sha256	2001/04/xmldsig-more#rsa-sha256	2.3.2
rsa-sha384	2001/04/xmldsig-more#rsa-sha384	2.3.3
rsa-sha512	2001/04/xmldsig-more#rsa-sha512	2.3.4
rsa-whirlpool	2007/05/xmldsig-more#rsa-whirlpool	2.3.5
rsaes-kem	2010/xmlsec-ghc#rsaes-kem	[GENERIC]

RSAKeyValue	2000/09/xmldsig#RSAKeyValue	[XMLDSIG11]
seed128-cbc	2007/05/xmldsig-more#seed128-cbc	2.6.5
sha1	2000/09/xmldsig#sha1	[RFC3275]
sha1-rsa-MGF1	2007/05/xmldsig-more#sha1-rsa-MGF1	2.3.10
sha224	2001/04/xmldsig-more#sha224	2.1.2
sha224-rsa-MGF1	2007/05/xmldsig-more#sha224-rsa-MGF1	2.3.10
sha256	2001/04/xmlenc#sha256	[XMLENC11]
sha256-rsa-MGF1	2007/05/xmldsig-more#sha256-rsa-MGF1	2.3.10
sha3-224	2007/05/xmldsig-more#sha3-224	2.1.5
sha3-224-rsa-MGF1	2007/05/xmldsig-more#sha3-224-rsa-MGF1	2.3.10
sha3-256	2007/05/xmldsig-more#sha3-256	2.1.5
sha3-256-rsa-MGF1	2007/05/xmldsig-more#sha3-256-rsa-MGF1	2.3.10
sha3-384	2007/05/xmldsig-more#sha3-384	2.1.5
sha3-384-rsa-MGF1	2007/05/xmldsig-more#sha3-384-rsa-MGF1	2.3.10
sha3-512	2007/05/xmldsig-more#sha3-512	2.1.5
sha3-512-rsa-MGF1	2007/05/xmldsig-more#sha3-512-rsa-MGF1	2.3.10
sha384	2001/04/xmldsig-more#sha384	2.1.3
sha384-rsa-MGF1	2007/05/xmldsig-more#sha384-rsa-MGF1	2.3.10
sha512	2001/04/xmlenc#sha512	[XMLENC11]
sha512-rsa-MGF1	2007/05/xmldsig-more#sha512-rsa-MGF1	2.3.10
SPKIData	2000/09/xmldsig#SPKIData	[XMLDSIG11]
tripledes-cbc	2001/04/xmlenc#tripledes-cbc	[XMLENC11]
whirlpool	2007/05/xmldsig-more#whirlpool	2.1.4
whirlpool-rsa-MGF1	2007/05/xmldsig-more#whirlpool-rsa-MGF1	2.3.10
WithComments	2006/12/xmlc14n11#WithComments	[CANON11]
WithComments	TR/2001/06/xml-exc-c14n#WithComments	[XCANON]
WithComments	TR/2001/REC-xml-c14n-20010315#WithComments	[CANON10]
X509Data	2000/09/xmldsig#X509Data	[XMLDSIG11]
xptr	2001/04/xmldsig-more#xptr	2.5.1

La partie initiale "http://www.w3.org/" de l'URI n'est pas incluse ci-dessus.

4.2 Index d'URI

La partie initiale "http://www.w3.org/" de l'URI n'est pas incluse ci-dessous.

URI	Paragraphe/document	Type
2000/09/xmldsig#base64	[RFC3275]	Transform
2000/09/xmldsig#DSAKeyValue	[RFC3275]	Retrieval type
2000/09/xmldsig#dsa-sha1	[RFC3275]	SignatureMethod
2000/09/xmldsig#enveloped-signature	[RFC3275]	Transform
2000/09/xmldsig#hmac-sha1	[RFC3275]	SignatureMethod
2000/09/xmldsig#MgmtData	[RFC3275]	Retrieval type
2000/09/xmldsig#minimal	2.4	Canonicalization
2000/09/xmldsig#PGPData	[RFC3275]	Retrieval type
2000/09/xmldsig#rawX509Certificate	[RFC3275]	Retrieval type
2000/09/xmldsig#rsa-sha1	[RFC3275]	SignatureMethod
2000/09/xmldsig#RSAKeyValue	[RFC3275]	Retrieval type
2000/09/xmldsig#sha1	[RFC3275]	DigestAlgorithm
2000/09/xmldsig#SPKIData	[RFC3275]	Retrieval type
2000/09/xmldsig#X509Data	[RFC3275]	Retrieval type
2001/04/xmldsig-more#arcfour	2.6.1	EncryptionMethod
2001/04/xmldsig-more#camellia128-cbc	2.6.2	EncryptionMethod
2001/04/xmldsig-more#camellia192-cbc	2.6.2	EncryptionMethod
2001/04/xmldsig-more#camellia256-cbc	2.6.2	EncryptionMethod
2001/04/xmldsig-more#ecdsa-sha1	2.3.6	SignatureMethod
2001/04/xmldsig-more#ecdsa-sha224	2.3.6	SignatureMethod
2001/04/xmldsig-more#ecdsa-sha256	2.3.6	SignatureMethod

2001/04/xmldsig-more#ecdsa-sha384	2.3.6	SignatureMethod
2001/04/xmldsig-more#ecdsa-sha512	2.3.6	SignatureMethod
2001/04/xmldsig-more#esign-sha1	2.3.7	SignatureMethod
2001/04/xmldsig-more#esign-sha224	2.3.7	SignatureMethod
2001/04/xmldsig-more#esign-sha256	2.3.7	SignatureMethod
2001/04/xmldsig-more#esign-sha384	2.3.7	SignatureMethod
2001/04/xmldsig-more#esign-sha512	2.3.7	SignatureMethod
2001/04/xmldsig-more#hmac-md5	2.2.1	SignatureMethod
2001/04/xmldsig-more#hmac-ripemd160	2.2.3	SignatureMethod
2001/04/xmldsig-more#hmac-sha224	2.2.2	SignatureMethod
2001/04/xmldsig-more#hmac-sha256	2.2.2	SignatureMethod
2001/04/xmldsig-more#hmac-sha384	2.2.2	SignatureMethod
2001/04/xmldsig-more#hmac-sha512	2.2.2	SignatureMethod
2001/04/xmldsig-more#KeyName	3.2	Retrieval type
2001/04/xmldsig-more#KeyValue	3.2	Retrieval type
2001/04/xmldsig-more#kw-camellia128	2.6.3	EncryptionMethod
2001/04/xmldsig-more#kw-camellia192	2.6.3	EncryptionMethod
2001/04/xmldsig-more#kw-camellia256	2.6.3	EncryptionMethod
2001/04/xmldsig-more#md5	2.1.1	DigestAlgorithm
2001/04/xmldsig-more#PKCS7signedData	3.2	Retrieval type
2001/04/xmldsig-more#psec-kem	2.6.4	EncryptionMethod
2001/04/xmldsig-more#rawPGPKeyPacket	3.2	Retrieval type
2001/04/xmldsig-more#rawPKCS7signedData	3.2	Retrieval type
2001/04/xmldsig-more#rawSPKISexp	3.2	Retrieval type
2001/04/xmldsig-more#rawX509CRL	3.2	Retrieval type
2001/04/xmldsig-more#RetrievalMethod	3.2	Retrieval type
2001/04/xmldsig-more#rsa-md5	2.3.1	SignatureMethod
2001/04/xmldsig-more#rsa-sha256	2.3.2	SignatureMethod
2001/04/xmldsig-more#rsa-sha384	2.3.3	SignatureMethod
2001/04/xmldsig-more#rsa-sha512	2.3.4	SignatureMethod
2001/04/xmldsig-more#rsa-ripemd160	2.3.5	SignatureMethod
2001/04/xmldsig-more#sha224	2.1.2	DigestAlgorithm
2001/04/xmldsig-more#sha384	2.1.3	DigestAlgorithm
2001/04/xmldsig-more#xptr	2.5.1	Transform
2001/04/xmldsig-more#PKCS7signedData	3.1	KeyInfo child
2001/04/xmlenc#aes128-cbc	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#aes192-cbc	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#aes256-cbc	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#dh	[XMLENC11]	AgreementMethod
2001/04/xmlenc#kw-aes128	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#kw-aes192	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#kw-aes256	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#ripemd160	[XMLENC11]	DigestAlgorithm
2001/04/xmlenc#rsa-1_5	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#rsa-oaep-mgf1p	[XMLENC11]	EncryptionMethod
2001/04/xmlenc#sha256	[XMLENC11]	DigestAlgorithm
2001/04/xmlenc#sha512	[XMLENC11]	DigestAlgorithm
2001/04/xmlenc#tripledes-cbc	[XMLENC11]	EncryptionMethod
2002/06/xmldsig-filter2	[XPATH]	Transform
2002/07/decrypt#XML	[DECRYPT]	Transform
2002/07/decrypt#Binary	[DECRYPT]	Transform
2006/12/xmlc12n11#	[CANON11]	Canonicalization
2006/12/xmlc14n11#WithComments	[CANON11]	Canonicalization
2007/05/xmldsig-more#ecdsa-ripemd160	2.3.6	SignatureMethod
2007/05/xmldsig-more#ecdsa-whirlpool	2.3.5	SignatureMethod
2007/05/xmldsig-more#kw-seed128	2.6.6	EncryptionMethod
2007/05/xmldsig-more#md2-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#md5-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#MGF1	2.3.9	SignatureMethod
2007/05/xmldsig-more#ripemd128-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#ripemd160-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#rsa-pss	2.3.9	SignatureMethod
2007/05/xmldsig-more#rsa-sha224	2.3.11	SignatureMethod

2007/05/xmldsig-more#rsa-whirlpool	2.3.5	SignatureMethod
2007/05/xmldsig-more#seed128-cbc	2.6.5	EncryptionMethod
2007/05/xmldsig-more#sha1-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha224-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha256-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha3-224	2.1.5	DigestAlgorithm
2007/05/xmldsig-more#sha3-224-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha3-256	2.1.5	DigestAlgorithm
2007/05/xmldsig-more#sha3-256-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha3-384	2.1.5	DigestAlgorithm
2007/05/xmldsig-more#sha3-384-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha3-512	2.1.5	DigestAlgorithm
2007/05/xmldsig-more#sha3-512-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha384-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#sha512-rsa-MGF1	2.3.10	SignatureMethod
2007/05/xmldsig-more#whirlpool	2.1.4	DigestAlgorithm
2007/05/xmldsig-more#whirlpool-rsa-MGF1	2.3.10	SignatureMethod
2009/xmlenc11#kw-aes-128-pad	[XMLENC11]	EncryptionMethod
2009/xmlenc11#kw-aes-192-pad	[XMLENC11]	EncryptionMethod
2009/xmlenc11#kw-aes-256-pad	[XMLENC11]	EncryptionMethod
2009/xmldsig11#dsa-sha256	[XMLDSIG11]	SignatureMethod
2009/xmldsig11#ECKeyValue	[XMLDSIG11]	Retrieval type
2009/xmldsig11#DEREncodedKeyValue	[XMLDSIG11]	Retrieval type
2009/xmlenc11#aes128-gcm	[XMLENC11]	EncryptionMethod
2009/xmlenc11#aes192-gcm	[XMLENC11]	EncryptionMethod
2009/xmlenc11#aes256-gcm	[XMLENC11]	EncryptionMethod
2009/xmlenc11#ConcatKDF	[XMLENC11]	EncryptionMethod
2009/xmlenc11#mgf1sha1	[XMLENC11]	SignatureMethod
2009/xmlenc11#mgf1sha224	[XMLENC11]	SignatureMethod
2009/xmlenc11#mgf1sha256	[XMLENC11]	SignatureMethod
2009/xmlenc11#mgf1sha384	[XMLENC11]	SignatureMethod
2009/xmlenc11#mgf1sha512	[XMLENC11]	SignatureMethod
2009/xmlenc11#pbkdf2	[XMLENC11]	EncryptionMethod
2009/xmlenc11#rsa-oaep	[XMLENC11]	EncryptionMethod
2009/xmlenc11#ECDH-ES	[XMLENC11]	EncryptionMethod
2009/xmlenc11#dh-es	[XMLENC11]	EncryptionMethod
2010/xmlsec-ghc#generic-hybrid	[GENERIC]	Generic Hybrid
2010/xmlsec-ghc#rsaes-kem	[GENERIC]	Generic Hybrid
2010/xmlsec-ghc#ecies-kem	[GENERIC]	Generic Hybrid
TR/1999/REC-xpath-19991116	[XPATH]	Transform
TR/1999/REC-xslt-19991116	[XSLT]	Transform
TR/2001/06/xml-exc-c14n#	[XCANON]	Canonicalization
TR/2001/06/xml-exc-c14n#WithComments	[XCANON]	Canonicalization
TR/2001/REC-xml-c14n-20010315	[CANON10]	Canonicalization
TR/2001/REC-xml-c14n-20010315#WithComments	[CANON10]	Canonicalization
TR/2001/REC-xmlschema-1-20010502	[Schema]	Transform

La partie initiale "http://www.w3.org/" de l'URI n'est pas incluse ci-dessus.

5. Considérations d'allocation

Les considérations d'allocation par le W3C et l'IANA sont données ci-cessous.

5.1 Considérations d'allocation par le W3C

Comme il est facile à tous de construire ses propres URI uniques [RFC3986] et, si approprié, d'obtenir un URI du W3C, il n'est pas prévu que des URI "http://www.w3.org/2007/05/xmldsig-more#" supplémentaires soient créés au delà de ceux énumérés dans la présente RFC. (Les règles de stabilité de l'espace de noms du W3C interdisent la création de nouveaux URI sous "http://www.w3.org/2000/09/xmldsig#" et les URI sous "http://www.w3.org/2001/04/xmldsig-more#" ont été gelés avec la publication de la [RFC4051].)

Un URI "xmldsig-more" n'implique aucun statut officiel du W3C ou de l'IETF pour ces algorithmes ou identifiants ni n'implique qu'ils ne soient utiles que dans les signatures numériques. Actuellement, déréférencer de tels URI peut produire ou non un document fourre-tout temporaire. La permission d'utiliser ces préfixes d'URI est accordée par le W3C.

5.2 Considérations relatives à l'IANA

L'IANA a établi un registre intitulé "XML Security URIs". Son contenu initial correspond au paragraphe 4.2 du présent document avec chaque numéro de paragraphe dans la colonne "Sec/Doc" donnant une référence à la présente RFC (par exemple, "2.6.4" signifie "[RFC6931], paragraphe 2.6.4").

De nouvelles entrées, incluant de nouveaux types, seront ajoutés sur la base d'une révision par expert [RFC5226]. Les critères d'inclusion sont (1) une documentation suffisante pour l'interopérabilité de l'algorithme ou type de données et la syntaxe XML pour sa représentation et son utilisation et (2) une importance suffisante comme normalement indiqué par l'inclusion en (2a) d'une note approuvée par le W3C, de Recommandation proposée, ou de Recommandation ou (2b) un document approuvé par l'IETF sur la voie de la normalisation. Normalement, le registre fera référence à un document du W3C ou de l'IETF spécifiant une telle syntaxe XML ; ce document contiendra une description plus abstraite de l'algorithme ou type de données ou fera référence à un autre document avec une description plus abstraite.

6. Considérations pour la sécurité

La présente RFC s'occupe de documenter les URI qui désignent les algorithmes et certains types de données utilisés en connexion avec la sécurité de XML. Les considérations de sécurité varient largement avec les algorithmes particuliers, et les considérations générales de sécurité pour la sécurité de XML sortent du domaine d'application de ce document mais apparaissent dans [XMLDSIG11], [XMLENC11], [CANON10], [CANON11], et [GENERIC].

La [RFC6151] devrait être consultée avant de considérer l'utilisation de MD5 comme méthode de résumé ou RSA-MD5 comme méthode de signature.

Voir la [RFC6194] pour les considérations de sécurité pour SHA-1 et la [RFC6151] pour les considérations de sécurité pour MD5.

Des considérations de sécurité supplémentaires sont données en connexion avec la description de certains algorithmes dans le corps du présent document.

Les développeurs devraient être conscients que les algorithmes de chiffrement s'affaiblissent avec le temps. Lorsque de nouvelles techniques de cryptanalyse sont développées et que les performance de calcul s'améliorent, le facteur travail pour casser un algorithme cryptographique particulier se réduit. Donc, les mises en œuvre cryptographiques devraient être modulaires, permettant que de nouveaux algorithmes soient directement insérés. C'est-à-dire que les mises en œuvre devraient être prêtes à ce que l'ensemble des algorithmes de mise en œuvre obligatoire change avec le temps.

7. Remerciements

Nous remercions chaleureusement de leurs contributions au présent document les personnes suivantes, en ordre alphabétique : Benoit Claise, Adrian Farrel, Stephen Farrell, Ernst Giessmann, Frederick Hirsch, Bjoern Hoehrmann, Russ Housley, Satoru Kanno, Charlie Kaufman, Konrad Lanz, HwanJin Lee, Barry Leiba, Peter Lipp, Subramanian Moonesamy, Thomas Roessler, Hanseong Ryu, Peter Saint-Andre, et Sean Turner.

Les contributeurs suivants à la [RFC4051], sur laquelle se fonde le présent document, ont aussi droit à nos remerciements : Glenn Adams, Merlin Hughs, Gregor Karlinger, Brian LaMachia, Shiho Moriai, Joseph Reagle, Russ Housley et Joel Halpern..

Appendice A. Changements depuis la RFC 4051

Les changements suivants ont été faits à la RFC 4051 pour produire le présent document.

1. Mise à jour et ajout de nombreuses références de RFC, W3C, et projets Internet.

2. Ajout de #ecdsa-ripemd160, #whirlpool, #ecdsa-whirlpool, #rsa-whirlpool, #seed128-cbc, et #kw-seed128.

3. Incorporation de l'errata de la RFC 4051 [Errata191].

4. Ajout des sections d'index d'URI et de fragments.

5. Pour MD5 et SHA-1, ajout des références aux [RFC6151] et [RFC6194].

5. Ajout d'un paragraphe fourre-tout SHA-3 / Keccak incluant #sha3-224, #sha3-256, #sha3-384, et #sha3-512.

6. Ajout des paragraphes RSASSA-PSS incluant #sha3-224-MGF1, #sha3-256-MGF1, #sha3-384-MGF1, #sha3-512-MGF1, #md2-rsa-MGF1, #md5-rsa-MGF1, #sha1-rsa-MGF1, #sha224-rsa-MGF1, #sha256-rsa-MGF1, #sha384-rsa-MGF1, #sha512-rsa-MGF1, #ripemd128-rsa-MGF1, #ripemd160-rsa-MGF1, et #whirlpool-rsa-MGF1.

7. Ajout de nouveaux URI provenant de XML 1.1 canonique et du chiffrement XML 1.1 incluant : #aes128-gcm, #aes192-gcm, #aes256-gc, #ConcatKDF, #pbkdf, #rsa-oaep, #ECDH-ES, et #dh-es.

8. Ajout du paragraphe d'acronymes.

9. Ajout de nombreux URI spécifiés dans les documents de sécurité XML du W3C dans les index. Ils n'ont pas de paragraphes et le corps du présent document -- par exemple, ceux pour dsa-sha256, mgf1sha*, decrypt#XML, et xmldsig-filter2.

10. Demande d'établissement d'un registre IANA.

11. Divers changements rédactionnelss.

Références normatives

[10118-3] ISO/CEI 10118-3:2004, "Technologies de l'information – techniques de sécurité -- fonctions de hachage -- partie 3 : fonctions de hachage dédiées", 2004.

[18033-2] ISO/CEI 18033-2:2010, "Technologies de l'information – techniques de sécurité – algorithme de chiffrements -- partie 3 : chiffrements asymétriques", 2010.

[Camellia] Aoki, K., Ichikawa, T., Matsui, M., Moriai, S., Nakajima, J., et T. Tokita, "Camellia: A 128-bit Block Cipher Suitable for Multiple Platforms - Design et Analysis", dans Selected Areas in Cryptography, 7th Annual International Workshop, SAC 2000, août 2000, Proceedings, notes de lecture dans Computer Science 2012, pp. 39-56, Springer-Verlag, 2001.

[FIPS180-4] US National Institute of Science and Technology, "Secure Hash Standard (SHS)", FIPS 180-4, mars 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf >.

[FIPS186-3 US National Institute of Science and Technology, "Digital Signature Standard (DSS)", FIPS 186-3, juin 2009, < http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf >.

[IEEEP1363a] IEEE 1363a-2004, "Standard Specifications for Public Key Cryptography- Amendment 1: Additional Techniques", 2004.

[RC4] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C", Second Edition, John Wiley and Sons, New York, NY, 1996.

[RFC1321] R. Rivest, "Algorithme de résumé de message MD5", avril 1992. *(Information)*

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par* 2184, 2231, 5335.)

[RFC2104] H. Krawczyk, M. Bellare et R. Canetti, "HMAC : Hachage de clés pour l'authentification de message", février 1997.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

(MàJ par RFC8174)

[RFC2315] B. Kaliski, "PKCS n° 7 : Syntaxe de message cryptographique, version 1.5", mars 1998. *(Information)*

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

[RFC3394] J. Schaad, R. Housley, "Algorithme d'enveloppe de clés pour la norme de chiffrement évoluée (AES)", septembre 2002. *(Information)*

[RFC3447] J. Jonsson et B. Kaliski, "Normes de cryptographie à clés publiques (PKCS) n° 1 : Spécifications de la cryptographie RSA version 2.1", février 2003. *(Obsolète, remplacée par* RFC8017) *(Information)*

[RFC3713] M. Matsui, J. Nakajima, S. Moriai, "Description de l'algorithme de chiffrement Camellia", avril 2004. *(Information)*

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005.

[RFC4050] S. Blake-Wilson et autres, "Utilisation de l'algorithme de signature à courbe elliptique (ECDSA) pour les signatures numériques XML", avril 2005. *(Information)*

[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.

[RFC4269] H.J. Lee et autres, "Algorithme de chiffrement SEED", décembre 2005. *(Remplace* RFC4009) *(Information)*

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[RFC6234] D. Eastlake 3rd, T. Hansen, "Algorithmes US de hachage sécurisé (SHA et HMAC fondé sur SHA et HKDF)", mai 2011. *(Remplace la* RFC4634) *(MàJ la* RFC3174) *(Information)*

[X9.62] American National Standards Institute, Accredited Standards Committee X9, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62:2005.

[XMLENC10] Reagle, J. et D. Eastlake, "XML Encryption Syntax et Processing", W3C Recommendation, 10 décembre 2002, < http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/ >.

[XMLENC11] Eastlake, D., Reagle, J., Hirsch, F., and T. Roessler, "XML Encryption Syntax and Processing Version 1.1", W3C Proposed Recommendation, 24 janvier 2013, < http://www.w3.org/TR/2013/PR-xmlenc-core1-20130124/ >.

[XPointer] Grosso, P., Maler, E., Marsh, J., and N. Walsh, "XPointer Framework", W3C Recommendation, 25 mars 2003, < http://www.w3.org/TR/2003/REC-xptr-framework-20030325/ >.

Références pour information

[CANON10] Boyer, J., "Canonical XML Version 1.0", W3C Recommendation, 15 mars 2001, <http://www.w3.org/TR/2001/REC-xml-c14n-20010315 >.

[CANON11] Boyer, J., and G. Marcy, "Canonical XML Version 1.1", W3C Recommendation, 2 mai 2008, <http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/ >.

[DECRYPT] Hughes, M., Imamura, T., and H. Maruyama, "Decryption Transform for XML Signature", W3C Recommendation, 10 décembre 2002, < http://www.w3.org/TR/2002/REC-xmlenc-decrypt-20021210 >.

[Errata191] RFC Errata, Errata ID 191, RFC 4051, < http://www.rfc-editor.org >.

[GENERIC] Nystrom, M. and F. Hirsch, "XML Security Generic Hybrid Ciphers", W3C Working Group Note, 24 janvier 2013, < http://www.w3.org/TR/2013/NOTE-xmlsec-generic-hybrid-20130124/ >.

[Keccak] Bertoni, G., Daeman, J., Peeters, M., and G. Van Assche, "The KECCAK sponge function family", janvier 2013, < http://keccak.noekeon.org >.

[RFC3075] D. Eastlake 3rd et autres, "Syntaxe et traitement de signature XML", mars 2001. (*Obsolète, voir* RFC3275) *(P.S.)*

[RFC3076] J. Boyer, "XML canonique, version 1.0", mars 2001.

[RFC3092] D. Eastlake 3rd, C. Manros et E. Raymond, "Étymologie de "Foo"", RFC 3092, 1er avril 2001.

[RFC3741] J. Boyer et autres, "Canonisation XML exclusive, version 1.0", mars 2004. *(Information)*

[RFC4010] J. Park et autres, "Utilisation de l'algorithme de chiffrement SEED dans la syntaxe de message cryptographique (CMS)", février 2005. *(P.S.)*

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

[RFC6090] D. McGrew, K. Igoe, M. Salter, "Algorithmes fondamentaux de cryptographie par courbe élliptique", février 2011. *(Info.)*

[RFC6151] S. Turner, L. Chen, "Mise à jour des considérations de sécurité pour les algorithmes de résumé de message MD5 et le HMAC-MD5", mars 2011. (*MàJ* RFC1321, RFC2104) *(Information)*

[RFC6194] T. Polk et autres, "Considérations sur la sécurité pour les algorithmes de résumé de message SHA-0 et SHA-1", mars 2011. *(Information)*

[Schema] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", W3C Recommendation, 28 octobre 2004, < http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ >.

Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, 28 octobre 2004, < http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ >.

[SHA-3] US National Institute of Science and Technology, "SHA-3 WINNER", février 2013, <http://csrc.nist.gov/groups/ST/hash/sha-3/winner_sha-3.html >.

[W3C] World Wide Web Consortium, < http://www.w3.org >.

[XCANON] Boyer, J., Eastlake, D., and J. Reagle, "Exclusive XML Canonicalization Version 1.0", W3C Recommendation, 18 juillet 2002, < http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/ >.

[XMLDSIG10] Eastlake, D., Reagle, J., Solo, D., Hirsch, F., and T. Roessler, "XML Signature Syntax et Processing (Second Edition)", W3C Recommendation, 10 juin 2008, < http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/ >.

[XMLDSIG11] Eastlake, D., Reagle, J., Solo, D., Hirsch, F., Nystrom, M., Roessler, T., and K. Yiu, "XML Signature Syntax and Processing Version 1.1", W3C Proposed Recommendation, 24 janvier 2013, <http://www.w3.org/TR/2013/PR-xmldsig-core1-20130124/ >.

[XMLDSIG-PROP] Hirsch, F., "XML Signature Properties", W3C Proposed Recommendation, 24 janvier 2013, <http://www.w3.org/TR/2013/PR-xmldsig-properties-20130124/ >.

[XMLSECXREF] Hirsch, F., Roessler, T., and K. Yiu, "XML Security Algorithm Cross-Reference", W3C Working Group Note, 24 janvier 2013, < http://www.w3.org/TR/2013/NOTE-xmlsec-algorithms-20130124/ >.

[XPATH] Boyer, J., Hughes, M., and J. Reagle, "XML-Signature XPath Filter 2.0", W3C Recommendation, 8 novembre 2002, < http://www.w3.org/TR/2002/REC-xmldsig-filter2-20021108/ >.

Berglund, A., Boag, S., Chamberlin, D., Fernandez, M., Kay, M., Robie, J., and J. Simeon, "XML Path Language (XPath) 2.0 (Second Edition)", W3C Recommendation, 14 décembre 2010, <http://www.w3.org/TR/2010/REC-xpath20-20101214/ >.

[XSLT] Saxonica, M., "XSL Transformations (XSLT) Version 2.0", W3C Recommendation, 23 janvier 2007, <http://www.w3.org/TR/2007/REC-xslt20-20070123/ >.

Adresse de l'auteur

Donald E. Eastlake, 3rd
Huawei Technologies
155 Beaver Street
Milford, MA 01757
USA
téléphone : 508-333-2270
mél : d3e3e3@gmail.com