# rfc5914

Format d'ancre de confiance

Ce document décris une structure pour représenter les informations d'ancre de confiance. Une ancre de confiance est une entité autoritative représentée par une clé publique et des données associées. La clé publique est utilisée pour vérifier les signatures numériques, et les données associées sont utilisée pour contraindre les types d'information ou actions pour lesquelles l'ancre de confiance a autorité. Les structures définies dans ce document sont prévus pour satisfaire les exigences liées au format définis dans les exigencse de gestion d'ancre de confiance.

Les ancres de confiance sons largement utilisées pour vérifier les signatures numériques et valider les chemins de certification. Ils sont requis pour valider les chemins de certification. Bien que largement utilisés, il n'y a pas de format standard pour représenter les informations d'ancre de confiance. ce document décris la structure TrustAnchorInfo. Cette structure est prévue pour satisfaire les exigences liées au format exprimées dans les exigences de gestion d'ancre de confiance ( rfc6024 ) et est exprimée en utilisant ASN.1. Il peut fournir une alternative plus compact aux certificats X.509 pour échanger les informations d'ancre de confiance et fournir un moyen d'associer des contraintes additionnelles ou alternatives avec les certificats sans casser la signature dans le certificat.

# Syntaxe d'information d'ancre de confiance

Cette section décris la structure TrustAnchorInfo :

```
TrustAnchorInfo ::= SEQUENCE {
 version TrustAnchorInfoVersion DEFAULT v1,
 pubKey SubjectPublicKeyInfo,
 keyId KeyIdentifier,
 taTitle TrustAnchorTitle OPTIONAL,
 certPath CertPathControls OPTIONAL,
 exts [1] EXPLICIT Extensions OPTIONAL,
 taTitleLangTag [2] UTF8String OPTIONAL }

TrustAnchorInfoVersion ::= INTEGER { v1(1) }
```

# Version

La version identifie la version de TrustAnchorInfo. De futures mises à jours de ce document peuvent inclure des changements dans cette structure, auquel cas le numéro de version devrait être incrémenté. Cependant, la valeur par défaut, v1, ne peut pas être changée.

# Clé publique

pubKey identifie la clé publique et l'algorithme associé avec l'ancre de confiance en utilisant la structure SubjectPublicKeyInfo (rfc5280). La structure SubjectPublicKeyInfo contient l'identifiant d'algorithme suivi par la clé publique elle-même. Le champ algorithm est un AlgorithmIdentifier, qui contient un identifiant d'objet et des paramètres optionnels. L'identifiant d'objet nomme l'algorithme de clé publique et indique la syntaxe des paramètres, si présent, aussi bien que le format de la clé publique. La clé publique est encodée en chaîne de bits.

# Titre de l'ancre de confiance

```
TrustAnchorTitle ::= UTF8String (SIZE (1..64))
```

taTitle est optionnel. Si présent, il fournis un nom compréhensible pour l'ancre de confiance. Le texte est encodé en UTF-8. Le champ taTitleLangTag identifie la langue utilisée pour exprimer le taTitle. Quand taTitleLangTag est absent, l'englais ("en") est utilisé. La valeur de taTitleLangTag devrait être un tag de langue comme décris dans la rfc5646.

# Contrôles de chemin de certification

```
CertPathControls ::= SEQUENCE {
 taName Name,
 certificate [0] Certificate OPTIONAL,
 policySet [1] CertificatePolicies OPTIONAL,
 policyFlags [2] CertPolicyFlags OPTIONAL,
 nameConstr [3] NameConstraints OPTIONAL,
 pathLenConstraint[4] INTEGER (0..MAX) OPTIONAL}
```

certPath est optionnel. Si présent, il fournis les contrôles nécessaires pour initialiser une implémentation d'algorithme de validation de certification X.509. Si absent, l'ancre de confiance ne peut pas être utilisée pour valider la signature dans un certificat X.509.

taName fournis le nom distinct X.500 associé avec l'ancre de confiance, et ce nom distinct est utilisé pour construire et valider un chemin de certification X.509. Le nom ne doit pas être une séquence vide.

certificate fournis un certificat X.509 optionnel, qui peut être utilisé dans certains environnements pour représenter l'ancre de confiance dans le développement et la validation du chemin de certification. Si le certificat est présent, le nom du sujet dans le certificat doit correspondre exactement le nom distinct X.500 fournis dans le champ taName, la clé publique doit correspondre exactement à la clé publique dans le champ pubKey, et l'extension subjectKeyIdentifier, si présent, doit correspondre exactement l'identifiant de clé dans le champ keyId. La description complète de la syntaxe et les sémantiques du certificat sont fournis dans la rfc5280. Les contraintes définies dans les champs policySet, policyFlags, nameConstr, pathLenConstraint, et exts dans TrustAnchorInfo remplacent les valeurs contenues dans un certificat ou fournissent des valeurs pour les extensions non présente dans le certificat. Les valeurs définies dans ces champs TrustAnchorInfo sont toujours forcés. Les extensions inclues dans un certificat sont forcés seulement s'il n'y a pas de valeur correspondante dans le TrustAnchorInfo. La correspondance entre les extensions dans le certificat et les champs TrustAnchorInfo sont définis comme suit :

- une extension de certificat id-ce-certificatePolicies correspond au champ CertPathControls.policySet
- une extension de certificat id-ce-policyConstraints correspond au champ CertPolicyFlags.inhibitPolicyMapping et CertPolicyFlags.requireExplicitPolicy
- Une extension de certificat id-ce-inhibitAnyPolicy correspond au champ CertPolicyFlags.inhibitAnyPolicy
- Une extension de certificat id-ce-nameConstraints correspond au champ CertPathControls.nameConstr
- Le champ pathLenConstraint d'une extension de certificat id-ce-basicConstraints correspond au champ CertPathControls.pathLenConstraint ( La présence d'une structure CertPathControls correspond à une valeur TRUE dans le champ cA de l'extension BasicConstraints )
- Toute autre extension de certificat correspond au même type d'extension dans le champ TrustAnchorInfo.exts

```
CertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
 policyIdentifier CertPolicyId,
 policyQualifiers SEQUENCE SIZE (1..MAX) OF PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER
```

policySet est optionnel. Si présent, il contient une séquence d'identifiant de stratégie de certificat fournis en entrée de l'algorithme de validation de chemin de certification. Si absent, la valeur spéciale any-policy est fournie comme entrée de l'algorithme de validation de chemin de certification. La description complète de la syntaxe et des sémantiques de CertificatePolicies sont fournis dans la rfc5280, incluant la syntaxe PolicyInformation. Dans ce contexte, la structure optionnelle policyQualifiers ne doit pas être inclue.

```
CertPolicyFlags ::= BIT STRING {
 inhibitPolicyMapping (0),
 requireExplicitPolicy (1),
 inhibitAnyPolicy (2) }
```

policyFlags est optionnel. Si présent, 3 valeurs booléennes pour l'entrée dans l'algorithme de validation de chemin de certification sont fournis dans un BIT STRING. Si absent, l'entrée de l'algorithme de validation de chemin de certification est { FALSE, FALSE, FALSE }, qui représente le paramètre le plus libéral de ces flags. Ces 3 bits sont utilisés comme suit :

- inhibitPolicyMapping indique si le mappage de stratégie est autorisé dans le chemin de certification. Quand mis à TRUE, le mappage de stratégie n'est pas permis. Cette valeur représente la valeur d'entrée initial-policy-mapping-inhibit dans l'algorithme de validation de chemin de certification décris dans la rfc5280.
- requireExplicitPolicy indique si le chemin de certification doit être valide pour au moins une des stratégies de certificat dans le policySet. À TRUE, tous les certificats dans le chemin de certification doivent contenir un identifiant de stratégie acceptable dans l'extension de stratégie de certificat. Cette valeur représente la valeur d'entrée initial-explicit-policy dans l'algorithme de validation de chemin de certification décris dans la rfc5280. Un identifiant de stratégie acceptable est un membre du policySet ou l'identifiant d'une stratégie qui est déclarée équivalente via le mappage de stratégie. Ce bit doit être à FALSE si policySet est absent.
- inhibitAnyPolicy indique si l'identifiant de stratégie anyPolicy, avec la valeur { 2 5 29 32 0 }, est considéré un match explicite pour d'autres stratégies de certificat. Cette valeur représente la valeur d'entrée initial-any-policy-inhibit dans l'algorithme de validation de chemin de certification décris dans la rfc5280.

```
NameConstraints ::= SEQUENCE {
 permittedSubtrees [0] GeneralSubtrees OPTIONAL,
 excludedSubtrees [1] GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
 base GeneralName,
 minimum [0] BaseDistance DEFAULT 0,
 maximum [1] BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

nameConstr est optionnel. Il a la même syntaxe et sématiques que l'extension de certificat de contrainte de noms, qui inclue une liste de noms permis et une liste de noms exclus. La définition de GeneralName peut être trouvée dans la rfc5280. Quand il est présent, les contraintes sont fournies sur les noms (incluant les noms alternatifs) qui doivent apparaître dans le certificats X.509 sous-jacents dans un chemin de certification. Ce champ est utilisé pour définir les valeurs d'entrée initial-permitted-subtrees et d'initial-excluded-subtrees dans l'algorithme de validation de chemin de certification décris dans la rfc5280. Quand ce champ est absent, la variable initial-permitted-subtrees n'est pas limitée et la variable initial-excluded-subtrees est vide.

Le champ pathLenConstraint donne le nombre maximum de certificats intermédiaire non-auto-émis qui peuvent suivre ce certificat dans un chemin de certification valide. ( Note : le dernier certificat dans le chemin de certification n'est pas un certificat intermédiaire et n'est pas inclus dans cette limite. Généralement, le dernier certificat est en certificat EE, mais il peut être un certificat CA). Un pathLenConstraint de 0 indique qu'aucun certificat d'autorité de certification intermédiaire non-auto-émis ne peut suivre dans le chemin de certification. Quand il apparaît, le champ pathLenConstraint doit être supérieur ou égal àl 0. Quand il n'apparaît pas, aucune limite n'est imposée.

Quand l'ancre de confiance est utilisée pour valider un chemin de certification, CertPathControls fournis les limitations dans le chemins de certification qui seront validés avec succès. Une application qui valide un chemin de certification ne devrait pas ignorer ces limitations, mais l'application peut imposer des limitations additionnelles pour s'assurer que le chemin de certification validé est approprié pour le contexte applicatif prévu. Comme entrée de l'algorithme de validation de chemin de certification, une application peut :

- Fournir un sous-jeu de stratégies de certification fournies dans le policySet
- Fournir une valeur TRUE, si approprié, pour les flags dans le policyFlags
- Fournir un sous-jeu de noms permis fournis dans nameConstr
- Fournir des noms exclus additionnels à ceux fournis dans le nameConstr
- Fournir une valeur plus petite pour pathLenConstraint

# Extensions

exts est optionnel. Si présent, il peut être utilisé pour associer des informations additionnelles avec l'ancre de confiance en utilisant la structure Extensions standard. Les extensions qui sont prévus pour être largement utilisé ont été inclus dans la structure CertPathControls pour éviter une surchage associée avec l'utilisation de la structure Extensions. Pour éviter la duplication avec le champ CertPathControls, les types d'extensions suivants ne doivent pas apparaître dans le champ exts et sont ignorés s'il n'apparaissent : id-ce-certificatePolicies, id-ce-policyConstraints, id-ce-inhibitAnyPolicy, et id-ce-nameConstraints.

# Liste d'ancre de confiance

TrustAnchorInfo permet la représentation d'une simple ancre de confiance. Dans de nombreux cas, il est préférable de représenter une collection d'ancre de confiance. La structure TrustAnchorList est définie dans ce but. TrustAnchorList est définis comme une séquence d'un ou plusieurs objects TrustAnchorChoice. TrustAnchorChoice fournis 3 options pour représenter une ancre de confiance. L'option certificat permet d'utiliser un certificat sans contraintes additionnelles. L'option tbsCert permet d'associer des contraintes en supprimant un signature dans un certificat et en changeant le champ extensions. L'option taInfo permet d'utiliser une structure TrustAnchorInfo définis dans ce document.

```
TrustAnchorList ::= SEQUENCE SIZE (1..MAX) OF TrustAnchorChoice

TrustAnchorChoice ::= CHOICE {
 certificate Certificate,
 tbsCert [1] EXPLICIT TBSCertificate,
 taInfo [2] EXPLICIT TrustAnchorInfo }

trust-anchor-list PKCS7-CONTENT-TYPE ::= { TrustAnchorList IDENTIFIED BY id-ct-trustAnchorList }
```

La structure TrustAnchorList peut être protégée en utilisant la structure SignedData définis dans CMS. L'identifiant d'objet id-ct-trustAnchorList a été définis pour représenter le payloads TrustAnchorList avec les structure CMS.

# Considérations de sécurité

La compromission d'une clé privée d'ancre de confiance permet à des tiers non-autorisés d'usurper l'ancre de confiance, avec des conséquences potentiellement sévères. Quand des contraintes basés sur TA sont forcés, une personne non-autorisée ayant la clé privée sera limité par les contrôles de chemin de certification associés avec l'ancre de confiance, comme exprimé dans les champs certPath et exts. Par exemple, les contraintes de nom dans l'ancre de confiance vont déterminer l'espace de nom qui sera accepté dans le certificats qui sont validés en utilisant l'ancre de confiance compromise. Le recours à une clé publique d'un ancre de confiance inapproprié ou incorrect a des conséquences potentiellement sévères.

La compromission d'une clé privée de CA a le même type de problème que la compromission de clé privée d'une ancre de confiance. Une entité non autorisée possédant la clé privée de la CA sera limité par les contrôles de chemin de certification associés avec l'ancre de confiance, comme exprimé dans le champ certPath ou comme extension.

L'utilisation d'un certificat indépendant de la structure TrustAnchorinfo qui l'enveloppe doit être géré avec prudence pour éviter de violer les contraintes exprimées dans le TrustAnchorInfo. En enveloppant un certificat dans une structure TrustAnchorinfo, les valeurs inclues dans le certificat devraient être évalués pour s'assurer qu'il n'y a pas de confusion ou de conflit avec les valeurs dans la structure TrustAnchorinfo.