ITUPublications
Recommandations

Union internationale des télécommunications
Secteur de la normalisation

Recommandation

UIT-T Y.4500.3 (01/2023)

SÉRIE Y: Infrastructure mondiale de l'information, protocole Internet, réseaux de prochaine génération, Internet des objets et villes intelligentes

Internet des objets et villes et communautés intelligentes – Cadres, architectures et protocoles

oneM2M – Solutions de sécurité

RECOMMANDATIONS UIT-T DE LA SÉRIE Y

Infrastructure mondiale de l'information, protocole Internet, réseaux de prochaine génération, Internet des objets et villes intelligentes

INFRASTRUCTURE MONDIALE DE L'INFORMATION	Y.100-Y.999
ASPECTS RELATIFS AU PROTOCOLE INTERNET	Y.1000-Y.1999
RÉSEAUX DE PROCHAINE GÉNÉRATION	Y.2000-Y.2999
RÉSEAUX FUTURS	Y.3000-Y.3499
INFORMATIQUE EN NUAGE	Y.3500-Y.3599
BIG DATA	Y.3600-Y.3799
RÉSEAUX DE DISTRIBUTION DE CLÉS QUANTIQUES	Y.3800-Y.3999
INTERNET DES OBJETS ET VILLES ET COMMUNAUTÉS INTELLIGENTES	Y.4000-Y.4999
Considérations générales	Y.4000-Y.4049
Termes et définitions	Y.4050-Y.4099
Exigences et cas d'utilisation	Y.4100-Y.4249
Infrastructure, connectivité et réseaux	Y.4250-Y.4399
Cadres, architectures et protocoles	**Y.4400-Y.4549**
Services, applications, calcul et traitement des données	Y.4550-Y.4699
Gestion, commande et qualité de fonctionnement	Y.4700-Y.4799
Identification et sécurité	Y.4800-Y.4899
Evaluation et analyse	Y.4900-Y.4999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

Recommandation UIT-T Y.4500.3

oneM2M – Solutions de sécurité

Résumé

La Recommandation UIT-T Y.4500.3 établit des spécifications relatives à la sécurité et à la protection de la confidentialité des communications de machine à machine (M2M).

Historique*

Édition	Recommandation	Approbation	Commission d'études	ID unique
1.0	UIT-T Y.4500.3	30-01-2023	20	11.1002/1000/15076

Mots clés

Authentification, autorisation, chiffrement, intégrité, oneM2M, confidentialité, sécurité.

* Pour accéder à la Recommandation, reporter cet URL https://handle.itu.int/ dans votre navigateur Web, suivi de l'identifiant unique.

AVANT-PROPOS

L'Union internationale des télécommunications (UIT) est une institution spécialisée des Nations Unies dans le domaine des télécommunications et des technologies de l'information et de la communication (ICT). Le Secteur de la normalisation des télécommunications (UIT-T) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets ou par des droits d'auteur afférents à des logiciels, et dont l'acquisition pourrait être requise pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux développeurs de consulter les bases de données appropriées de l'UIT-T disponibles sur le site web de l'UIT-T à l'adresse http://www.itu.int/ITU-T/ipr/.

TABLE DES MATIÈRES

Page

Recommandation UIT-T Y.4500.3

oneM2M – Solutions de sécurité

1 Domaine d'application

La présente Recommandation définit des solutions de sécurité applicables au sein du système de machine à machine (M2M).

Elle contient la version 2 de la spécification oneM2M – Solutions de sécurité oneM2M V2.4.1; elle est équivalente aux normes des partenaires de oneM2M, à savoir: ARIB, ATIS [b-ATIS.oneM2M.TS0003], CCSA, ETSI [b-ETSI TS 118 103], TIA, TSDSI [b-TSDSI STD T1.oneM2M TS-0003-2.4.1 V1.0.0], TTA [b-TTAT.MM-TS.0003 v2.4.1] et TTC [b-TTC TS-M2M-0003v2.4.1].

Note rédactionnelle – La structure de la présente Recommandation, notamment des paragraphes 3, 4 et 5, s'écarte légèrement de la structure habituelle des Recommandations UIT-T afin de conserver l'alignement avec les normes ou standards équivalents cités dans les paragraphes concernés. En outre, plusieurs annexes, appendices et paragraphes ont été laissés en blanc de façon intentionnelle, pour la même raison.

2 Références

Les Recommandations UIT-T et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions de la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes les Recommandations et autres références étant sujettes à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des Recommandations et autres références énumérées ci-dessous. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

[UIT-T X.509]	Recommandation UIT-T X.509 (2019), *Technologies de l'information – Interconnexion des systèmes ouverts– L'annuaire: cadre général des certificats de clé publique et d'attribut*.
[UIT-T Y.4500.1]	Recommandation UIT-T Y.4500.1 (2018), *oneM2M – Architecture fonctionnelle*.
[UIT-T Y.4500.4]	Recommandation UIT-T Y.4500.4 (2018), *oneM2M – Spécification du protocole central de couche service*.
[UIT-T Y.4500.11]	Recommandation UIT-T Y.4500.11 (2018), *oneM2M – Terminologie commune*.
[UIT-T Y.4500.22]	Recommandation UIT-T Y.4500.22 (2018), *oneM2M – Configuration des dispositifs de terrain*.
[UIT-T Y.4500.32]	Recommandation UIT-T Y.4500.32 (2018), *oneM2M – Spécification d'interface pour les fonctions MEF et MAF*.
[ARIB STD-T64-C.S0078-0]	ARIB STD-T64-C.S0078-0 v1.0 (2006), *Secured packet structure for CDMA Card Application Toolkit (CCAT) applications*.
[ARIB STD-T64 C.S0079-0]	ARIB STD-T64- C.S0079-0 v1.0 (2006), *Remote APDU Structure for CDMA Card Application Toolkit (CCAT) applications*.
[ETSI TS 101 220]	ETSI TS 101 220 V16.0.0 (2021), *Smart Cards; ETSI numbering system for telecommunication application providers (Release 16)*.

[ETSI TS 102 221] ETSI TS 102 221 V16.2.0 (2020), *Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 16)*.

[ETSI TS 102 225] ETSI TS 102 225 V11.0.0 (2012), *Smart Cards; Secured packet structure for UICC based applications (Release 11)*.

[ETSI TS 102 226] ETSI TS 102 226 V11.0.0 (2012), *Smart Cards; Remote APDU structure for UICC based applications (Release 11)*.

[ETSI TS 102 484] ETSI TS 102 484 V11.2.0 (2019), *Smart Cards; Secure channel between a UICC and an end-point terminal (Release 11)*.

[ETSI TS 102 671] ETSI TS 102 671 V11.0.0 (2018), *Smart Cards; Machine to Machine UICC; Physical and logical characteristics (Release 11)*.

[ETSI TS 131 115] ETSI TS 131 115 V10.1.1 (2013), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Secured packet structure for (U)SIM Toolkit applications (3GPP TS 31.115 version 10.1.1 Release 10)*.

[ETSI TS 131 116] ETSI TS 131 116 V10.3.0 (2012), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Remote APDU Structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications (3GPP TS 31.116 version 10.3.0 Release 10)*.

[ETSI TS 133 220] ETSI TS 133 220 V12.3.0 (2014), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) (3GPP TS 33.220 version 12.3.0 Release 12)*.

[IETF RFC 2104] IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*.

[IETF RFC 3548] IETF RFC 3548 (2003), *The Base16, Base32, and Base64 Data Encodings*.

[IETF RFC 3629] IETF RFC 3629 (2003), *UTF-8, a transformation format of ISO 10646*.

[IETF RFC 4279] IETF RFC 4279 (2005), *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*.

[IETF RFC 4492] IETF RFC 4492 (2006), *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*.

[IETF RFC 5246] IETF RFC 5246 (2008), *The Transport Layer Security (TLS) Protocol Version 1.2*.

[IETF RFC 5280] IETF RFC 5280 (2008), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

[IETF RFC 5289] IETF RFC 5289 (2009), *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*.

[IETF RFC 5480] IETF RFC 5480 (2009), *Elliptic Curve Cryptography Subject Public Key Information*.

[IETF RFC 5487] IETF RFC 5487 (2009), *Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode*.

[IETF RFC 5705] IETF RFC 5705 (2010), *Keying Material Exporters for Transport Layer Security (TLS)*.

[IETF RFC 5869] IETF RFC 5869 (2010), *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)*.

[IETF RFC 6066] IETF RFC 6066 (2011), *Transport Layer Security (TLS) Extensions: Extension Definitions*.

[IETF RFC 6347] IETF RFC 6347 (2012), *Datagram Transport Layer Security Version 1.2*.

[IETF RFC 6655] IETF RFC 6655 (2012), *AES-CCM Cipher Suites for Transport Layer Security (TLS)*.

[IETF RFC 6920] IETF RFC 6920 (2013), *Naming Things with Hashes*.

[IETF RFC 6960] IETF RFC 6960 (2013), *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*.

[IETF RFC 6961] IETF RFC 6961 (2013), *The Transport Layer Security (TLS) Multiple Certificate Status Request Extension*.

[IETF RFC 7030] IETF RFC 7030 (2013), *Enrollment over Secure Transport*.

[IETF RFC 7250] IETF RFC 7250 (2014), *Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*.

[IETF RFC 7251] IETF RFC 7251 (2014), *AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*.

[IETF RFC 7515] IETF RFC 7515 (2015), *JSON Web Signature (JWS)*.

[IETF RFC 7516] IETF RFC 7516 (2015), *JSON Web Encryption (JWE)*.

[IETF RFC 7518] IETF RFC 7518 (2015), *JSON Web Algorithms (JWA)*.

[IETF RFC 7519] IETF RFC 7519 (2015), *JSON Web Token (JWT)*.

[IETF RFC 8824] IETF RFC 8824 (2020), *Simple Certificate Enrolment Protocol*.

[ISO/CEI 7816-4] ISO/CEI 7816-4:2020, *Cartes d'identification – Cartes à circuit intégré – Partie 4: Organisation, sécurité et commandes pour les échanges*.

[NIST EC] National Institute of Standards and Technology (Juillet 1999), *Recommended Elliptic Curves for Federal Government user*. http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[OIDF CC] OpenID Foundation (2014), *OpenID Connect Core 1.0 incorporating errata set 1*.

[TIA-1098-A] TIA-1098-A (2008), *Generic Bootstrapping Architecture (GBA) Framework, (3GPP2 S.S0109-A)*.

[Unicode] Unicode 5.1.0 (2008), *Annexe 15 au standard Unicode; Unicode Normalization Forms*. http://www.unicode.org

[W3C XMLSIG] Recommandation du W3C (2013), *XML Signature Syntax and Processing v1.1*. http://www.w3.org/TR/xmlsig-core1/

[W3C XMLENC] Recommandation du W3C (2013), *XML Encryption Syntax and Processing v1.1*. http://www.w3.org/TR/xmlenc-core1/

3 Définitions

3.1 Termes définis ailleurs

Aux fins de la présente Recommandation, les termes et définitions figurant dans la Recommandation [UIT Y.4500.11] s'appliquent.

3.1.1 données authentifiées supplémentaires [TIA-1098-A]: données qui sont authentifiées, mais non chiffrées par un chiffrement authentifié avec un algorithme de données associé.

3.1.2 chiffrement authentifié avec données associées [TIA-1098-A]: algorithme assurant la confidentialité du texte en clair et permettant d'en vérifier l'intégrité et l'authenticité, tout en offrant la possibilité de vérifier l'intégrité et l'authenticité de certaines données authentifiées supplémentaires. On entend ici par "texte en clair" des données qui sont authentifiées et chiffrées.

3.1.3 fonction de serveur d'amorçage (BSF, *bootstrap server function*) [ETSI TS 133 220]: la fonction BSF est hébergée dans un élément du réseau contrôlé par un opérateur de réseau mobile. La fonction BSF, le système d'abonnement de rattachement (HSS, *home subscriber system*) et les équipements d'utilisateurs (UE, *user equipment*) font partie d'une architecture d'amorçage générique (GBA, *generic bootstrapping architecture*) dans laquelle l'exécution de la procédure d'amorçage établit un secret partagé entre le réseau et un UE.

NOTE – Le secret partagé peut être utilisé par exemple entre les NAF et les UE, à des fins d'authentification.

3.1.4 identificateur de transaction d'amorçage (B-TID, *bootstrapping transaction identifier*) [ETSI TS 133 220]: l'identificateur de transaction d'amorçage (B-TID) sert à lier l'identité de l'abonné aux données de clé au niveau des points de référence Ua, Ub et Zn de l'architecture GBA.

3.1.5 certificat de CA [b-Menezes]: certificat créé par une autorité de certification (CA), certifiant la clé publique d'une autre CA.

3.1.6 signature numérique [b-UIT-T X.510]: résultat de la transformation cryptographique de données qui, lorsqu'il est mis en œuvre correctement, fournit un mécanisme d'authentification de l'origine, d'intégrité des données et de non-répudiation du signataire.

NOTE – Le résumé est produit par une fonction de hachage non réversible et le chiffrement est effectué au moyen de la clé privée du signataire.

3.1.7 information d'identification personnelle [ETSI TS 133 220]: toute information relative à un individu conservée par un organisme, notamment toute information pouvant être utilisée pour connaître ou retrouver l'identité d'un individu, par exemple son nom, son numéro de sécurité sociale, la date et le lieu de sa naissance, le nom de jeune fille de sa mère, ou des enregistrements biométriques; et toute autre information liée ou pouvant être liée à un individu, par exemple des données médicales, scolaires, financières ou liées à l'emploi.

3.1.8 point de décision de politique [b-OASIS XACML]: entité de système qui évalue une politique applicable et prend une décision d'autorisation.

3.1.9 point d'application de politique [b-OASIS XACML]: entité de système qui effectue un contrôle d'accès, en formulant des demandes de décision et en mettant en œuvre les décisions d'autorisation.

3.1.10 point d'informations de politique [b-OASIS XACML]: entité de système qui agit comme source des valeurs d'attribut.

3.2 Termes définis dans la présente Recommandation

La présente Recommandation définit les termes suivants:

3.2.1 certificat d'AE-ID: certificat relié par une chaîne de certificats à un certificat d'ancre de confiance et contenant un AE-ID dans l'extension subjectAltName.

NOTE – Un certificat d'AE-ID peut être utilisé pour vérifier que l'AE-ID a été assigné à une entité dans le certificat.

3.2.2 configuration de l'association: phase d'un cadre d'établissement d'association de sécurité au cours de laquelle l'entité qui établit l'association de sécurité (et le serveur central de distribution de clés, dans le cas des cadres de sécurité centralisés), reçoit les identités (et tout autre justificatif d'identité pertinent) pour assurer que l'association de sécurité est établie entre les entités prévues.

3.2.3 prise de contact de sécurité de l'association: phase d'un cadre d'établissement d'association de sécurité au cours de laquelle les points d'extrémité de l'association de sécurité s'authentifient mutuellement.

3.2.4 justificatif d'identité d'amorçage: justificatif d'identité préconfiguré permettant l'authentification mutuelle de l'entité inscrite et de la fonction d'inscription M2M.

3.2.5 configuration des justificatifs d'identité d'amorçage: phase d'un cadre de configuration à distance de la sécurité au cours de laquelle les justificatifs d'identité d'amorçage sont préconfigurés sur l'entité inscrite et la fonction d'inscription M2M.

3.2.6 prise de contact d'inscription d'amorçage: phase d'un cadre de configuration à distance de la sécurité au cours de laquelle l'entité inscrite et la fonction d'inscription M2M s'authentifient mutuellement.

3.2.7 configuration d'instruction d'amorçage: phase d'un cadre de configuration à distance de la sécurité au cours de laquelle des identités (et tout autre justificatif d'identité pertinent) sont fournies à l'entité inscrite et à la fonction d'inscription M2M pour permettre à la fonction d'inscription M2M d'établir un justificatif d'identité maître entre l'entité inscrite attendue et la fonction d'authentification M2M.

3.2.8 certificat: voir "certificat de clé publique".

3.2.9 autorité de certificat: autorité de certification.

3.2.10 chaîne de certificats: séquence d'un ou plusieurs certificats de CA, dans laquelle: la clé de vérification publique inscrite dans chaque certificat de CA est certifiée dans le certificat de CA précédent; et la clé publique du premier certificat de CA est considérée comme fiable a priori.

NOTE – La confiance dans la clé publique de chaque certificat de CA peut reposer sur la confiance dans le certificat de CA précédent.

3.2.11 nom de certificat: identificateur unique figurant dans un champ de nom d'un certificat (par exemple, l'attribut "Subject" or "Subject Alternative Name" dans la Recommandation UIT-T X.509).

3.2.12 configuration de certificat (procédure de): procédure exécutée par un principal de sécurité et une fonction MEF pour configurer, sur le principal de sécurité, un certificat configuré par la fonction MEF et un ou plusieurs certificats émanant de l'autorité de certificat de la fonction MEF.

NOTE – Des certificats d'autorités de certificat supplémentaires peuvent également être configurés par d'autres moyens, par exemple par préconfiguration ou comme décrit dans la Recommandation [UIT-T Y.4500.22].

3.2.13 reconfiguration de certificat (procédure de): procédure de configuration de certificat exécutée lorsque le principal de sécurité peut s'authentifier lui-même grâce à un certificat inscrit valide.

3.2.14 demande de signature de certificat: message de demande de certificat de clé publique.

3.2.15 vérification de certificat: processus permettant d'établir la fiabilité du certificat d'une entité.

3.2.16 autorité de certification [b-Menezes]: autorité chargée d'établir et de garantir l'authenticité de clés publiques.

NOTE – Ceci consiste notamment à lier les clés publiques à des noms distinctifs par le biais de certificats signés, à gérer les numéros de série des certificats et à révoquer les certificats.

3.2.17 clé de chiffrement de contenu: clé symétrique utilisée pour chiffrer du texte en clair et en obtenir le cryptogramme, et pour générer un contrôle d'intégrité du message (MIC).

NOTE – Dans le cas d'un chiffrement authentifié avec données associées (AEAD, *authenticated encryption with associated data*), la clé de chiffrement de contenu est utilisée directement, tandis qu'avec d'autres algorithmes, la clé de chiffrement de contenu sert à générer des clés distinctes pour l'algorithme de chiffrement et l'algorithme de protection de l'intégrité.

3.2.18 configuration des justificatifs d'identité: phase d'un cadre d'établissement d'association de sécurité au cours de laquelle les justificatifs d'identité requis par le cadre en question sont configurés sur les entités et les fonctions concernées.

3.2.19 ID de type d'un ID de justificatif d'identité: partie d'un identificateur de justificatif d'identité qui indique le type du justificatif d'identité que l'on identifie.

3.2.20 Certificat de CSE-ID: certificat relié par une chaîne de certificats à une racine de confiance et contenant un CSE-ID dans l'extension subjectAltName.

NOTE – Un certificat de CSE-ID peut être utilisé pour vérifier que le CSE-ID a été assigné à une entité dans le certificat.

3.2.21 certificat de dispositif: certificat relié par une chaîne de certificats à une racine de confiance et contenant au moins un identificateur unique mondialement d'instance matérielle dans l'extension subjectAltName.

NOTE – Un certificat de dispositif peut être utilisé pour vérifier qu'une entité s'exécute sur l'instance matérielle identifiée.

3.2.22 extrémité initiatrice ESCertKE: point d'extrémité ESCertKE qui initie la procédure ESCertKE (*End-to-end Security Certificate Key Establishment*).

3.2.23 messages ESCertKE: messages échangés entre l'extrémité initiatrice ESCertKE et l'extrémité de destination ESCertKE dans le cadre d'un établissement de clés fondé sur des certificats de bout en bout.

3.2.24 procédure ESCertKE: séquence de messages ESCertKE échangés et traitement fondé sur ces messages visant à établir des clés fondées sur des certificats de bout en bout.

3.2.25 extrémité de destination ESCertKE: point d'extrémité ESCertKE avec lequel l'extrémité initiatrice ESCertKE souhaite exécuter la procédure ESCertKE.

3.2.26 établissement de clés fondé sur des certificats de bout en bout: cadre interopérable permettant à deux points d'extrémité d'utiliser des certificats pour établir des clés symétriques secrètes de bout en bout destinées à être utilisées dans d'autres cadres de sécurité de bout en bout, par exemple le cadre ESData (*End-to-End Security of Data*) ou ESPrim (*End-to-End Security of Primitives*).

3.2.27 sécurité des données de bout en bout: cadre interopérable pour la protection des données transportées *in fine* à l'aide de points de référence oneM2M, afin qu'il ne soit pas nécessaire de confier la sécurité de ces données aux CSE traversées.

3.2.28 sécurité des primitives de bout en bout: cadre interopérable destiné à sécuriser l'échange de primitives oneM2M, afin qu'il ne soit pas nécessaire de confier la confidentialité et l'intégrité de ces primitives aux CSE.

3.2.29 entité inscrite: AE ou CSE requérant la configuration à distance d'une clé symétrique qui sera partagée avec une cible d'inscription.

3.2.30 clé d'inscription: clé symétrique établie entre une entité inscrite et une fonction d'inscription M2M lorsque l'authentification mutuelle a réussi.

NOTE – Une clé symétrique qui sera partagée entre l'entité inscrite et une cible d'inscription peut être dérivée (au niveau de l'entité inscrite et de la fonction d'inscription M2M) de la clé d'inscription actuellement valide, la fonction d'inscription M2M assurant ensuite la distribution sécurisée de la clé symétrique à la cible d'inscription.

3.2.31 génération de clé d'inscription: phase d'un cadre de configuration à distance de la sécurité au cours de laquelle l'entité inscrite et la fonction d'inscription M2M établissent une clé d'inscription et un identificateur de clé d'inscription.

3.2.32 phase d'inscription: étape du cycle de vie d'un équipement M2M au cours de laquelle cet équipement est configuré pour fonctionner avec un fournisseur de services M2M spécifique.

3.2.33 cible d'inscription: fonction d'authentification M2M, CSE, ou AE avec laquelle une entité inscrite souhaite établir une clé symétrique (justificatif d'identité maître ou clé de connexion sécurisée préconfigurée) via une configuration à distance de la sécurité.

3.2.34 identificateur d'entité: CSE-ID (ou, respectivement, AE-ID) d'une CSE (ou, respectivement d'une AE).

3.2.35 enveloppe ESData: objet de données contenant le résultat de la protection d'une charge utile de données sécurisées de bout en bout (ESData) au moyen des procédures ESData.

3.2.36 charge utile ESData: données qui doivent être protégées par le cadre de sécurité des données de bout en bout (ESData).

3.2.37 certificat FQDN: certificat relié par une chaîne de certificats à une racine de confiance et contenant un nom de domaine complet (FQDN, *fully qualified domain name*).

3.2.38 architecture d'amorçage générique: ensemble de spécifications 3GPP et 3GPP2 fournissant des fonctions de sécurité et un mécanisme chargé d'assurer l'authentification à l'amorçage et la concordance des clés pour la sécurité des applications, à partir des mécanismes d'authentification 3GPP et 3GPP2 sur le réseau sous-jacent.

3.2.39 configuration initiale de certificat (procédure de): procédure de configuration de certificat exécutée lorsque le principal de sécurité ne peut pas s'authentifier lui-même au moyen d'un certificat inscrit valide.

3.2.40 primitive de demande interne: primitive de demande destinée à être protégée au moyen d'un cadre de sécurité de bout en bout des primitives (ESPrim).

3.2.41 primitive de réponse interne: primitive de réponse destinée à être protégée au moyen d'un cadre de sécurité de bout en bout des primitives (ESPrim).

3.2.42 code d'intégrité de message: étiquette calculée à partir d'un message et d'une clé symétrique, et adjointe au message.

NOTE 1 – L'objectif d'un code d'intégrité de message est de faciliter, sans aucun mécanisme supplémentaire, l'obtention des garanties relatives à la source d'un message ainsi qu'à son intégrité.

NOTE 2 – Un code d'intégrité de message est parfois appelé "code d'authentification de message"; l'expression "code d'intégrité de message" est employée depuis qu'il existe un risque de confusion entre abréviations, "MAC" pouvant aussi bien désigner, en anglais, *message authentication code* ou *media access control*. La présente définition s'appuie sur celle donnée à la page 323 du document [b-Menezes] (p. 323).

3.2.43 clé de connexion sécurisée M2M: clé symétrique établie entre deux entités (CSE ou AE), par une fonction d'authentification M2M, afin de sécuriser la communication entre ces deux entités.

NOTE – Cette clé de connexion sécurisée M2M est le produit d'une procédure d'établissement d'association de sécurité M2M exécutée avec succès.

3.2.44 générateur de confiance M2M: partie prenante à laquelle est confiée l'authentification d'entités CSE/AE vis-à-vis d'autres CSE/AE.

3.2.45 client MAF: CSE ou AE configurée pour utiliser les services d'une fonction d'authentification M2M.

3.2.46 procédure de configuration de justificatif d'identité MAF: procédure de cadre de sécurité MAF utilisée pour la phase d'inscription d'un point d'extrémité, qui établit les justificatifs d'identité pour l'authentification mutuelle entre un point d'extrémité et une fonction MAF.

3.2.47 procédure de prise de contact MAF: procédure de cadre de sécurité fondé sur une fonction d'authentification M2M (MAF) au cours de laquelle une entité et la fonction MAF s'authentifient mutuellement et génèrent une clé symétrique qui peut ensuite être utilisée pendant la prise de contact de sécurité de l'association, en vue de l'authentification mutuelle entre l'entité et d'autres entités.

3.2.48 procédure d'enregistrement de clé MAF: procédure du cadre de sécurité fondé sur une fonction d'authentification M2M (MAF) au cours de laquelle un point d'extrémité source et la fonction MAF génèrent une clé symétrique qui peut ensuite être utilisée pour l'authentification mutuelle entre le point d'extrémité source et un ou plusieurs points d'extrémité cibles.

3.2.49 procédure de récupération de clé MAF: procédure du cadre de sécurité fondé sur une fonction d'authentification M2M (MAF) au cours de laquelle un point d'extrémité cible récupère la clé symétrique générée au préalable par la fonction MAF et un point d'extrémité source, pour permettre l'authentification mutuelle entre le point d'extrémité source et le point d'extrémité cible.

3.2.50 justificatifs d'identité maîtres: justificatifs d'identité utilisés pour l'authentification mutuelle entre une ASN-CSE ou une MN-CSE et fondé sur une fonction d'authentification M2M (MAF). Cette authentification sécurise l'accès à l'infrastructure d'un fournisseur de services M2M.

NOTE – Les justificatifs d'identité maîtres peuvent être préconfigurés ou configurés à distance (sans s'appuyer sur les justificatifs en question).

3.2.51 autorité de certificat MEF: rôle d'une autorité de certificat qui transmet des certificats configurés par une fonction d'inscription M2M (MEF) à un principal de sécurité, par le biais de cette même fonction.

NOTE – Ce terme se rapporte à la fonction MEF, ainsi une autorité de certificat MEF pour une fonction MEF donnée ne constitue pas une autorité de certificat MEF pour une autre fonction MEF.

3.2.52 client MEF: fonctionnalité chargée d'exécuter des procédures fondées sur une fonction d'inscription M2M (MEF) pour le compte d'une CSE ou d'une AE associée, ou pour le compte d'une CSE ou d'AE résidant sur un nœud associé, ou sur une fonction d'authentification M2M (MAF) associée.

3.2.53 certificat configuré par une fonction MEF: certificat émis par une autorité de certificat, par le biais d'une fonction MEF, pour authentifier le principal de sécurité.

NOTE – Ce terme se rapporte à la fonction MEF, ainsi un certificat émanant d'une fonction MEF donnée ne constitue pas un certificat pour une autre fonction MEF.

3.2.54 principal de sécurité (oneM2M): CSE, AE, nœud ou dispositif M2M qui peut être authentifié.

NOTE – Lorsque le principal de sécurité est un nœud ou un dispositif M2M, ce nœud ou ce dispositif M2M agit pour le compte de la CSE et/ou de l'AE exécutées sur ce dernier.

3.2.55 protocole de statut en ligne de certificat: protocole chargé de demander le statut d'un ou plusieurs certificats UIT-T X.509 [IETF RFC 6960].

3.2.56 phase opérationnelle: période du cycle de vie d'un équipement M2M pendant laquelle celui-ci est effectivement utilisé pour fournir des services M2M.

3.2.57 primitive de demande externe: primitive de demande utilisée pour transporter l'objet de données contenant une primitive de demande intérieure à laquelle a été appliquée un cadre de sécurité de bout en bout des primitives (ESPrim).

3.2.58 primitive de réponse externe: primitive de réponse utilisée pour transporter l'objet de données contenant une primitive de réponse intérieure à laquelle a été appliquée un cadre de sécurité de bout en bout des primitives (ESPrim).

3.2.59 point de stockage des politiques: entité système chargée de rechercher une politique ou un ensemble de politiques applicables.

3.2.60 clé de connexion sécurisée préconfigurée: clé symétrique qui est préconfigurée sur deux entités (lesquelles peuvent être des AE ou des CSE) et doit être utilisée pour l'authentification mutuelle de ces entités lors de l'établissement d'association de sécurité.

3.2.61 identificateur de clé de connexion sécurisée préconfigurée: identificateur d'une clé de connexion sécurisée préconfigurée.

3.2.62 clé symétrique d'entité inscrite préconfigurée: clé symétrique préconfigurée sur l'entité inscrite et la fonction d'inscription M2M.

3.2.63 identificateur de clé symétrique d'entité inscrite préconfigurée: identificateur d'une clé symétrique d'entité inscrite préconfigurée.

3.2.64 clé de signature privée: clé secrète qui peut générer des signatures pouvant être vérifiées au moyen d'une clé publique de vérification correspondante.

3.2.65 certificat de clé publique: document électronique qui utilise une signature numérique pour lier une clé publique à une identité.

NOTE – [b-Menezes] Un *certificat de clé publique* est une structure de données constituée d'une partie données et d'une partie signature. La partie données contient du texte en clair notamment, au minimum, une clé publique [de vérification] et une chaîne d'identification de la partie (entité sujet) à laquelle elle doit être associée. La partie signature est composée de la signature numérique de la partie données par une autorité de certification, ce qui permet de lier l'identité de l'entité sujet à la clé publique en question.

3.2.66 variante de certificat de clé publique: nom décrivant l'utilisation d'un certificat de clé publique dans le système oneM2M

3.2.67 infrastructure de clés publiques: ensemble de matériels, logiciels, personnes, politiques et procédures nécessaires pour créer, gérer, distribuer, utiliser, stocker et révoquer des certificats de clés publiques. Voir le document [b-Menezes] pour plus d'informations.

3.2.68 clé de vérification publique: justificatif d'identité qui permet de vérifier des signatures numériques générées par une clé de signature privée, mais qui ne peut pas être utilisé pour générer des signatures numériques.

3.2.69 certificat de clé publique brute: certificat comprenant uniquement la structure SubjectPublicKeyInfo d'un certificat UIT-T X.509 qui comporte les paramètres nécessaires pour décrire la clé publique [IETF RFC 7250].

3.2.70 autorité d'enregistrement: entité fonctionnelle chargée de vérifier les demandes de signature de certificat et d'autoriser une autorité de certification à émettre les certificats correspondants.

3.2.71 identificateur de clé d'inscription relatif: partie d'un identificateur de clé d'inscription qui est unique dans le contexte d'une fonction d'inscription M2M.

3.2.72 établissement de l'association de sécurité: exécution séquentielle de la configuration des justificatifs d'identité, de la configuration de l'association et de la prise de contact de sécurité de l'association entre deux entités.

3.2.73 cadre d'établissement d'association de sécurité: cadre de sécurité servant à établir une association de sécurité.

3.2.74 cadre d'amorçage de sécurité ou **cadre de configuration à distance de la sécurité**: mécanisme de configuration à distance d'un justificatif d'identité maître et d'un identificateur de justificatif d'identité maître sur une entité inscrite et une fonction d'authentification M2M.

3.2.75 environnement sécurisé: entité logique qui protège les données sensibles et les fonctions sensibles contre la falsification et la surveillance ou l'exécution non autorisées, et qui donne accès à ces données et fonctions sensibles aux entités oneM2M autorisées.

3.2.76 cadre de sécurité: ensemble de procédures assurant l'établissement de l'association de sécurité ou la configuration à distance de la sécurité.

3.2.77 identificateur d'utilisation de sécurité: identifie une fonction de sécurité (cadre d'établissement d'association de sécurité, cadre de sécurité de bout en bout des primitives ou cadre de sécurité de bout en bout des données, par exemple), un protocole utilisé pour cette fonction de sécurité et, le cas échéant, une option au sein d'un protocole donné.

NOTE – L'identificateur d'utilisation de sécurité sert à limiter la façon dont un justificatif d'identité peut être utilisé.

3.2.78 fonction sensible: fonction exécutée dans l'environnement sécurisé qui nécessite d'être protégée contre la surveillance, la falsification ou l'exécution non autorisées et qui manipule des données sensibles. Il peut s'agir par exemple d'une fonction de calcul de clés à partir de clés à longue durée de validité de la couche service M2M et d'algorithmes cryptographiques.

3.2.79 certificat autosigné: certificat de clé publique signé par l'entité même dont il certifie l'identité.

3.2.80 point d'extrémité ESData source: entité produisant une enveloppe de données de sécurité de bout en bout (ESData) à partir d'une charge utile ESData.

3.2.81 clé symétrique: clé secrète partagée entre deux entités.

3.2.82 point d'extrémité ESData cible: entité produisant la charge utile de données de sécurité de bout en bout (ESData) à partir d'une enveloppe ESData.

3.2.83 certificat d'ancre de confiance: certificat estimé fiable a priori.

3.3 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

(D)TLS-PSK	(D)TLS à clé pré-partagée (systèmes cryptographiques) (*(D)TLS pre-shared key (ciphersuites)*)
AAA	authentification, autorisation et comptabilité (*authentication, authorization and accounting*)
ABAC	contrôle d'accès fondé sur des attributs (*attribute based access control*)
ACP	instance "AccessControlPolicy" (*accesscontrolpolicy instance*)
AE	entité d'application (*application entity*)
AEAD	chiffrement authentifié avec données associées (*authenticated encryption with associated data*)
AE-ID	identificateur d'entité d'application (*application entity identifier*)
API	interface de programmation d'application (*application programming interface*)
App-ID	identificateur d'application (*application identifier*)
ASN-CSE	CSE qui réside dans le nœud de service d'applications (*application service node-CSE*)

BSF	fonction de serveur d'amorçage (*bootstrapping server function*)
B-TID	identificateur de transaction d'amorçage (*bootstrapping transaction identifier*)
CA	autorité de certification ou de certificat (*certification authority* ou *certificate authority*)
CIDR	routage entre domaines sans classification (*classless inter-domain routing*)
CoAP	protocole d'application avec contraintes (*constrained application protocol*)
CSE	entité de services communs (*common service entity*)
CSE-ID	identificateur d'entité de services communs (*common service entity identifier*)
CSR	demande de signature de certificat (*certificate signing request*)
DAS	système d'autorisation dynamique (*dynamic authorization system*)
DTLS	sécurité de la couche transport en mode datagramme (protocole) (*datagram transport layer security (protocol)*)
EKU	utilisation de clé étendue (*extended key usage*)
Enrolee-ID	identité de l'entité inscrite (*enrolee identity*)
ESCertKE	établissement de clé fondé sur un certificat de sécurité de bout en bout (*end-to-end security certificate-based key establishment*)
ESData	sécurité de bout en bout des données (*end-to-end security of data*)
ESF	fonction de sécurité de bout en bout (*end-to-end security function*)
ESPrim	sécurité de bout en bout des primitives (*end-to-end security of primitives*)
EST	enrolment over secure transport (protocole cryptographique)
FQDN	nom de domaine complet (*fully qualified domain name*)
GBA_ME	architecture GBA fondée sur l'équipement mobile (*ME-based GBA*)
GBA_U	architecture GBA comportant des fonctions améliorées sur cartes UICC (*GBA with UICC-based enhancements*)
GUSS	réglages de sécurité utilisateur de la GBA (*GBA user security settings*)
HLR	enregistreur de localisation nominal (*home location register*)
HSS	système d'abonné de rattachement (*home subscriber system*)
HTTP	HyperText Transfer Protocol
HW	matériel (*hardware*)
ID	identificateur
IdA	identificateur de l'entité A
IdB	identificateur de l'entité B
IN	nœud d'infrastructure (*infrastructure node*)
IN-CSE	CSE qui réside dans le nœud d'infrastructure (*infrastructure node-CSE*)
IPv4	version 4 du protocole Internet (*Internet protocol version 4*)
IPv6	version 6 du protocole Internet (*Internet protocol version 6*)
IV	vecteur d'initialisation (*initialization vector*)
JWE	chiffrement web JSON (*JSON web encryption*)

JWS	signature web JSON (*JSON web signature*)
JWT	Jetons web JSON (*JSON Web Token*)
Kc	clé de connexion sécurisée M2M (*M2M secure connection key*)
KcID	identificateur de clé de connexion sécurisée M2M (*M2M secure connection key identifier*)
Ke	clé d'inscription (*enrolment key*)
KeID	identificateur de clé d'inscription (*enrolment key identifier*)
Ker	clé de réauthentification d'inscription (*enrolment re-authentication key*)
Km	justificatif d'identité maître (*master credential*)
KmID	identificateur de justificatif d'identité maître (*master credential identifier*)
Kpm	justificatif d'identité préconfiguré utilisé pour configurer le justificatif d'identité maître (*pre-provisioned credential for master credential provisioning*)
KpmID	identificateur du justificatif d'identité préconfiguré utilisé pour configurer le justificatif d'identité maître (*pre-provisioned credential for master credential provisioning identifier*)
Kpsa	justificatif d'identité configuré pour l'établissement de l'association de sécurité M2M (*provisioned credential for M2M security association establishment*)
KpsaID	identificateur du justificatif d'identité configuré pour l'établissement de l'association de sécurité M2M (*provisioned credential for M2M security association establishment identifier*)
Ks	données de clé temporaire servant dans l'architecture GBA (*temporary key material referred to in GBA*)
Ks. NAF	abréviation de Ks_(int/ext)_NAF
Ks_(ext/int)_NAF	clé dérivée dans une GBA_ME, ou clé dérivée dans une GBA_U qui demeure sur la carte UICC (*derived key in GBA_ME or derived key in GBA_U which remains on UICC*)
Ks_ext_NAF	clé dérivée dans une GBA_U envoyée à l'équipement mobile (*derived key in GBA_U sent to the ME*)
Ks_int_NAF	clé dérivée dans une GBA_U qui demeure sur la carte UICC (*derived key in GBA_U which remains on UICC*)
Ks_NAF	clé dérivée dans l'équipement mobile (*derived key in the ME*)
M2M	de machine à machine (*machine to machine*)
M2M-SP	fournisseur de services M2M (*M2M service provider*)
MAF	fonction d'authentification M2M (*M2M authentication function*)
MAF-ID	identificateur de fonction d'authentification M2M (*M2M authentication function identifier*)
Mca	point de référence pour la communication M2M avec l'AE (*reference point for M2M communication with AE*)
Mcc	point de référence pour la communication M2M avec la CSE (*reference point for M2M communication with CSE*)

Mcc'	point de référence pour la communication M2M avec la CSE d'un fournisseur de services M2M différent (*reference point for M2M communication with CSE of different M2M service provider*)
Mcn	point de référence pour la communication M2M avec la NSE (*reference point for M2M communication with NSE*)
MEF	fonction d'inscription M2M (*M2M enrolment function*)
MIC	code d'intégrité de message (*message integrity code*)
MN-CSE	CSE qui réside dans le nœud intermédiaire (*middle node CSE*)
MTE	générateur de confiance M2M (*M2M trust enabler*)
NAF	fonction d'application de réseau (*network application function*)
OCSP	protocole de statut du certificat en ligne (*online certificate status protocol*)
OTA	hertzien (*over the air*)
PDP	point de décision de politique (*policy decision point*)
PEP	point d'application de politique (*policy enforcement point*)
PII	informations d'identification personnelle (*personally identifiable information*)
PIP	point d'informations de politique (*policy information point*)
PKI	infrastructure de clés publiques (*public key infrastructure*)
PRP	point de stockage des politiques (*policy retrieval point*)
RA	autorité d'enregistrement (*registration authority*)
RSPF	cadre de configuration à distance de la sécurité (*remote security provisioning framework*)
SAEF	cadre d'établissement d'association de sécurité (*security association establishment framework*)
SCEP	Simple Certificate Enrolment Protocol
SE	environnement sécurisé (*secure environment*)
SUID	identificateur d'utilisation de sécurité (*security usage identifier*)
SW	logiciel (*software*)
T&C	conditions générales (*terms and conditions*)
TEE	environnement d'exécution fiable (*trusted execution environment*)
TEF	fonction de génération de confiance (*trust enabling function*)
TLS	sécurité dans la couche transport (*transport layer security*)
UE	équipement d'utilisateur (*user equipment*)
UICC	carte à circuit intégré universelle (*universal integrated circuit card*)
UNSP	fournisseur de services de réseau sous-jacent (*underlying network service provider*)
URI	identificateur uniforme de ressource (*uniform resource identifier*)
USS	réglages de sécurité utilisateur (*user security settings*)
XACML	langage de balisage extensible de contrôle d'accès (*eXtensible access control markup language*)

3.4 Symboles

Pour les besoins de la présente Recommandation, le symbole suivant est utilisé:

|| Concaténation

4 Conventions

Les mots clés "doit", "ne doit pas", "devrait", "ne devrait pas", "peut" et "n'a pas besoin" utilisés dans la présente Recommandation doivent être interprétés comme décrit ci-après:

Doit/ne doit pas:

Exigences

1) Effet pour la norme: la norme doit décrire la fonctionnalité requise (à savoir spécifier une solution technique pour l'exigence).
2) Effet pour les produits: chaque mise en œuvre (solution M2M conforme à la Recommandation) doit prendre en charge la fonctionnalité.
3) Effet pour les déploiements: chaque déploiement (service M2M fondé sur la Recommandation) doit utiliser la fonctionnalité normalisée le cas échéant, faute de quoi, par exemple, des problèmes d'interopérabilité avec les autres services pourraient se poser.

Devrait/ne devrait pas:

Recommandation

1) Effet pour la norme: la norme doit décrire une solution qui permet à la fonctionnalité d'être présente ou absente.
2) Effet pour les produits: une mise en œuvre peut ou non prendre en charge la fonctionnalité; toutefois, la prise en charge est recommandée.
3) Effet pour les déploiements: un déploiement peut ou non utiliser la fonctionnalité; toutefois, l'utilisation est recommandée.

Peut/n'a pas besoin:

Permission/option

1) Effet pour la norme: la norme doit décrire une solution qui permet à la fonctionnalité d'être présente ou absente.
2) Effet pour les produits: une mise en œuvre peut ou non prendre en charge la fonctionnalité.
3) Effet pour les déploiements: un déploiement peut ou non utiliser la fonctionnalité.

5 Architecture de sécurité

5.1 Vue d'ensemble

5.1.0 Introduction

La Figure 5.1.0-1 donne une vue d'ensemble de haut niveau de l'architecture de sécurité.

L'architecture est composée des couches suivantes:

Couche des fonctions de sécurité

Cette couche contient un ensemble de fonctions de sécurité exposées au niveau des points de référence Mca et Mcc. Ces fonctions de sécurité peuvent être classées en six catégories: identification, authentification, autorisation, association de sécurité, traitement des données sensibles et administration de la sécurité.

Couche d'abstraction de l'environnement de sécurité

Cette couche met en œuvre diverses fonctions de sécurité telles que la dérivation de clés, le chiffrement et le déchiffrement de données, la génération et la vérification de signatures, la lecture et l'écriture de justificatifs de sécurité depuis/vers des environnements sécurisés, etc. Les fonctions de sécurité de la couche des fonctions de sécurité invoquent les présentes fonctions afin de protéger les opérations dans les environnements sécurisés. Cette couche assure également l'accès physique aux environnements sécurisés. La mise en œuvre de cet accès ne relève pas de la présente Recommandation. Cette couche n'est pas détaillée dans la présente version mais devrait être prise en compte dans les versions futures.

Couche des environnements sécurisés

Cette couche contient un ou plusieurs environnements sécurisés qui fournissent divers services de sécurité assurant une protection adéquate du stockage des données sensibles et de l'exécution des fonctions sensibles. Les données sensibles comprennent la capacité de mise en œuvre d'environnements sécurisés (SE), les clés de sécurité telles que les clés symétriques à longue durée de validité et les clés privées asymétriques, les justificatifs d'identité locaux, les politiques de sécurité, les informations d'identité, les informations d'abonnement, etc. Les fonctions sensibles comprennent le chiffrement des données, le déchiffrement des données, et autres. Bien que la mise en œuvre d'environnements sécurisés ne relève pas du champ de la présente Recommandation, un cadre de référence pour l'interface des entités M2M avec les cartes UICC est fourni à l'Annexe D.

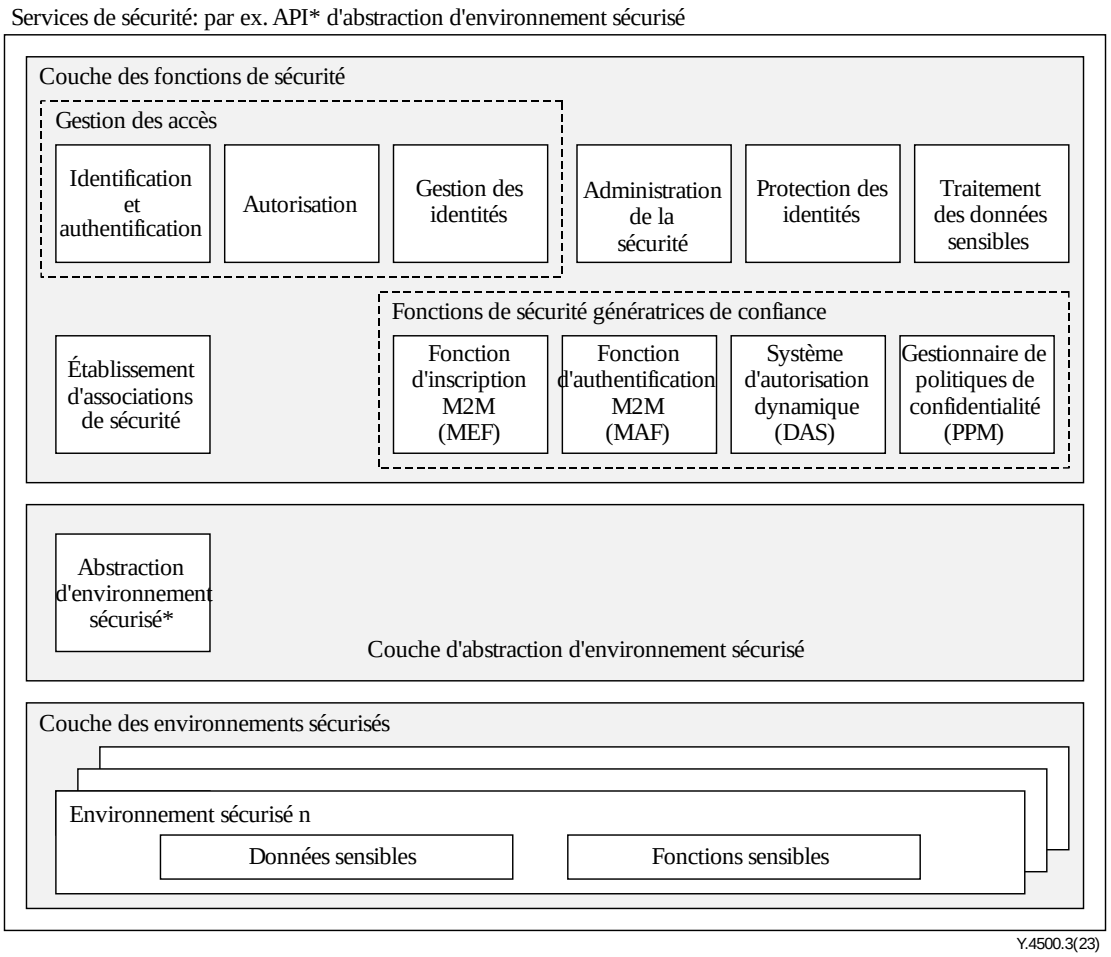


Figure 5.1.0-1 – Vue d'ensemble de haut niveau de l'architecture de sécurité

Principes de conception

Les services de sécurité sont modulaires et configurables en fonction des besoins de l'entité de services communs (CSE) hôte, des points de référence qu'elle prend en charge, et de son objectif.

L'architecture est divisée en plusieurs composants et sous-composants, qui en constituent la structure modulaire. Cette modularité permet de mapper l'architecture sur les différents nœuds et les différentes entités.

En fonction des exigences de chaque entité, la sécurité comprend les composants adéquats pour répondre aux besoins du nœud ou de l'entité respectifs et du cas d'utilisation prévu.

On adaptera l'architecture pour qu'elle puisse être mise en œuvre dans des entités différentes. Par exemple, l'architecture peut être mappée sur des classes de dispositifs différentes.

Le composant d'administration de la sécurité est censé permettre l'administration de toutes les ressources sensibles (données et fonctions), mais aussi la configuration et l'extension des services de sécurité eux-mêmes.

L'environnement sécurisé au sein de la CSE est accessible via la couche d'abstraction de l'environnement sécurisé et est censé fournir un niveau de protection adéquat aux informations sensibles énumérées au § 6.2.3.2.

Gestion des accès

Les trois composants suivants font partie de la couche de gestion des accès. Consulter le § 6.2 pour plus d'informations.

5.1.1 Identification et authentification

La fonction d'identification et d'authentification assure l'identification et l'authentification mutuelle des entités de services communs (CSE) et des entités d'applications (AE).

L'identification est l'opération consistant à vérifier si l'identité fournie à des fins d'authentification est valide. La manière d'effectuer une opération d'identification dépend de l'objectif de l'authentification. Par exemple, dans le cas de l'accès à une ressource, la fonction d'authentification peut nécessiter que l'identification vérifie si l'AE ou la CSE est enregistrée auprès de la CSE locale; dans le cas de l'enregistrement d'une AE ou d'une CSE, la fonction d'authentification peut nécessiter que l'identification vérifie si l'identité fournie par une AE ou une CSE correspond à un certificat. Une fois cette opération de vérification réussie, l'AE ou la CSE est identifiée et l'identité identifiée en question fournie à l'opération d'authentification.

L'authentification est l'opération qui consiste à valider si l'identité fournie lors de l'étape d'identification est associée à un justificatif d'identité de confiance. La manière d'effectuer une opération d'authentification dépend du mécanisme d'authentification mutuelle employé. Par exemple, dans le cas d'un mécanisme d'authentification fondé sur des certificats, la fonction d'authentification peut nécessiter que l'authentification vérifie une signature numérique; dans le cas d'un mécanisme d'authentification fondé sur des clés symétriques, la fonction d'authentification peut nécessiter que l'authentification vérifie un code d'intégrité de message (MIC). Une fois cette opération de validation terminée, l'AE ou la CSE est authentifiée.

5.1.2 Autorisation

La fonction d'autorisation est chargée d'autoriser les entités authentifiées à accéder aux services et aux données, en fonction des politiques de contrôle d'accès (ACP) configurées et des rôles assignés.

La politique de contrôle d'accès est un ensemble de conditions qui définissent si une entité peut accéder à une ressource protégée. La fonction d'autorisation peut prendre en charge différents mécanismes d'autorisation: liste de contrôle d'accès (ACL), contrôle d'accès fondé sur les rôles (RBAC), etc. La fonction d'autorisation peut avoir à évaluer plusieurs politiques de contrôle d'accès

lors d'une opération d'autorisation pour arriver à une décision finale de contrôle d'accès. Ce processus est décrit plus en détail au paragraphe 7 "Autorisation".

Le processus d'évaluation de l'autorisation repose sur la ressource d'abonnement aux services, qui précise à quels services M2M et à quels rôles de service M2M s'est abonnée l'entité authentifiée et indique les politiques de contrôle d'accès associées à la ressource protégée. Le processus d'évaluation d'autorisation peut également prendre en compte des attributs contextuels, par exemple l'heure ou l'emplacement géographique.

Avant l'autorisation, il est possible d'effectuer une authentification mutuelle entre la CSE ou l'AE à l'origine de la demande et la CSE hôte, comme indiqué au paragraphe 8. Le paragraphe 6.1.2.2.1 décrit les conditions dans lesquelles l'authentification mutuelle est obligatoire. Une règle de contrôle d'accès peut également inclure un indicateur précisant que la règle de contrôle d'accès ne s'applique que lorsque l'authentification mutuelle a été effectuée avec succès et que le résultat de l'authentification mutuelle est toujours valable; consulter le § 7.1.3 pour plus de détails.

5.1.3 Gestion des identités

La fonction de gestion des identités fournit des identités/identifiants oneM2M à l'entité requérante dans le cas où ces identités sont stockées dans l'environnement sécurisé. Les identificateurs oneM2M tels que définis dans l'architecture oneM2M ([UIT-T Y.4500.1]) peuvent également être traités comme des données sensibles accessibles aux AE ou CSE et utilisés indépendamment des fonctions d'authentification ou d'autorisation.

5.2 Couches de sécurité

5.2.1 Couche des fonctions de sécurité

Cette couche assure les services suivants:

- Gestion des accès:
  - Identification et autorisation.
  - Authentification.
  - Contrôle d'accès.
- Traitement des données sensibles:
  - Protection des fonctions sensibles.
  - Stockage sécurisé.
- Fonctions de sécurité génératrices de confiance:
  - MEF (fonction d'inscription M2M).
  - MAF (fonction d'authentification M2M).
  - DAS (système d'autorisation dynamique).
  - PPM (gestionnaire de politique de confidentialité).
- Établissement de l'association de sécurité
  - Connexion sécurisée via l'établissement d'une session sécurisée.
  - Connexion sécurisée via la sécurité des objets.
- Administration de la sécurité (y compris la configuration à distance de la sécurité).
- Protection des identités

Chacun de ces services fournit des fonctions et des ressources sur l'interface de programmation d'application (API) du service de sécurité et de l'administration.

5.2.2 Couche d'abstraction de l'environnement sécurisé

La couche d'abstraction de l'environnement sécurisé (non décrite dans la présente Recommandation) fournit un accès à l'environnement sécurisé via une API de transport de sécurité générale. Un module d'extension associé au type d'environnement sécurisé assure la connectivité physique/logique à l'environnement concerné. La couche d'abstraction de l'environnement sécurisé doit également être accessible sur la couche service.

5.3 Intégration dans l'architecture oneM2M globale

Les services de sécurité sont fournis au sein des composants architecturaux suivants et interagissent sur les différents points de référence, comme décrit dans la Recommandation [UIT-T Y.4500.1]. La Figure 5.3-1 présente l'architecture fonctionnelle oneM2M.

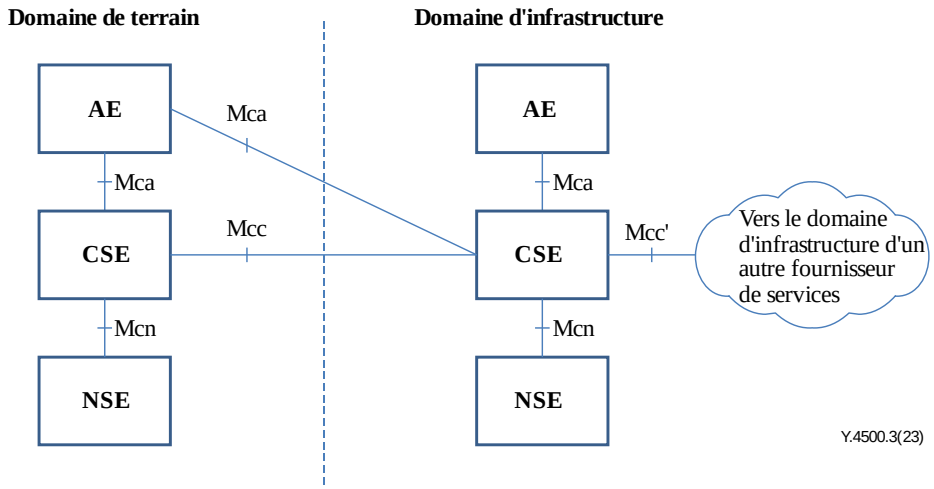


Figure 5.3-1 – Architecture fonctionnelle oneM2M

6 Services de sécurité et interactions

6.1 Intégration de la sécurité dans un flux d'événements oneM2M

6.1.1 Interactions entre couches

Avant qu'une procédure puisse avoir lieu sur la couche des services communs M2M, il est nécessaire d'établir la connectivité sur la couche service du réseau sous-jacent, ce qui peut impliquer des procédures de configuration et d'enregistrement de service demandées par le réseau sous-jacent.

Les procédures de configuration de sécurité de la couche service (préconfiguration de la sécurité ou amorçage de la sécurité) et d'établissement d'associations de sécurité décrites dans la présente Recommandation peuvent être exécutées indépendamment d'éventuelles procédures d'établissement de connectivité de la couche service du réseau, et généralement consécutivement à celles-ci.

Enfin, il faut tenir compte des exigences en matière de configuration de sécurité et d'établissement d'associations de sécurité imposées par les fournisseurs de services d'applications M2M. Au niveau de la couche service, l'établissement de l'association de sécurité donne lieu à une session de sécurité de la couche de transport (TLS) ou de sécurité de la couche de transport des datagrammes (DTLS) qui protège les messages échangés entre AE/CSE adjacentes, c'est-à-dire distantes d'un seul bond. Les AE qui doivent préserver la confidentialité de leurs échanges d'informations vis-à-vis d'associations intermédiaires non fiables permettent de chiffrer le contenu des ressources échangées entre les AE via la couche service. Dans certains scénarios (voir le § 8.2.1), l'établissement d'une association de sécurité entre AE/CSE adjacentes nécessite des sessions TLS ou DTLS distinctes pour chaque sens de transmission, c'est-à-dire qu'une paire de nœuds d'associations de sécurité peut être

configurée pour prendre en charge une association de sécurité directe entre elles. Les interactions directes des associations de sécurité entre les couches sont détaillées au § 8.5.

6.1.2 Séquence d'événements de haut niveau

6.1.2.1 Phase d'inscription

Les équipements M2M nécessitent généralement des phases de provisionnement et de configuration avant d'être mis en service. Cela peut se faire par une préconfiguration qui peut être intégrée dans la phase de fabrication ou de déploiement du produit, ou au moyen d'une procédure d'amorçage de la sécurité (c'est-à-dire une configuration à distance de la sécurité) qui a lieu avant que l'équipement ne commence à fonctionner.

Au niveau de la couche service, ce provisionnement et cette configuration nécessitent de choisir l'acteur qui fournira les services par le biais de l'équipement, en particulier le fournisseur de services M2M. Cette phase d'inscription nécessite que les parties prenantes s'accordent.

Une phase d'inscription peut se produire plusieurs fois pendant le cycle de vie d'un équipement M2M, mais elle n'est répétée que lorsqu'un changement au niveau du fournisseur de service affecte le provisionnement ou la configuration de l'équipement.

La phase de configuration de sécurité pour les différentes couches peut être combinée, grâce à une méthode commune de préconfiguration de la sécurité.

Les cadres de configuration à distance de la sécurité (RSPF, *remote security provisioning frameworks*) assurent la préconfiguration des informations essentielles à l'établissement d'une association de sécurité entre une entité sur le terrain et la fonction d'authentification M2M d'un fournisseur de services M2M sélectionné. Les informations de sécurité essentielles comprennent notamment les justificatif d'identité et les identificateurs. Les procédures de configuration à distance de la sécurité reposent sur une fonction d'inscription M2M qui peut être externe au fournisseur de services M2M pour établir les justificatifs d'identité adéquats.

Cadre de configuration à distance de la sécurité fondé sur une clé symétrique d'entité inscrite préconfigurée: une clé symétrique est préconfigurée pour l'entité inscrite et la fonction d'inscription M2M en vue de l'authentification mutuelle de ces entités. Consulter le § 8.3.2.1 pour plus d'informations.

Cadre de configuration à distance de la sécurité fondé sur des certificats: l'entité inscrite et la fonction d'inscription M2M se voient chacune attribuer une clé de signature privée et un certificat contenant la clé de vérification publique, ce qui leur permet de s'authentifier mutuellement. Consulter le § 8.3.2.2 pour plus d'informations.

Cadre de configuration à distance de la sécurité fondé sur une architecture GBA: cette méthode de configuration à distance concerne uniquement les dispositifs 3GPP (*3rd generation partnership project*). Dans ce cas, la fonction d'inscription M2M intègre la fonction de serveur d'amorçage GBA. Ce cadre utilise des clés symétriques 3GPP ou 3GPP2 (*3rd generation partnership project 2*) pour authentifier l'entité inscrite et la fonction d'inscription M2M (qui est également une fonction de serveur d'amorçage de la GBA). Les spécifications 3GPP TS 33.220 [ETSI TS 133 220] et 3GPP2 S.S0109-A [TIA-1098-A] précisent les détails de ce cadre de configuration. Consulter le § 8.3.2.3 pour plus d'informations.

NOTE 1 – L'utilisation de l'architecture GBA n'est prévue que pour la phase d'inscription. Elle permet d'obtenir un certain nombre de justificatifs d'identité symétriques qui peuvent être utilisés à n'importe quelle fin au niveau de la couche application, pas nécessairement pour l'authentification. Les spécifications oneM2M utilisent ensuite de ces justificatifs pour générer un certain nombre de justificatifs d'identité symétriques supplémentaires comme données d'entrée dans les protocoles de sécurité d'authentification à clé pré-partagée (PSK, *pre-shared key*) et de protection de la confidentialité et de l'intégrité, en particulier (D)TLS-PSK et les protocoles SAEF, ESPrim et ESData reposant sur des clés pré-partagées, comme décrit au § 8.3.2.3.

NOTE 2 – La spécification technique TS 33.221 "Support for subscriber certificates (SSC)" n'est pas prise en charge dans la présente Recommandation car de nombreuses autres façons reconnues de déployer un certificat dans un dispositif sont apparues depuis la publication du document TS 33.221. oneM2M prend en charge le protocole SCEP d'inscription simple par certificat au § 8.3.6.3 "Procédure de configuration de certificats utilisant le protocole SCEP" et l'inscription par transport sécurisé (EST) au § 8.3.6.2 "Procédure de configuration de certificats utilisant le protocole EST". Consulter le § 8.3.2.3 pour plus d'informations.

La Figure 6.1.2.1-1 illustre les différents cadres de configuration à distance de la sécurité. On notera l'absence de communication entre les entités M2M A et B pendant la procédure de configuration à distance de la sécurité. Une fois la procédure de configuration à distance de la sécurité exécutée avec succès, une procédure d'établissement d'association de sécurité est lancée.

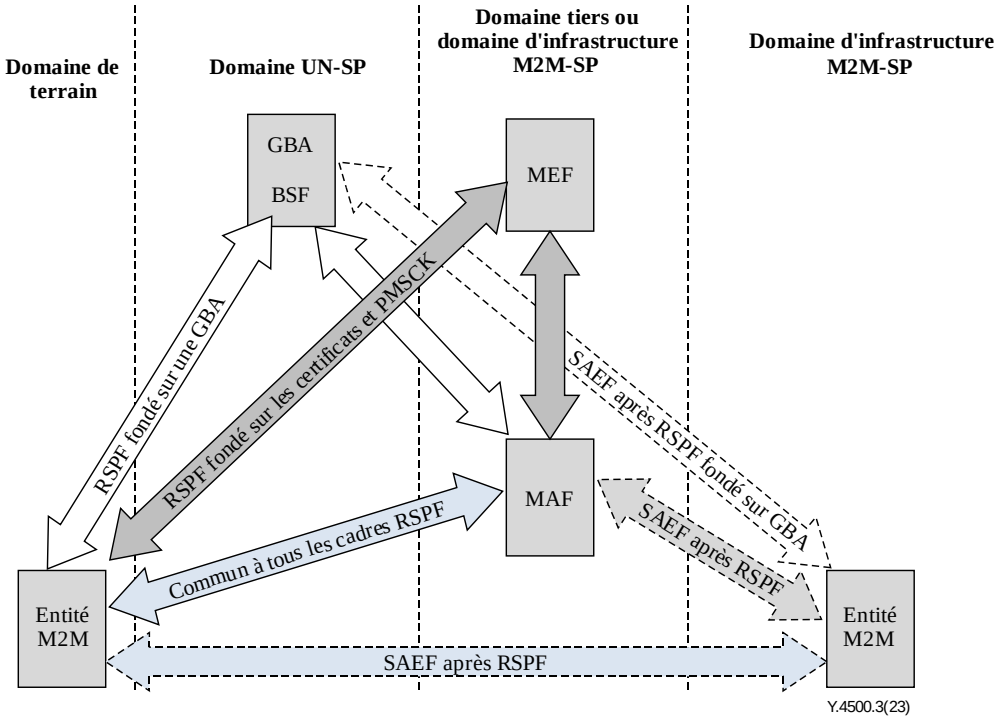


Figure 6.1.2.1-1 – Entités impliquées dans la configuration à distance de la sécurité

6.1.2.2 Phase opérationnelle

6.1.2.2.1 Accès aux services M2M

Les CSE proposent des services M2M aux AE ainsi qu'à d'autres CSE. Pour pouvoir utiliser les services M2M offerts par une CSE, les AE et/ou les CSE doivent être mutuellement identifiées et authentifiées auprès de cette CSE, pour être protégées contre les accès non autorisés et les attaques de type déni de service (DoS). Cette authentification mutuelle permet de fournir un chiffrement et une protection de l'intégrité lors de l'échange de messages au niveau d'un point de référence Mca, Mcc ou Mcc' unique. En outre, les AE communicantes qui ont besoin d'une protection similaire pour leurs propres échanges d'informations peuvent être configurées pour appliquer la même méthode de sécurité à leurs communications.

C'est là l'objectif de la procédure d'établissement d'association de sécurité, qui doit être exécutée avant les procédures liées aux services décrites dans la Recommandation [UIT-T Y.4500.1] pour le point de référence correspondant.

Sur les points de référence Mca et Mcc, il est obligatoire d'établir une association de sécurité entre une AE ou une CSE, respectivement, et une IN-CSE.

Sur le point de référence Mcc', il est obligatoire d'établir une association de sécurité entre une IN-CSE et une IN-CSE.

Sur le point de référence Mca, il est fortement recommandé d'établir une association de sécurité entre l'AE et la CSE sur le terrain.

NOTE 1 – L'établissement d'une association de sécurité sur l'interface Mca dans un domaine local est facultatif en fonction de l'évaluation des risques, par exemple dans les scénarios où l'accès non autorisé peut être empêché par d'autres mesures de sécurité hors du champ d'application de la présente spécification. Sont inclus les cas d'utilisation suivants:

L'AE et la CSE (c'est-à-dire les points d'extrémité Mca) sont intégrées de façon sécurisée sur le même dispositif physique (c'est-à-dire un ASN).

La communication sécurisée est garantie par le réseau sous-jacent (WLAN ou VPN, par exemple).

Les communications sur le Mca passent par un réseau filaire, par exemple Ethernet, dans un environnement physique sûr.

Les phases d'établissement de l'association de sécurité de la couche service M2M et de la couche application M2M sont généralement indépendantes de procédures similaires potentiellement requises par la couche réseau, bien qu'elles puissent reposer sur les services de sécurité fournis par cette dernière.

Le système oneM2M prend en charge les mécanismes d'authentification suivants pour l'établissement d'une association de sécurité. Ces mécanismes sont décrits plus en détail au § 8.2.1 "Vue d'ensemble des cadres d'établissement d'association de sécurité":

Cadre d'établissement d'association de sécurité fondé sur la configuration d'une clé symétrique: une clé symétrique est préconfigurée sur les points d'extrémité de l'association. Consulter le § 8.2.2.1 pour plus d'informations.

Cadre d'établissement d'association de sécurité fondé sur des certificats: les points d'extrémité de l'association s'authentifient au moyen de clés privées racines et de certificats contenant la clé de vérification publique correspondante. Consulter le § 8.2.2.2 pour plus d'informations.

Cadre d'établissement d'association de sécurité fondé sur une fonction d'authentification M2M: dans un SAEF fondé sur une fonction MAF, le serveur de distribution centralisée des clés est une fonction MAF hébergée par un fournisseur de services tiers lié par une relation de service au fournisseur de service M2M (M2M-SP), ou hébergée par le M2M-SP lui-même. La fonction MAF authentifie une entité de terrain pour le compte d'une IN-CSE au moyen d'une clé symétrique. Consulter le § 8.2.2.3 pour plus d'informations.

NOTE 2 – L'utilisation de l'architecture GBA n'est prévue que pour la phase d'inscription. Elle permet d'obtenir un certain nombre de justificatifs d'identité symétriques qui peuvent être utilisés à n'importe quelle fin au niveau de la couche application, pas nécessairement pour l'authentification. Les spécifications oneM2M utilisent ensuite de ces justificatifs pour générer un certain nombre de justificatifs d'identité symétriques supplémentaires comme données d'entrée dans les protocoles de sécurité d'authentification à clé pré-partagée (PSK) et de protection de la confidentialité et de l'intégrité, en particulier (D)TLS-PSK et les protocoles SAEF, ESPrim et ESData reposant sur des clés pré-partagées, comme décrit au § 8.3.2.3.

La Figure 6.1.2.2.1-1 illustre les différents cas d'utilisation et les entités impliquées dans les divers cadres d'établissement de la sécurité (SAEF) envisagés dans la présente Recommandation.

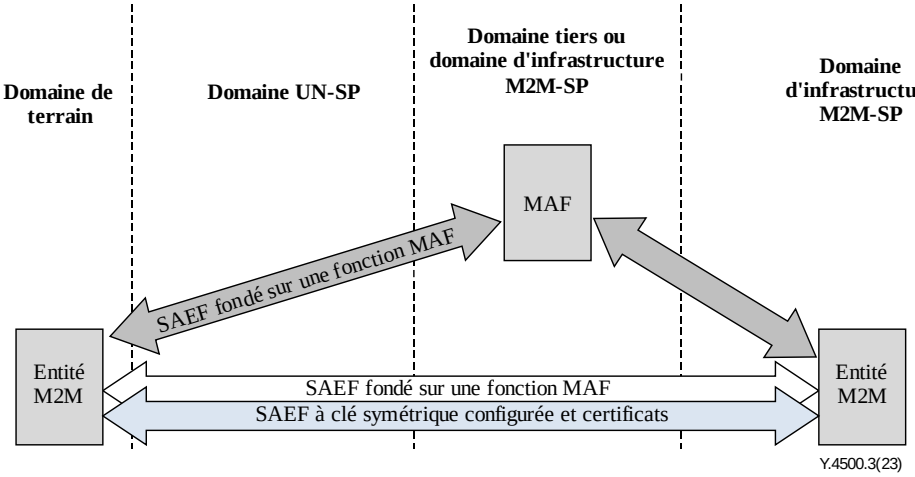


Figure 6.1.2.2.1-1 – Entités impliquées dans l'établissement d'une association de sécurité

6.1.2.2.2 Autorisation d'accès aux ressources M2M

Dès lors qu'une AE ou une CSE a obtenu un accès aux services M2M, la procédure de décision de contrôle d'accès abordée au § 7.1.5 de la présente Recommandation est exécutée avant d'accéder à une ressource M2M, comme décrit dans la Recommandation [UIT-T Y.4500.1].

6.2 Couche des fonctions de sécurité

6.2.1 Gestion des accès

Ce composant fournit des services d'authentification à la couche application. L'Annexe B propose une description générale des mécanismes d'authentification. Les mécanismes d'authentification mutuelle oneM2M permettent aux entités oneM2M de prouver qu'elles connaissent des justificatifs d'identité liés, par exemple un justificatif d'identité maître, sans devoir échanger la valeur de ces justificatifs d'identité ni des données sensibles comme les identités de sécurité et les identificateurs de sécurité. Pour empêcher la lecture et la copie des justificatifs d'identité, un environnement sécurisé intégré au cadre CSF de sécurité assure une protection contre l'altération de ces justificatifs et des informations de traitement associées.

6.2.1.1 Remarques relatives au stockage des clés à longue durée de validité

Les clés de connexion sécurisée configurées à longue durée de validité peuvent poser un risque de sécurité si elles ne sont pas protégées de façon adéquate, raison pour laquelle il est recommandé de les stocker dans des environnements sécurisés.

De même, les clés symétriques d'entité inscrite préconfigurées à longue durée de validité peuvent poser un risque de sécurité si elles ne sont pas protégées de façon adéquate; il est donc également conseillé de les stocker dans des environnements sécurisés.

Il est prévu, par définition, une grande diversité de dispositifs non conçus suivant les normes 3GPP, ainsi que de nombreuses implémentations différentes. Comme il est peu probable que ces dispositifs et ces implémentations soient normalisés, ce point n'entre pas dans le champ d'application des spécifications oneM2M ou 3GPP. Cependant, des guides de bonnes pratiques de sécurité sont disponibles dans les documents [b-ETSI TS 103 645], [b-IoTSF SD-BP] et [b-GSMA IoT-SGA].

6.2.2 Autorisation

La Figure 6.2.2-1 donne une vue d'ensemble de haut niveau d'une fonction d'autorisation générique. Une telle fonction comprend les quatre sous-composants décrits ci-après.

Point d'application de politique (PEP):

Le PEP intercepte les demandes d'accès aux ressources, envoie des demandes de décision de contrôle d'accès et met en œuvre ces décisions. Le PEP coexiste avec l'entité requérant des services d'autorisation.

Point de décision de politique (PDP):

Le PDP interagit avec le point de stockage des politiques (PRP) et le point d'informations de politique (PIP) pour obtenir les politiques d'autorisation applicables et les attributs nécessaires pour évaluer les politiques d'autorisation respectives. Il examine ensuite la demande d'accès sur la base des politiques d'autorisation, afin de renvoyer une décision de contrôle d'accès. Le PDP est situé dans le service d'autorisation.

Point de stockage des politiques (PRP):

Le PRP extrait les politiques d'autorisation applicables en fonction de la demande de décision de contrôle d'accès. Ces politiques applicables devraient être combinées pour arriver à une décision finale de contrôle d'accès. Le PRP est situé dans le service d'autorisation.

Point d'informations de politique (PIP):

Le PIP fournit les attributs requis pour évaluer les politiques d'autorisation, par exemple l'adresse IP du demandeur, l'heure de création de la ressource, l'heure actuelle ou la localisation géographique du demandeur. Le PIP est situé dans le service d'autorisation.

Le service d'autorisation peut comprendre l'un quelconque de ces sous-composants: PDP, PRP et/ou PIP, ce qui signifie que les sous-composants PEP, PRP, PDP et PIP peuvent être répartis sur plusieurs nœuds différents. Par exemple, le PEP pourra être situé dans un ASN/MN et le PDP dans l'IN.

La présente version ne prend pas en charge la séparation du PRP et du PIP sur une CSE différente de celle du PDP. La procédure générique décrite ci-dessous est fournie à titre d'information et pour prendre en charge des extensions ultérieures, le paragraphe 7 abordant plus en détail les mécanismes d'autorisation dans la version actuelle.

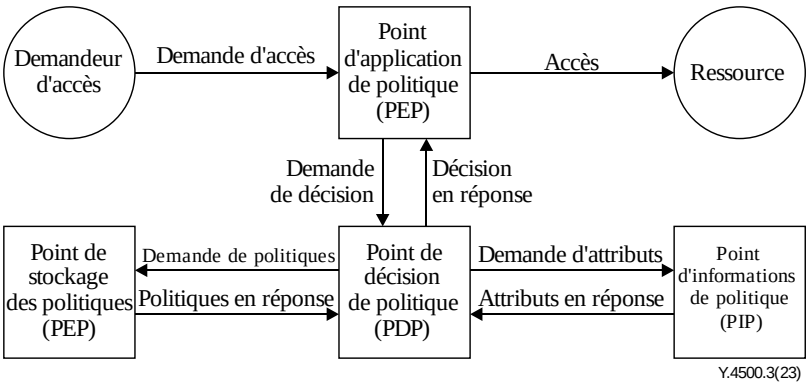


Figure 6.2.2-1 – Vue d'ensemble de la fonction d'autorisation

La procédure d'autorisation générique est illustrée à la Figure 6.2.2-2.

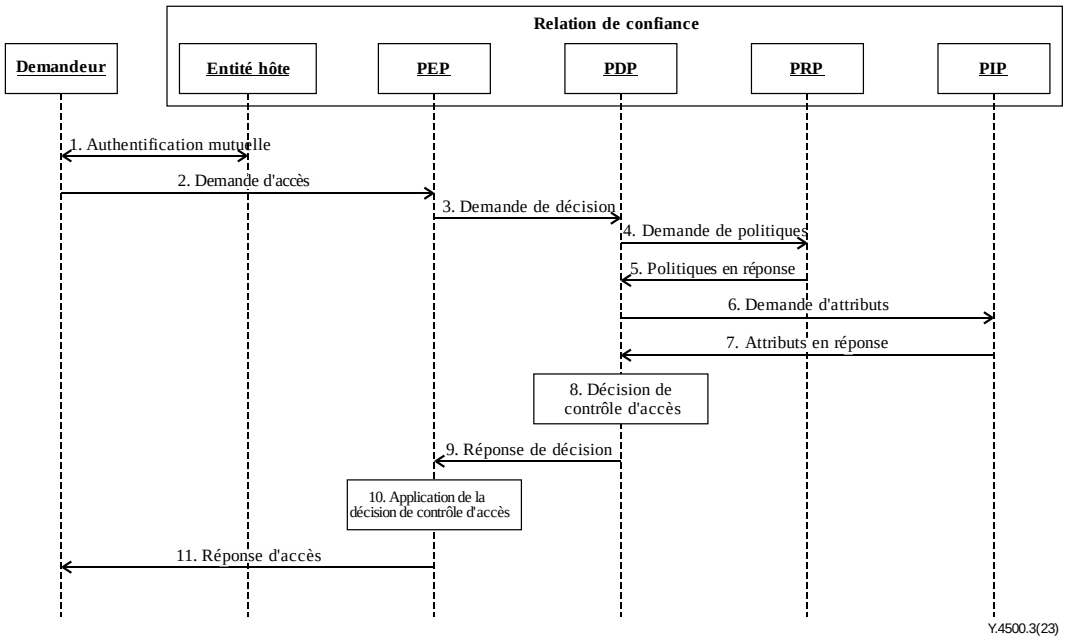


Figure 6.2.2-2 – Procédure d'autorisation

Étape 001: Authentification mutuelle (prérequis).

Étape 002: Le demandeur d'accès envoie une demande d'accès au PEP.

Étape 003: Le PEP génère, à partir de la demande d'accès du demandeur, une demande de décision de contrôle d'accès qu'il envoie au PDP.

Étape 004: Le PDP envoie au PRP une demande de politiques de contrôle d'accès basée sur la demande de décision de contrôle d'accès.

Étape 005: Le PRP récupère toutes les politiques de contrôle d'accès applicables à la demande d'accès, et les renvoie au PDP. Lorsque plusieurs politiques de contrôle d'accès entrent en jeu, le PRP renvoie également au PDP un algorithme de combinaison de politiques chargé de combiner les différents résultats d'évaluation en un résultat final.

Étape 006: Le PDP envoie, le cas échéant, une demande d'attributs au PIP s'il est nécessaire de connaître certains attributs pour appliquer ces politiques de contrôle d'accès.

Étape 007: Le PIP reçoit les attributs demandés et les transfère au PDP.

Étape 008: Le PDP évalue la demande d'accès au moyen des politiques de contrôle d'accès. Lorsque plusieurs politiques de contrôle d'accès entrent en jeu, le PEP doit calculer une décision finale de contrôle d'accès au moyen de l'algorithme de combinaison des politiques.

Étape 009: Le PDP renvoie la décision de contrôle d'accès au PEP.

Étape 010: Le PEP applique la décision de contrôle d'accès, c'est-à-dire qu'il transfère la demande d'accès à la ressource demandée, ou il rejette la demande.

Étape 011: Le PEP renvoie le résultat d'accès au demandeur d'accès.

6.2.3 Administration de la sécurité

6.2.3.0 Introduction

Le service d'administration de la sécurité offre la possibilité de gérer les fonctions de sécurité, les ressources et les attributs, ce qui inclut la gestion des ressources mises à disposition via l'environnement sécurisé. Ce service peut également fournir des fonctions pour gérer les données sensibles ainsi que les identificateurs et abonnements associés pour le compte d'autres entités.

L'administration de la sécurité dépend ainsi du type d'environnement sécurisé utilisé (module matériel indépendant, environnement d'exécution fiable intégré ou protection logicielle). Selon le type d'environnement sécurisé, différents standards parmi les standards existants peuvent être utilisés pour l'administration à distance de ces environnements.

6.2.3.1 Préconfiguration de la sécurité d'un environnement sécurisé

Certaines données sensibles et objets associés sont souvent configurés par l'établissement préalable d'un environnement sécurisé (voir le § 6.3.1 "Environnement sécurisé") avant le déploiement du dispositif M2M auquel sont associées ces données.

Les cartes UICC décrites dans les spécifications [ETSI TS 102 671] et [ETSI TS 102 221] sont couramment utilisées à cette fin car elles permettent d'accéder à certains réseaux sous-jacents, elles offrent un niveau de sécurité élevé et elles constituent une interface de transport interopérable détaillée dans la spécification [ETSI TS 102 221]. La préconfiguration oneM2M reposant sur les cartes UICC doit respecter le cadre décrit à l'Annexe D pour garantir l'interopérabilité.

6.2.3.2 Administration à distance de la sécurité d'un environnement sécurisé

Les données et fonctions sensibles de sécurité qui sont protégées et isolées au sein de l'environnement sécurisé peuvent rester accessibles à distance aux administrateurs de sécurité légitimes après le déploiement. L'administration de la sécurité à distance se distingue de la gestion standard des dispositifs par le fait que l'on s'attend à ce qu'un canal sécurisé soit établi entre le serveur d'administration et l'environnement sécurisé du nœud M2M (c'est-à-dire que le secret utilisé pour sécuriser la connexion n'est pas disponible dans le nœud M2M en dehors de l'environnement sécurisé). Les protocoles d'administration à distance de la sécurité applicables dépendent du niveau de risque de chaque application M2M et pas seulement des technologies réseau sous-jacentes. Des technologies répandues qui permettent d'administrer la sécurité à distance pour les différents niveaux de sécurité établis dans le rapport technique oneM2M TR-0008 [b-oneM2M TR0008] sont envisagées à l'Annexe C.

Étant donné que l'administration à distance de la sécurité exige que les informations sensibles cibles soient modifiables à distance, la protection de ces informations sensibles contre le piratage logiciel à distance du dispositif est particulièrement critique. Si l'environnement sécurisé repose uniquement sur une protection logicielle, l'administration à distance des données suivantes ne devrait être autorisée que si l'accès à distance par des attaquants potentiels peut être empêché:

– Clé privée et identificateurs associés.
– Clé symétrique partagée à longue durée de validité (par comparaison avec la durée de vie prévue du nœud M2M) et identificateurs associés.

Toute opération et paramètres associés qui manipulent les informations ci-dessus, c'est-à-dire les fonctions de sécurité.

6.2.4 Protection des identités

La protection des identités fournit des services à la couche application, par exemple des pseudonymes et la protection de l'anonymat des transactions.

6.2.5 Traitement des données sensibles

6.2.5.0 Introduction

Le service de traitement des données sensibles fournit certaines fonctions sensibles à la couche application.

Les fonctions sensibles comprennent les fonctions suivantes:

– Stockage sécurisé.
– Opérations de chiffrement.

– Méthodes d'amorçage des secrets initiaux (architecture GBA, par exemple).

6.2.5.1 Fonctions sensibles

Ce service permet aux AE et aux CSE d'accéder aux fonctions sensibles de l'environnement sécurisé.

6.2.5.2 Stockage sécurisé

Ce service permet aux AE et aux CSE d'accéder aux moyens de stockage sécurisé de l'environnement sécurisé. Les données stockées de manière sécurisée par l'AE ou la CSE sont censées être accessibles uniquement par l'API de sécurité et par les entités autorisées. Le stockage sécurisé devrait être géré par l'environnement sécurisé. Les données stockées de manière sécurisée sont destinées à rester sous le contrôle de la partie prenante propriétaire des données, c'est-à-dire l'entité qui a demandé que les données soient stockées dans le stockage sécurisé, indépendamment des autres parties prenantes.

6.2.6 Fonctions de sécurité génératrices de confiance

L'architecture de génération de confiance oneM2M peut nécessiter la présence de fonctionnalités de sécurité au sein du domaine d'infrastructure: fonction d'authentification M2M (MAF) et fonction d'inscription M2M (MEF), toutes deux étant considérées comme des fonctions génératrices de confiance (TEF, *trust enabling functions*) et assurant des fonctions d'authentification et de sécurité de bout en bout, mais aussi serveur de système d'autorisation dynamique (DAS) ou autorités en matière de rôle assurant des fonctions d'autorisation. La fonction MAF et les fonctions MEF doivent intégrer la capacité d'assurer l'inscription et l'enregistrement des justificatifs d'identité de bout en bout. Il est également possible de mettre en œuvre une fonctionnalité de gestion de politique de confidentialité (PPM, *privacy policy manager*) pour protéger la confidentialité des utilisateurs. Toutes ces fonctions peuvent être soit sous le contrôle du fournisseur de services M2M, soit déléguées à un générateur de confiance M2M (c'est-à-dire une partie à laquelle font confiance toutes les parties prenantes de l'écosystème M2M).

Fonction d'inscription M2M (MEF):

Utilisée pendant la phase d'inscription, la fonction MEF prend en charge la procédure d'amorçage de sécurité qui permet la configuration des justificatif d'identité maîtres à utiliser pour l'authentification mutuelle des entités qui accèdent à l'infrastructure d'un fournisseur de services M2M. La fonction MEF repose sur un justificatif d'identité initial préconfiguré dans le nœud M2M, par exemple lors de la fabrication.

Les justificatifs d'identité configurés par une fonction MEF peuvent être utilisés pour l'authentification au moyen d'une fonction MAF dans un cadre d'établissement d'association de sécurité (SAEF) fondé sur une fonction MAF, un cadre de sécurité de bout en bout des primitives (ESPrim) ou un cadre de sécurité de bout en bout des données (ESData). Les justificatifs d'identité configurés peuvent aussi être utilisés directement par les cadres SAEF, ESPrim ou ESData.

Fonction d'authentification M2M utilisée pendant la phase opérationnelle des services M2M:

Les justificatifs d'identité maîtres, servant à l'authentification mutuelle des CSE/AE pendant la phase opérationnelle, sont stockés de façon sécurisée dans une fonctionnalité d'infrastructure spécifique appelée fonction d'authentification M2M (MAF).

La fonction MAF conserve de façon sécurisée les justificatifs d'identité maîtres utilisés pour l'authentification des CSE/AE qui ont été inscrites par le fournisseur de services M2M ou le générateur de confiance M2M. La fonction MAF stocke les justificatifs d'identité maîtres et, éventuellement, les identificateurs des CSE/AE associées.

Une seule fonction MAF peut prendre en charge tous les services de sécurité des communications (SAEF, ESPrim and ESData), ou certains d'entre eux seulement. Une fonction MAF assurant un cadre d'établissement d'association de sécurité (MAF-SAEF) est exécutée par le fournisseur de services M2M, ou par un générateur de confiance M2M pour le compte du fournisseur de services M2M.

Un fournisseur de services M2M ou un générateur de confiance M2M peut exécuter d'autres fonctions MAF, et il n'est pas supposé qu'une relation de confiance existe entre le générateur de confiance M2M et le fournisseur de services M2M dans ces cas.

La fonction MAF est également chargée de toutes les opérations de sécurité faisant appel aux justificatifs d'identité maîtres.

Serveur de système d'autorisation dynamique (DAS) et autorités de rôles:

Ces fonctionnalités gèrent les privilèges d'autorisation d'accès aux ressources qui peuvent être attribués pendant le fonctionnement et sont respectivement décrites aux § 7.3 et 7.4.

Gestionnaire de politique de confidentialité (PPM):

Cette fonctionnalité aide à gérer les préférences en matière de confidentialité exprimées par le sujet des données vis-à-vis des exigences de service et de la règlementation en vigueur. Elle est décrite au paragraphe 11.

6.3 Environnement sécurisé et abstraction d'environnement sécurisé

6.3.1 Environnement sécurisé

L'environnement sécurisé est une entité logique qui fournit des fonctions sensibles utilisant des données sensibles, un stockage sécurisé et d'autres ressources et fonctions.

Les données sensibles en matière de sécurité et les fonctions de sécurité contenues dans les nœuds M2M sur le terrain doivent être protégées contre tout accès non autorisé ou toute altération, comme déterminé par l'analyse des risques. Les données et fonctions sensibles comprennent notamment les justificatifs d'identité utilisés pour la sécurité et les algorithmes les utilisant. L'objectif d'un environnement sécurisé est de fournir le niveau de protection requis (voir le Tableau 6.3.1-1) aux données sensibles pendant leur stockage et leur utilisation, notamment et principalement les secrets cryptographiques symétriques ou asymétriques à longue durée de validité utilisés pendant le fonctionnement. En outre, l'isolement des données et fonctions sensibles utilisés pour la sécurité contrôlées par différentes parties prenantes au sein d'un nœud M2M peut être assuré par des environnements sécurisés distincts. Cet aspect est particulièrement essentiel pour les nœuds M2M auxquels des attaquants potentiels pourraient accéder physiquement ou à distance.

Le choix d'un environnement sécurisé est guidé par une analyse des risques prenant en compte toutes les couches d'une application M2M, même s'il doit s'appuyer, dans la mesure du possible, sur les capacités fournies par la couche service M2M ou le réseau sous-jacent, par exemple la carte à circuit intégré universelle (UICC) dans les réseaux 3GPP et 3GPP2, ou sur les exigences relatives aux environnements d'exécution fiables.

Aucune hypothèse n'est faite sur la mise en œuvre particulière de l'environnement sécurisé. Celui-ci peut être mis en œuvre sous la forme d'un élément de sécurité matériel indépendant, ou d'une fonction logicielle intégrée. Chaque environnement sécurisé peut être associé à un certain niveau de sécurité selon la façon dont il est mis en œuvre. Les différents environnements sécurisés offrent différents niveaux de sécurité et de protection, comme indiqué dans le Tableau 6.3.1-1.

Tableau 6.3.1-1 – Classification des niveaux de protection

Niveau de protection	Description
0	Aucune protection. Les données sont exposées, même sans attaque active.

Tableau 6.3.1-1 – Classification des niveaux de protection

Niveau de protection	Description
1	Protection faible. Les données sont protégées contre les observateurs passifs mais peuvent être exposées à des attaques actives menées localement ou à distance. Par exemple, il existe des solutions logicielles qui reposent sur le matériel de traitement général de l'équipement de support.
2	Protection moyenne. La protection des données contre les attaques à distance est prise en compte, mais les attaques locales, en particulier les attaques physiques, restent possibles. En d'autres termes, une protection moyenne fournit des contre-mesures uniquement contre les attaques logicielles. Par exemple, les solutions logicielles destinées à protéger les données et les fonctions sensibles reposent sur un traitement spécifique assurant une isolation renforcée et permettant de maintenir le code et les données sensibles à l'écart d'un environnement d'exploitation, de logiciels et de mémoires non protégés. Le code exécuté dans l'environnement protégé est vérifié par cryptographie pour garantir son intégrité.
3	Protection élevée. La protection prend en compte les tentatives locales et à distance d'accéder aux données, y compris les attaques impliquant un accès physique. Ce niveau comprend des contre-mesures fortes contre les attaques logicielles et matérielles, par exemple la détection de conditions de fonctionnement anormales, le brouillage et le masquage matériel de la mémoire et l'analyse des opérations impliquant des données sensibles sur des canaux latéraux.

Il convient de prévoir au moins un environnement sécurisé dans chaque nœud M2M fournissant un stockage sécurisé aux CSE et AE locales, mais il peut y en avoir plusieurs.

6.3.2 Module d'extension SE

Le module d'extension SE permet l'accès physique à l'environnement sécurisé considéré. Ce module peut être mis en œuvre de différentes manières selon le type d'environnement sécurisé.

NOTE – La spécification du module d'extension SE ne relève pas du champ d'application de la présente Recommandation.

6.3.3 Abstraction de l'environnement sécurisé

Ce sujet n'est pas abordé dans la présente Recommandation.

7 Autorisation

7.1 Mécanisme de contrôle d'accès

7.1.1 Description générale

La procédure d'autorisation M2M commande l'accès aux ressources et aux services hébergés par les CSE et les AE. Cette procédure impose que l'expéditeur du message de demande d'accès aux ressources ait été identifié par la fonction d'authentification, et que l'expéditeur et le destinataire soient mutuellement authentifiés l'un par rapport à l'autre.

La ressource sollicitée dans un message de demande possède un attribut *accessControlPolicyIDs* associé (lequel est inclus explicitement comme attribut de la ressource sollicitée dans le message de demande, impliqué par le parent de la ressource, ou fixé par le système; voir le § 9.6.1 de la Recommandation [UIT-T Y.4500.1]). L'attribut *accessControlPolicyIDs* contient une liste d'identificateurs des ressources <*accessControlPolicy*> applicables à la ressource sollicitée.

La structure générale des ressources <*accessControlPolicy*> est décrite au § 9.6.2 de la Recommandation [UIT-T Y.4500.1].

Chacune de ces ressources <*accessControlPolicy*> contient des attributs *privileges* et *selfPrivileges*, lesquels comprennent les informations, appelées règles de contrôle d'accès (*access control rules*) dans la présente Recommandation, qui sont évaluées par rapport aux paramètres associés au message de demande pour obtenir la décision d'accès.

La Figure 7.1.1-1 illustre la relation entre les instances des ressources <*accessControlPolicy*> (ACP) et les instances des ressources protégées, notées Ressource_1 à Ressource_N.

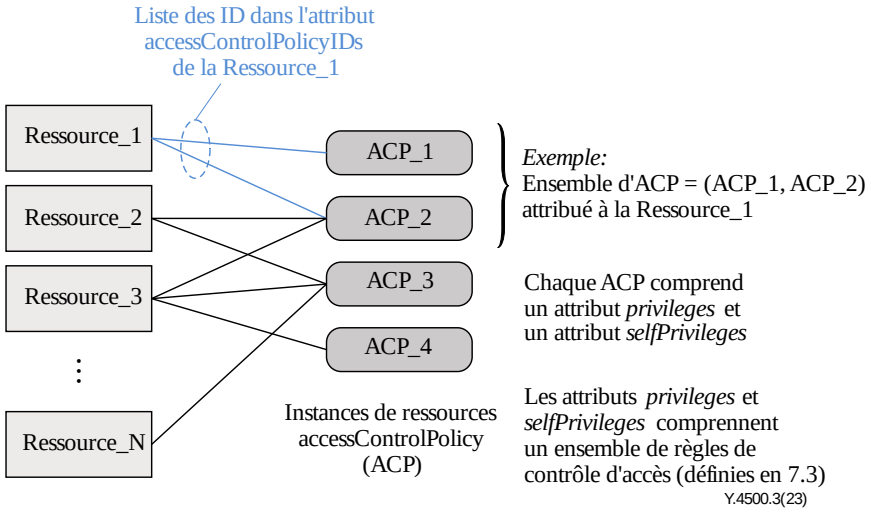


Figure 7.1.1-1 – Relation entre les instances de ressources et les politiques de contrôle d'accès

Les demandes d'accès à une ACP elle-même sont évaluées par rapport à l'attribut *selfPrivileges* de cette ACP. Les demandes d'accès aux instances de tous les autres types de ressources sont évaluées par rapport aux attributs *privileges* de l'ensemble d'ACP associées à la ressource ciblée.

Les demandes d'accès au type de ressource <*accessControlPolicy*> obtiennent une autorisation si le résultat de l'évaluation de la demande est "Permit" (autoriser) pour au moins un attribut *selfPrivileges*. Pour les autres types de ressources, l'autorisation d'accès est accordée si le résultat de l'évaluation de la demande est "Permit" pour au moins un attribut *privileges*.

Les attributs *privileges* et *selfPrivileges* définis dans la ressource *accessControlPolicy* déterminent quel expéditeur est autorisé à accéder à la ressource contenant l'attribut considéré, pour quelle opération spécifique (par exemple création, récupération, mise à jour, suppression, etc.) et pour quelles contraintes de contexte spécifiques (portant sur l'heure d'accès, l'adresse IP de l'expéditeur et localisation géographique de l'expéditeur).

Une telle approche du contrôle d'accès est conforme au concept du contrôle d'accès fondé sur des attributs (ABAC, *attribute based access control*) défini dans le document [b-NIST 800-162].

Les politiques définies dans les ressources <*accessControlPolicy*> sont appliquées par un mécanisme de contrôle d'accès qui utilise l'architecture logique d'autorisation décrite au § 6.2.2.

Le mécanisme de contrôle d'accès rassemble les informations nécessaires pour prendre la décision d'accès:

– Les informations contenues dans le message de demande d'accès aux ressources, définies au § 7.1.2 (Tableau 7.1.2-1).
– Les informations contextuelles, ainsi que définies au § 7.1.2 (Tableau 7.1.2-2).
– Les jetons (le cas échéant) associés à la demande d'accès aux ressources.

– Les politiques régissant l'accès, définies au § 7.1.3.

7.1.2 Paramètres du message de demande

Ce paragraphe définit les paramètres du message de demande qui sont évalués par le mécanisme de contrôle d'accès.

Les types de données applicables à ces paramètres sont définis au § 6.4 de la Recommandation [UIT-T Y.4500.4].

Le Tableau 7.1.2-1 ci-après recense les différents paramètres.

Tableau 7.1.2-1 – Paramètres indiqués dans le message de demande

Paramètre	Description	Obligatoire/ Facultatif	Utilité dans le mécanisme de contrôle d'accès
To	URI de la ressource cible	O	Sélection de la politique de contrôle d'accès (*accessControlPolicy*) associée à la ressource cible
From	Identificateur représentant l'expéditeur de la demande	O (Note 1)	Évalué par rapport à accessControlOriginators dans les attributs *privileges* et *selfPrivileges*
Role IDs	Identificateur de rôle de l'expéditeur	F	Évalué par rapport à accessControlOriginators dans les attributs *privileges* et *selfPrivileges*
Operation	Opération demandée	O	Évalué par rapport à accessControlOperations dans les attributs *privileges* et *selfPrivileges*
Resource type	Type de la ressource cible	F (Note 2)	Évalué par rapport à accessControlObjectDetails dans les attributs *privileges.* S'applique uniquement pour les opérations de type Create.
Filter criteria	Étiquette conditionnelle ***filterUsage*** dans les critères de filtrage	F	Différenciation entre les opérations de récupération et de découverte.
Tokens	Jetons protégés par un cadre ESData	F	Contient les informations d'autorisation (par exemple des identificateurs de rôle) à utiliser lors de la prise de décision par rapport à la demande.
Token IDs	tokenID **ou** Local-Token-ID	F	Identifie les jetons contenant les informations d'autorisation (par exemple les identificateurs de rôle) à utiliser lors de la prise de décision par rapport à la demande.
NOTE 1 – Le paramètre de primitive ***From*** est obligatoire dans toutes les demandes; il n'est facultatif que dans la procédure d'enregistrement des AE, comme décrit dans la Recommandation [UIT-T Y. 4500.1]. NOTE 2 – Le paramètre de primitive ***resource Type*** est présent uniquement dans les primitives des demandes Create.			

Le Tableau 7.1.2-2 présente la liste des paramètres contextuels associés à un message de demande qui sont évalués par le mécanisme de contrôle d'accès. Ces paramètres ne sont pas explicitement inclus dans un message de demande, mais peuvent être obtenus au niveau du destinataire et validés par rapport aux paramètres contextuels des politiques, comme indiqué dans le Tableau 7.1.2-2.

Tableau 7.1.2-2 – Paramètres contextuels associés à un message de demande

Paramètre	Description	Utilité dans le mécanisme de contrôle d'accès
rq_time	Horodatage de réception du message de demande par la CSE hôte. Généré par l'horloge système de la CSE hôte.	Validé par rapport au paramètre *accessControlTimeWindow* d'une règle de contrôle d'accès. Voir le § 7.1.3.
rq_loc	Emplacement géographique de l'expéditeur de la demande. Obtenu via le point de référence Mcn.	Validé par rapport au paramètre *accessControlLocationRegion* d'une règle de contrôle d'accès. Voir le § 7.1.3.
rq_ip	Adresse IP source associée aux paquets IP qui transportent le message de demande. Obtenue via le point de référence Mcn.	Validé par rapport au paramètre *accessControlIPAddress* d'une règle de contrôle d'accès. Voir le § 7.1.3.

Des jetons, conformes à la définition donnée au § 7.3.3.1 "Structure des jetons", peuvent être associés à un message de demande. Un jeton peut se retrouver associé à un message de demande s'il a été inclus dans le paramètre de primitive ***Tokens*** du message en question, ou s'il a été identifié dans le paramètre de primitive ***Token IDs*** de ce même message. Si la CSE hôte a obtenu un jeton du serveur du système d'autorisation dynamique (DAS) par une procédure d'autorisation dynamique directe, ce jeton sera associé à une demande si le paramètre du détenteur dans le jeton correspond à l'AE-ID ou au CSE-ID absolu de l'expéditeur de la demande. L'autorisation dynamique est décrite au § 7.3.

Le Tableau 7.1.2-3 recense les paramètres de sécurité contextuels associés à un message de demande.

Tableau 7.1.2-3 – Paramètres de sécurité contextuels associés à un message de demande

Paramètre	Description	Obligatoire/ Facultatif	Utilité dans le mécanisme de contrôle d'accès
rq_authn	Valeur booléenne (TRUE/FALSE) indiquant si l'expéditeur est considéré comme ayant été authentifié par la CSE hôte, et si le paramètre From correspondait bien à l'identité authentifiée de l'expéditeur.	O	Validé par rapport au paramètre accessControlAuthentificationFlag d'une règle de contrôle d'accès. Voir le § 7.1.3.

Les critères suivants doivent être appliqués pour déterminer si un expéditeur est considéré comme ayant été authentifié par la CSE hôte.

Si l'expéditeur est une AE enregistrée auprès de la CSE hôte, les critères permettant de décider si cet expéditeur est authentifié sont spécifiques au déploiement et/ou à la mise en œuvre et dépendent du niveau de confiance garanti par le mode de réalisation physique et logique du dispositif qui supporte les AE et la CSE hôte (par exemple, le démarrage sécurisé et la résistance aux intrusions). Dans de nombreux cas, il convient de s'attendre à ce qu'un canal sécurisé impliquant une authentification (par exemple une session TLS ou DTLS) soit utilisé pour protéger les primitives sur l'interface MCA, auquel cas l'authentification est considérée comme valide pendant la durée de la session TLS. Lorsque ce n'est pas le cas, par exemple parce que la conception physique et logique est fiable, l'authentification peut être considérée comme valide en permanence, sauf s'il est détecté que le dispositif est compromis.

Si l'expéditeur est une CSE enregistrée auprès de la CSE hôte, il est considéré comme authentifié pendant la durée d'une session (D)TLS car le Mcc doit toujours être protégé par TLS ou DTLS en application d'un cadre d'établissement d'association de sécurité (SAEF), comme décrit au § 8.2. L'autre CSE peut être l'entité d'enregistrement ou l'entité enregistrée vis-à-vis de la CSE hôte.

Si l'expéditeur est une AE ou une CSE enregistrée auprès d'une CSE autre que la CSE hôte, il est considéré comme authentifié par la CSE hôte si et seulement si la primitive de demande est protégée par un cadre de sécurité des primitives de bout en bout (ESPrim), comme décrit au § 8.4.

7.1.3 Format des attributs privileges et selfPrivileges

Les attributs *privileges* et *selfPrivileges* présentent le même format de type de données qu'indiqué ci-après.

Chaque attribut *privileges* ou *selfPrivileges* comprend un ensemble de règles de contrôle d'accès. Dans ce qui suit, l'ensemble de règles de contrôle d'accès est noté *acrs*, chaque règle de contrôle d'accès individuelle étant notée *acr*. Les règles de contrôle d'accès dans l'*acrs* sont indexées avec la lettre *k*. Le nombre de règles de contrôle d'accès dans l'ensemble est noté K:

acrs = { *acr*(1), *acr*(2), ..., *acr*(*k*), ..., *acr*(K) }

Chaque règle de contrôle d'accès *acr*(*k*) est composée de trois types de composants, appelés accessControlOriginators, accessControlOperations et accessControlContexts. Le composant accessControlContext est un paramètre facultatif.

Une règle de contrôle d'accès *acr*(*k*) sera donc représentée comme un doublet:

acr(*k*) = {*acr*(*k*)_accessControlOriginators, *acr*(*k*)_accessControlOperations}

ou un triplet:

acr(*k*) = {*acr*(*k*)_accessControlOriginators, *acr*(*k*)_accessControlOperations, *acr*(*k*)_accessControlContexts}

Le terme générique "multiplet de règles de contrôle d'accès" est employé pour désigner une règle *acr*(*k*).

Un ensemble *acrs* de règles de contrôle d'accès peut comprendre un mélange de doublets et de triplets. Pour les doublets, tous les paramètres de contexte associés à un message de demande sont admissibles.

Les paramètres des trois composants d'un multiplet de règles de contrôle d'accès pris en charge dans la présente Recommandation sont détaillés dans le Tableau 7.1.3-1.

Tableau 7.1.3-1 – Paramètres d'un multiplet de règles de contrôle d'accès

Paramètre	Utilisation	Obligatoire/ Facultatif	Format
accessControlOriginators	Liste des expéditeurs qui peuvent être autorisés	O	Liste de CSE-ID et/ou d'AE-ID, ou mot-clé "all" qui octroie l'accès à tous les expéditeurs
accessControlOperations	Liste des opérations qui peuvent être autorisées	O	Énumération d'opérations parmi Create Retrieve, Update, Delete, Discover, Notify
accessControlContexts	Voir le Tableau 7.1.3-3	F	Voir le Tableau 7.1.3-3
accessControlObjectDetails	Voir le Tableau 7.1.3-4	F	Voir le Tableau 7.1.3-4

Tableau 7.1.3-1 – Paramètres d'un multiplet de règles de contrôle d'accès

Paramètre	Utilisation	Obligatoire/ Facultatif	Format
accessControlAuthentication Flag	Indique si la règle s'applique uniquement aux expéditeurs considérés comme authentifiés par la CSE hôte	F	Booléen

Le paramètre accessControlOriginators comprend une liste de noms de domaines SP, CSE-ID, AE-ID, identificateurs de ressources des ressources <group> et/ou identificateurs de rôles, sous tous les formats définis dans la Recommandation [UIT-T Y.4500.1]. Si l'on doit autoriser l'accès à tous les expéditeurs, le mot-clé réservé "all" peut être inclus dans l'espace de valeur du paramètre accessControlOriginators.

L'utilisation d'un nom de domaine de fournisseur de services dans le paramètre accessControlOriginators signifie que tous les AE-ID et CSE-ID correspondant au nom de domaine donné peuvent être autorisés.

Il est également permis d'utiliser le caractère de troncation "*", dans les représentations des CSE-ID et AE-ID. Le champ d'action de "*" est terminé par une barre oblique "/" placée à sa suite. Le Tableau 7.1.3-2 présente des exemples d'utilisation de caractères de troncation dans les CSE-ID et les AE-ID.

Les caractères de troncation ne peuvent pas être employés avec les noms de domaine des fournisseurs de services, les identificateurs des ressources <*group*> et les identificateurs de rôle.

Tableau 7.1.3-2 – Exemples d'utilisation des caractères de troncation dans les CSE-ID et AE-ID de accessControlOriginators

	Forme de l'ID	Exemples	Signification
CSE-ID	Absolue	//m2msp.org/myCSEID //*/myCSEID //*/myCSE*	Toute CSE dont l'ID correspond aux caractères de troncation
	Relative au SP	//*/myCSEID //*/myCSE*	Toute CSE correspondante du fournisseur de services qui héberge la ressource cible
AE-ID	Absolue	//m2msp.org/S988 //*/myCSEID/C9886 //*/myCSE*/C9886	Toute AE dont l'ID correspond aux caractères de troncation
	Relative au SP	/myCSEID/C9886 /myCSEID/C98* /myCSE*/C98* /SmyAE*	Toute AE correspondante du fournisseur de services qui héberge la ressource cible

Le type de données applicable au paramètre accessControlOriginators est défini dans la Recommandation [UIT-T Y.4500.4].

Le paramètre accessControlOperations comprend une liste d'opérations admissibles qui peut être un sous-ensemble quelconque des éléments suivants: Create, Retrieve, Update, Delete, Discover et Notify. Au contraire des opérations Create, Retrieve, Update, Delete et Notify, indiquées explicitement dans le paramètre *op* d'un message de demande, l'opération Discovery est indiquée par ***op*** = Retrieve combiné à la spécification des paramètres ***fc*** et *Disrestype* dans le message de demande.

Le type de données applicable au paramètre accessControlOperations est défini dans la Recommandation [UIT-T Y.4500.4].

Le Tableau 7.1.3-3 ci-après énumère les paramètres de accessControlContexts.

Tableau 7.1.3-3 – Paramètre de accessControlContexts

Paramètre	Utilisation	Obligatoire/ Facultatif	Format
accessControlTimeWindow	Liste des fenêtres temporelles qui peuvent être autorisées	F	Liste des intervalles de temps pendant lesquels l'accès peut être octroyé, au format crontab étendu
accessControlLocationRegion	Liste des emplacements géographiques qui peuvent être autorisées	F	1) Coordonnées en latitude/longitude, associées à un rayon définissant une région circulaire autour du point indiqué par les coordonnées 2) Code de pays
accessControlIpAddress	Liste des adresses IPv4 et IPv6 qui peuvent être autorisées	F	IPv4: notation décimale séparée par des points, avec suffixe CIDR IPv6: groupes de chiffres hexadécimaux séparés par deux points, avec suffixe CIDR

Le paramètre accessControlTimeWindow représente une liste d'éléments qui respectent la syntaxe crontab étendue décrite au § 7.3.8 de la Recommandation [UIT-T Y.4500.4]. Il permet de définir des intervalles de temps récurrents, pendant lesquels l'accès peut être autorisé lorsque le paramètre ***rq_time*** associé au message de demande d'accès tombe pendant l'un de ces intervalles.

Les éléments du paramètre accessControlLocationRegion peuvent être représentés de deux façons: par un code de pays sur deux lettres, ou sous la forme d'un cercle de rayon *R* centré sur un point défini par une longitude et une latitude. Consulter l'Annexe F pour des informations détaillées. Chaque élément de accessControlLocationRegion définit un emplacement géographique admissible, qui est comparée au paramètre ***rq_loc*** associé au message de demande d'accès.

Les types de données applicables au paramètre accessControlLocationRegion et ***rq_loc*** sont définis dans la Recommandation [UIT-T Y.4500.4].

Le paramètre accessControlIpAddress représente une liste d'adresses IP, notées respectivement sous forme décimale séparée par des points avec un suffixe CIDR pour IPv4, et sous forme de groupes de chiffres hexadécimaux séparés par des deux-points avec un suffixe CIDR pour IPv6. Si le paramètre ***rq_loc*** associé au message de demande d'accès correspond à l'une de ces adresses, l'accès peut être autorisé sur la base de ce critère.

Les types de données applicables aux paramètres accessControlIpAddress et ***rq_ip*** sont définis dans la Recommandation [UIT-T Y.4500.4].

Le paramètre accessControlAuthenticationFlag est une valeur booléenne. Si ce paramètre est absent, on lui attribue la valeur FALSE. Si sa valeur est TRUE, la règle de contrôle d'accès s'applique uniquement aux expéditeurs considérés comme déjà authentifiés par la CSE hôte. Le § 7.1.2 décrit les critères utilisés pour décider si l'expéditeur est considéré comme déjà authentifié par la CSE hôte.

Le Tableau 7.1.3-4 ci-après énumère les paramètres de accessControlObjectDetails.

Tableau 7.1.3-4 – Paramètres de accessControlObjectDetails

Paramètre	Utilisation	Obligatoire/Facultatif	Format
resourceType	Type de ressource auquel s'applique la règle de contrôle d'accès	F	Identificateur de type de ressource
specializationID	Identificateur de mgmtDefinition ou de containerDefinition	F	mgmtDefinition ou containerDefinition, représentés sous forme de chaîne de caractères
childResourceType	Ensemble d'identificateurs de type de ressources qui peuvent être créées sous la ressource parent	F	Liste de types de ressource

L'attribut accessControlObjectDetails décrit un sous-ensemble de types de ressources enfants de la ressource cible auxquels s'applique la règle de contrôle d'accès. Si une règle de contrôle d'accès comprend un attribut *accessControlObjectDetails*, un type *childResourceType* est indiqué. Une règle de contrôle d'accès qui ne comprend aucun paramètre *accessControlObjectDetails* s'applique à tous les types de ressources enfants de la ressource cible. Le paramètre *accessControlObjectDetails* est décrit dans le Tableau 9.6.2.4-1 de la Recommandation [UIT-T Y.4500.1]. Les types de ressources enfants énumérés dans l'élément *childResourceType* sont soumis à un contrôle d'accès uniquement pour l'opération Create. Quand une ressource enfant a été créée, les politiques de contrôle d'accès qui lui sont attribuées directement s'appliquent. Les éléments *resourceType* et *specializationID* sont facultatifs. Si l'un ou l'autre des éléments *resourceType* et *specializationID* est présent dans *accessControlObjectDetails*, la CSE compare le type de ressource ou la spécialisation de la ressource cible avec la valeur indiquée dans l'élément *resourceType* or *specializationID*. Une vérification plus poussée de *childResourceType* n'intervient que si *resourceType* ou *specializationID* correspond. Si aucun des éléments *resourceType* et *specializationI* n'est fourni, la CSE vérifie uniquement la correspondance pour *childResourceType*.

7.1.4 Décision de contrôle d'accès

La décision d'accès est prise en comparant les paramètres associés à un message de demande d'accès à une ressource, décrit au § 7.1.2, aux règles de contrôle d'accès figurant dans les attributs *privileges* ou *selfPrivileges* de tous les ensembles d'ACP attribués à la ressource protégée par les *accessControlPolicyIDs* (voir la Figure 7.1.1-1).

Le résultat de l'algorithme de décision d'accès, c'est-à-dire la décision d'accès, est le résultat global de l'évaluation de l'ensemble applicable de règles de contrôle d'accès, *acrs*, par rapport aux paramètres associés au message de demande d'accès. Cette décision peut être représentée par une valeur de type binaire. Le résultat global de l'algorithme de décision d'accès est désigné ici par le nom de variable *res_acrs*:

ou

$$res\_acrs = \begin{cases} \text{TRUE ou } 1 & \text{si la demande correspond aux règles de contrôle d'accès} \\ \text{FALSE ou } 0 & \text{sinon} \end{cases}$$

L'algorithme de décision d'accès est décrit au § 7.1.5. Pour tout ensemble donné d'entrées, une mise en œuvre du traitement de décision d'accès doit renvoyer le même résultat que l'algorithme de décision d'accès de référence renverrait pour ces entrées.

Si l'algorithme de décision d'accès renvoie le résultat *res_acrs* = TRUE, la décision d'accès à la ressource demandée doit être "Permit" (autoriser).

Si l'algorithme de décision d'accès renvoie le résultat *res_acrs* = FALSE, ou qu'il est incapable de produire un résultat final (en raison, par exemple, de paramètres indéterminés), la décision d'accès à la ressource demandée doit être "Deny" (refuser).

7.1.5 Description de l'algorithme de décision d'accès

L'algorithme de décision d'accès de référence décrit dans ce paragraphe combine les résultats partiels de contrôle d'accès obtenus pour chacune des règles de contrôle d'accès individuelles que contient un attribut *privileges* ou *selfPrivileges*. En outre, si plusieurs instances d'ACP sont attribuées à la ressource protégée, l'algorithme de décision d'accès de référence combine les résultats de contrôle d'accès partiel obtenus pour les ACP individuelles d'un ensemble d'ACP.

L'algorithme décrit dans ce paragraphe adopte un algorithme de combinaison "Permit-overrides" (l'autorisation s'impose) vis-à-vis des règles de contrôle d'accès et des ACP, comme défini dans le standard XACML [b-OASIS XACML]. Cet algorithme se comporte de la façon suivante:

Si une décision est "Permit" pour une seule des règles de contrôle d'accès de l'attribut *privileges* (ou *selfPrivileges*) d'une seule ACP, le résultat est "Permit".

Dans tous les autres cas, le résultat est "Deny".

La logique d'évaluation d'une demande par rapport à un privilège peut être décrite mathématiquement comme suit. Un attribut *privileges* ou *selfPrivileges* inclus dans une ressource <*accessControlPolicy*> représente un ensemble de règles de contrôle d'accès, *acrs*, construit comme illustré à la Figure 7.1.5-1.

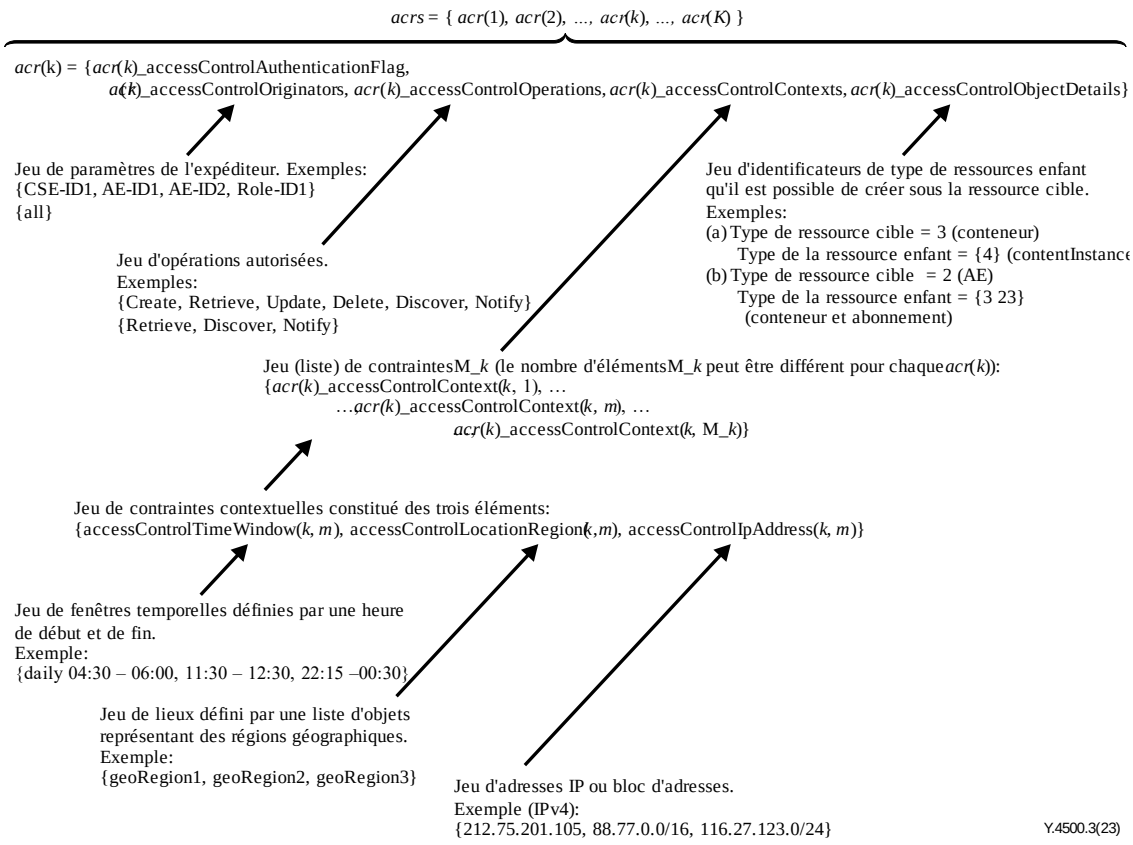


Figure 7.1.5-1 – Logique d'évaluation des privilèges dans l'algorithme de décision d'accès de référence

Les paramètres associés à une demande, qui sont évalués par rapport aux paramètres contenus dans les règles de contrôle d'accès, sont décrits au § 7.1.3.

La décision d'accès *res_acrs* définie au § 7.1.4 est obtenue en évaluant si les paramètres associés au message de demande énumérés aux tableaux 7.1.2-1 et 7.1.2-2 correspondent à l'une des règles de contrôle d'accès que contient l'ensemble des règles de contrôle d'accès défini au § 7.1.3, comme suit:

res_acrs = *res_acr*(1) OR *res_acr*(2) ... OR *res_acr*(k) … OR *res_acr*(K),

où *res_acr*(*k*) représente le résultat de l'évaluation logique (c'est-à-dire TRUE/FALSE ou 1/0) des paramètres de la demande par rapport à la $k^{ième}$ règle de contrôle d'accès de l'ensemble de règles *acrs*, résultat qui peut être exprimé de la façon suivante:

res_acr(*k*) = *res_authn(k)* AND *res_origs*(*k*) AND *res_ops*(*k*) AND *res_ctxts*(*k*) AND *res_objd*(*k*), *k* = 1…K.

La première variable du résultat logique partiel *res_authn(k)* du côté droit de l'équation ci-dessus doit être évaluée suivant le Tableau 7.1.5-1:

Tableau 7.1.5-1 – Évaluation de *res_authn(k)*

***acr(k)*_accessControlAuthenticationFlag**	***rq_authn***	***res_authn***
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	TRUE
FALSE	FALSE	TRUE

Les quatre variables de résultat logique partiel restantes du côté droit de l'équation ci-dessus peuvent être définies au moyen de la fonction d'ensemble suivante:

$$\text{ismember}(x, \text{setX}) = \begin{cases} \text{TRUE ou } 1 & \text{si } x \in \text{setX} \\ \text{FALSE ou } 0 & \text{sinon} \end{cases}$$

Avec cette définition:

res_origs(*k*) = ismember(***Originator***, *acr*(*k*)_accessControlOriginators)

res_ops(*k*) = ismember(***Operation***, acr(*k*)_ accessControlOperations)

Dans l'équation ci-dessus, la variable ***Originator*** désigne l'identité authentifiée de l'expéditeur de la primitive de demande qui correspond au paramètre ***From***.

Le troisième résultat logique partiel *res_ctxts*(*k*) est obtenu de la façon suivante:

res_ctxts(*k*) = *res_context*(*k*, 1) ... OR *res_context*(*k*, *m*) ... OR *res_context*(*k*, M_*k*),

où:

res_context(*k*, *m*) = *res_time*(*k*, *m*) AND *res_ip*(*k*, *m*) AND *res_loc* (*k*, *m*), k = 1…K, *m* = 1…M_*k*

et

res_time(*k*, *m*) = ismember(***rq_time***, *acr*(*k*)_accessControlTimeWindow(*m*))

res_ip(*k*, *m*) = ismember(***rq_ip***, *acr*(*k*)_accessControlIpAddress(*m*))

res_loc (*k*, *m*) = ismember(***rq_loc***, *acr*(*k*)_accessControlLocationRegion(*m*))

Le quatrième résultat logique partiel *res_objd*(*k*) s'applique uniquement aux primitives des demandes Create et peut être obtenu de la façon suivante:

res_ objd(*k*) = *res_ objdetails*(*k*, 1) ... OR *res_ objdetails*(*k*, *m*) ... OR *res_ objdetails*(*k*, M_*k*),

où:

res_ objdetails(*k*, *m*) = *res_resourceType*(*k*, *m*) AND *res_specializationID*(*k*, *m*) AND *res_childResource*(*k*,*m*),

pour *m* = 1…M_*k***.** Les trois arguments logiques sont définis ci-après.

Pour chaque élément donné les arguments *acr*(*k*)_accessControlObjectDetails(*m*) dans une règle de contrôle d'accès déterminent si le paramètre facultatif *resourceType* est présent.

resourceType = *acr*(*k*)_accessControlObjectDetails(*m*)/resourceType

Selon que *resourceType* est présent ou non, l'argument *res_resourceType*(*k*, *m*) sera défini comme

$$res\_resourceType(k,m) = \begin{cases} \text{TRUE ou } 1, & \text{si absent de } acr(k)\_\text{accessControlObjectDetails}(m) \\ \text{TRUE ou } 1, & \text{si présent et } resourceType = targetResourceTypeID \\ \text{FALSE ou } 0, & \text{si présent et } resourceType \neq targetResourceTypeID \end{cases}$$

où *targetResourceTypeID* est l'identificateur de type de ressource associé à la ressource désignée dans le paramètre ***To*** de la primitive de demande Create.

Si la valeur de l'élément *resourceType* est 13 (spécialisation <mgmtObject>) ou 28 (spécialisation <flexContainer>), l'élément facultatif *specializationID* doit également figurer dans accessControlObjectDetails:

specializationID = *acr*(*k*)_accessControlObjectDetails(*m*)/specializationID

Si *specializationID* est présent, il doit correspondre aux attributs *mgmtDefinition* ou *containerDefinition* indiqués dans le paramètre ***Content*** de la primitive de demande Create.

$$res\_specializationID(k.m) = \begin{cases} \text{TRUE ou } 1, & \text{si } specializationID \text{ absent de } acr(k)\_accessControlObjectDetails(m) \\ \text{TRUE ou } 1, & specializationID = mgmtDefinition\,(resourceType = 13) \\ \text{TRUE ou } 1, & specializationID = containerDefinition\,(resourceType = 28) \\ \text{FALSE ou } 0, & specializationID \neq mgmtDefinition\,(resourceType = 13) \\ \text{FALSE ou } 0, & specializationID \neq containerDefinition\,(resourceType = 28) \end{cases}$$

L'élément *childResourceType* est obligatoire dans tous les éléments accessControlObjectDetails d'une règle de contrôle d'accès. Il comprend une liste des $j = 1 \ldots J$ identificateurs de type de ressource enfant auquel s'applique la règle. Le $j^{ième}$ élément de la liste est noté comme suit:

childResourceType(*k, m. j*) = *acr*(*k*)_accessControlObjectDetails(*m*)/childResourceType(*j*), *j* = 1…J

La variable logique *res_childResource*(*k, m*) est obtenue comme étant

res_ childResource (*k, m*) = ismember(***Resource Type***, *childResourceType*(*k, m, j*))

où ***Resource Type*** désigne la valeur des paramètres de la primitive de demande Create.

Si *resourceType* et *specializationID* sont absents de *acr*(*k*)_accessControlObjectDetails(*m*), *res_ objdetails*(*k, m*) = *res_resourceType*(*k, m*) AND *res_specializationID*(*k, m*) AND *res_childResource*(*k,m*) = *res_childResource*(*k,m*)

Grâce à l'approche combinée "Permit-overrides" (l'autorisation s'impose), si la décision de contrôle d'accès pour une règle de contrôle d'accès de *res_acr* a la valeur TRUE, l'algorithme de décision d'accès de référence peut s'arrêter sans évaluer les autres règles de contrôle d'accès éventuellement applicables de l'ACP actuelle ou de toute autre ACP de l'ensemble d'ACP, et la décision d'accès finale est "Permit" (autoriser).

7.2 Prévention de l'usurpation d'identité d'une AE

7.2.1 Vérification de l'AE-ID par une entité d'enregistrement

Étant donné que certaines AE peuvent se comporter de manière malveillante et se faire passer pour un autre AE dont l'ID a été modifié, la CSE hôte a besoin d'un mécanisme de prévention de l'usurpation d'identité des AE. Ce mécanisme est mis en œuvre au niveau de la CSE d'enregistrement, puisque celle-ci est un point d'entrée du système M2M.

Lorsque la CSE d'enregistrement reçoit une demande, elle doit exécuter la procédure illustrée ci-dessous à la Figure 7.2.1-1.

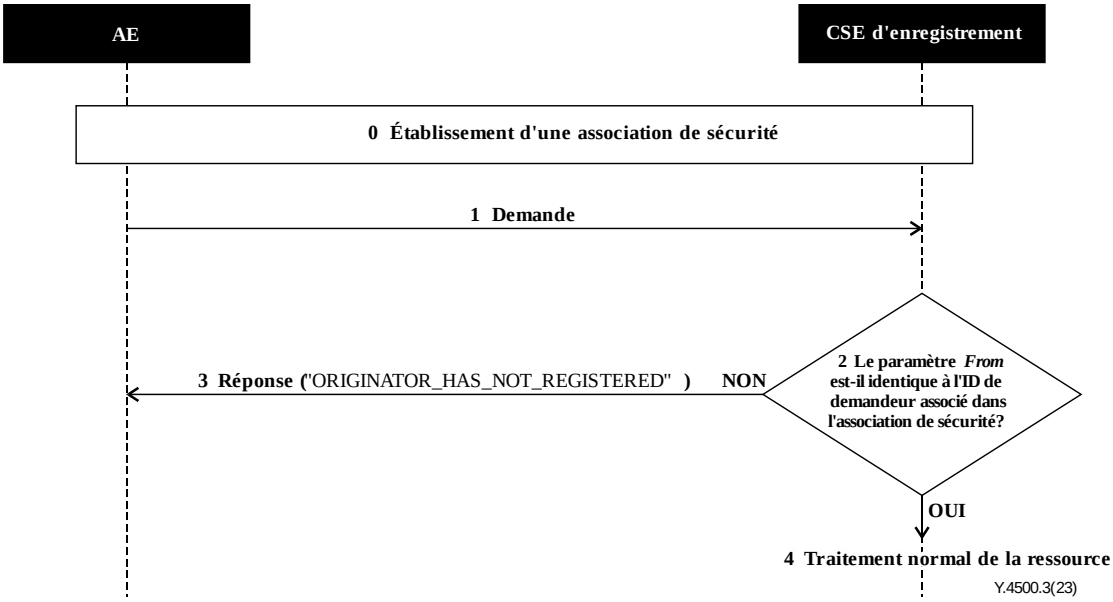


Figure 7.2.1-1 – Procédure de vérification de l'usurpation d'identité d'une AE

0 Il est possible d'effectuer un établissement d'association de sécurité. Le paragraphe 6.1.2.2.1 décrit les scénarios dans lesquels l'établissement d'une association de sécurité entre une AE et une CSE est obligatoire, ainsi que les scénarios dans lesquels l'établissement d'une telle association de sécurité est recommandé. Les procédures ci-après doivent être exécutées si une association de sécurité a été établie.

1 L'AE envoie une demande à la CSE hôte via la CSE d'enregistrement de cette dernière, comme indiqué dans la Recommandation [UIT-T Y.4500.1] (la CSE hôte n'est pas représentée sur cette figure et peut être la CSE d'enregistrement ou une autre CSE).

2 La CSE d'enregistrement vérifie si la valeur dans le paramètre ***From*** est identique à l'ID associé dans l'association de sécurité:

3 Si les valeurs ne sont pas identiques, la CSE d'enregistrement doit renvoyer une réponse portant le code d'état de réponse '4106' ("ORIGINATOR_HAS_NOT_REGISTERED").

4 Si les valeurs sont identiques, la CSE d'enregistrement doit exécuter les procédures décrites au § 8.2 de la Recommandation [UIT-T Y.4500.1]. En fonction du nombre de CSE de transit, la CSE d'enregistrement doit soit traiter la demande, soit la transférer à la CSE hôte ou à une autre CSE de transit.

NOTE – Cette procédure de vérification d'usurpation d'identité n'est pas applicable aux CSE. En effet, lorsqu'une CSE de transit transfère une demande à une autre CSE, le paramètre ***From*** de la demande est l'identificateur de l'expéditeur, qui est différent de l'identificateur de la CSE de transit.

7.2.2 Vérification au moyen de la sécurité des primitives de bout en bout (ESPrim)

Le cadre de sécurité des primitives de bout en bout (ESPrim), décrit au § 8.4, permet à une cible (CSE hôte ou AE) d'authentifier l'expéditeur de primitives de demandes gérées par d'autres CSE. ESPrim assure également la confidentialité et la protection de l'intégrité de ces primitives de demande et de réponse. Les primitives ainsi protégées sont appelées primitives internes. Le chiffrement ESPrim est appliqué aux primitives internes pour former des objets ESPrim. Des primitives externes transportent les objets ESPrim entre l'expéditeur et la CSE ou l'AE cible. L'entité d'enregistrement de l'expéditeur ne peut pas voir la primitive interne chiffrée, et ne peut pas vérifier si le paramètre ***From*** de la primitive interne est correct. C'est donc à la cible qu'il incombe de vérifier que le paramètre ***From*** de la primitive interne correspond à l'identité authentifiée de l'expéditeur.

Lorsque la cible reçoit une demande protégée par ESPrim, elle doit exécuter la procédure décrite ci-dessous dans la Figure 7.2.2-1.

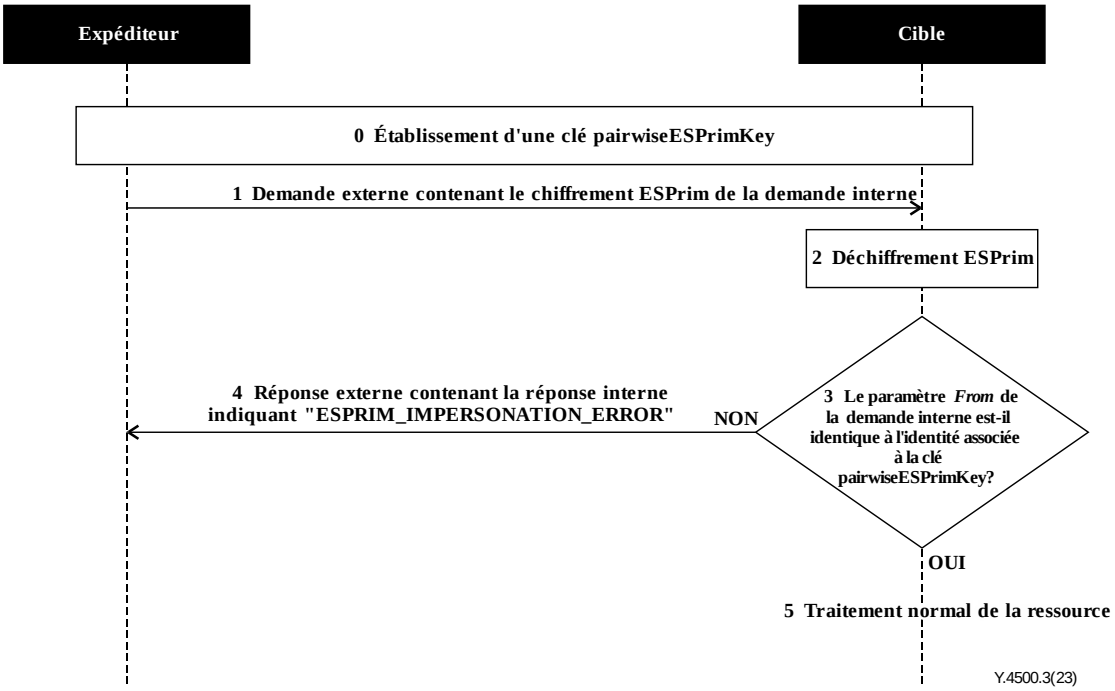


Figure 7.2.2-1 – Procédure de vérification de l'usurpation d'identité d'une AE

0 La cible et l'expéditeur ont établi au préalable une clé pairwiseESPrimKey symétrique. La cible associe une identité à la clé pairwiseESPrimKey symétrique.

1 L'expéditeur compose la primitive de demande interne, la chiffre avec ESPrim pour former un objet ESPrim, puis l'envoie à la cible comme décrit au § 8.4.

NOTE – Qu'un chiffrement ESPrim ait été appliqué ou non, chaque "bond" Mcc toujours protégé au moyen d'un cadre SAEF, et chaque "bond" Mca est éventuellement protégé au moyen d'un SAEF; voir le § 6.1.2.2.1.

2 La cible applique les procédures décrites au § 8.4 pour déchiffrer l'objet ESPrim Object et obtenir la primitive de demande interne.

3 La cible vérifie si la valeur dans le paramètre ***From*** est identique à l'ID associé à la clé pairwiseESPrimKey:

4 Si les valeurs ne sont pas identiques, la cible doit renvoyer une réponse portant le code d'état de réponse 4116 ("ESPRIM_IMPERSONATION_ERROR").

5 Si les valeurs sont identiques, la cible doit noter que l'expéditeur a été authentifié, et exécuter les procédures décrites au § 8.2 de la Recommandation [UIT-T Y.4500.1].

7.3 Autorisation dynamique

7.3.1 Objectif de l'autorisation dynamique

L'autorisation dynamique fournit un cadre interopérable qui permet d'octroyer dynamiquement à un expéditeur des permissions temporaires lui donnant accès à une ou plusieurs ressources ou CSE.

Le document oneM2M TR-0019 [b-oneM2M TR0019] présente des cas d'utilisation, liste les exigences et énonce des propositions relatives à ce cadre.

La présente Recommandation décrit les paramètres d'autorisation dynamique échangés et le traitement associé au niveau de l'expéditeur et de la CSE hôte. Le transport des paramètres d'autorisation dynamique est abordé dans les Recommandations [UIT-T Y.4500.1] et [UIT-T Y.4500.4].

7.3.2 Détails de l'étape 2 de l'autorisation dynamique

7.3.2.1 Modèle de référence de l'autorisation dynamique

La Figure 7.3.2.1-1 présente le modèle de référence de l'autorisation dynamique.

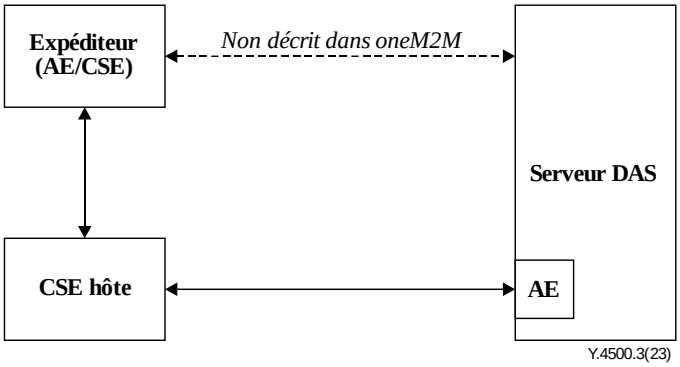


Figure 7.3.2.1-1 – Modèle de référence de l'autorisation dynamique

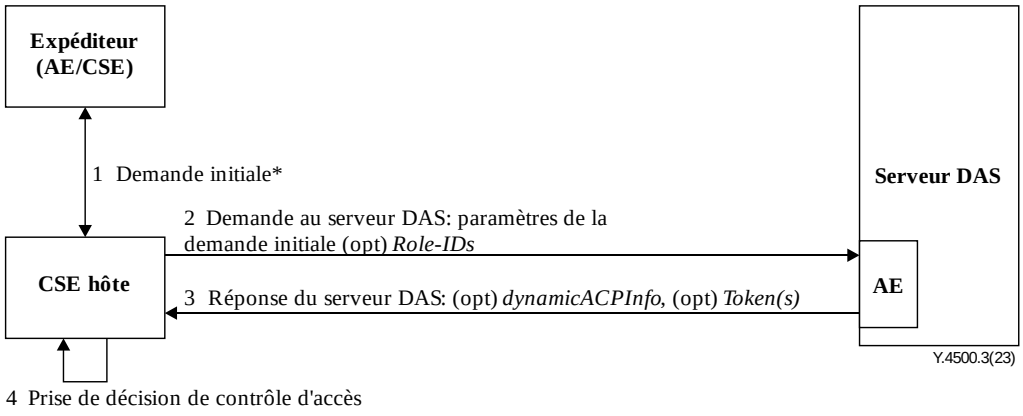
Le modèle de référence de l'autorisation dynamique introduit les systèmes et entités ci-après:

– Système d'autorisation dynamique (DAS, *dynamic authorization system*): système qui prend en charge l'autorisation dynamique pour le compte des propriétaires de ressources. La présente Recommandation ne décrit pas le traitement et le flux de messages au sein du système d'autorisation dynamique. Ce système peut résider en interne ou en externe dans le réseau du fournisseur de services.

– Serveur DAS: serveur configuré avec des politiques d'autorisation dynamique, et possédant les justificatif d'identité nécessaires pour émettre des jetons. Le serveur DAS peut comprendre une AE pour interagir avec le système oneM2M.

Le modèle décrit les procédures d'autorisation dynamique suivantes:

Autorisation dynamique directe décrite à la Figure 7.3.2.1-2. Dans cette procédure, la CSE hôte interagit avec le serveur DAS pour obtenir une autorisation dynamique.



** La demande initiale peut contenir des jetons ou des identificateurs de jetons. Voir les détails applicables dans les autres figures.*

Figure 7.3.2.1-2 – Autorisation dynamique directe

Autorisation dynamique indirecte, décrite à la Figure 7.3.2.1-3:

– Étapes 1-2: la CSE hôte peut fournir à l'expéditeur la ***Token Request Information*** (demande d'informations de jeton) dans la réponse négative.

– Étape 3: l'expéditeur interagit avec le serveur DAS dans l'intention que le serveur DAS émette des jetons (*Tokens*) autorisant l'expéditeur, ce dernier recevant le jeton ou un identificateur de jeton. Cette interaction n'est pas décrite dans la présente Recommandation.

– Étapes 4-7: l'expéditeur fournit un jeton (*Token*) ou un identificateur de jeton (*Token-ID*) à la CSE hôte, pour indiquer que le jeton doit être pris en compte dans la décision d'accès. Dans le cas d'un identificateur de jeton, la CSE hôte récupère le jeton correspondant auprès d'une AE du serveur DAS. Ces informations sont ensuite prises en compte dans la décision d'accès. La CSE hôte peut fournir à l'expéditeur un *Local-Token-ID* qui peut servir à identifier le jeton.

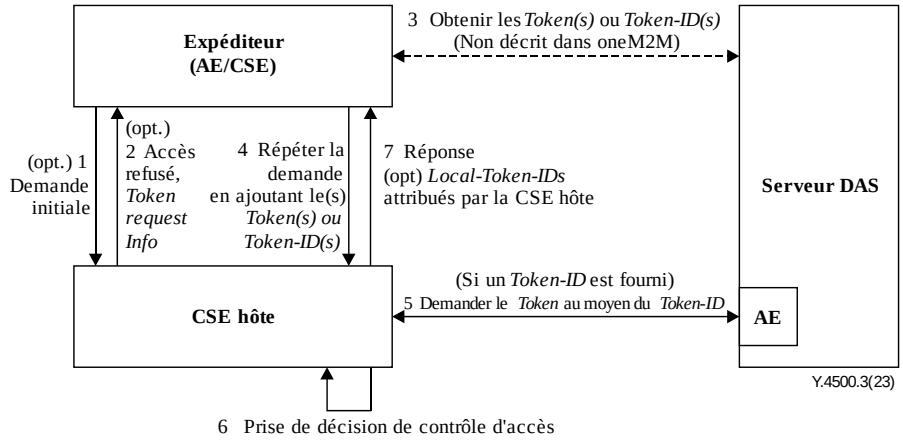


Figure 7.3.2.1-3 – Autorisation dynamique indirecte

7.3.2.2 Autorisation dynamique directe

La présente Recommandation décrit les paramètres échangés et le traitement associé au niveau de la CSE hôte. Le transport des paramètres est abordé au § 11.5.2 de la Recommandation [UIT-T Y.4500.1].

Les échanges de messages pour l'autorisation dynamique directe sont illustrés à la Figure 7.3.2.2-1, et décrits à la suite de la figure.

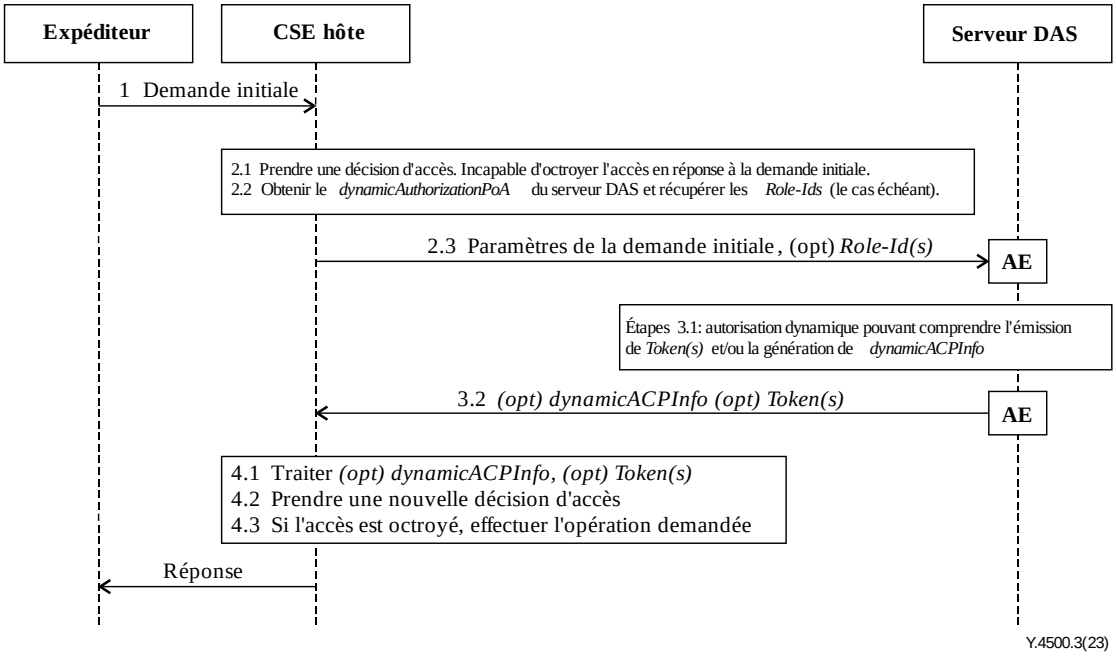


Figure 7.3.2.2-1 – Flux de messages lors d'une autorisation dynamique directe

1 L'expéditeur envoie la demande (appelée demande de l'expéditeur dans ce flux de messages) à la CSE hôte. Cette demande peut contenir des jetons (*Tokens*) ou identificateurs de jeton (*Token-ID*); voir le § 7.3.2.3 "Autorisation dynamique indirecte".

2 Traitement initial par la CSE hôte:

2.1 Si la demande de l'expéditeur comprend des informations de jeton (***Token*** ou ***Token-ID***), ces informations sont traitées comme décrit dans le § 7.3.2.3 "Autorisation dynamique indirecte". La CSE hôte évalue l'algorithme de décision d'accès, mais n'est pas capable d'accorder l'accès requis par la demande de l'expéditeur sur la base des politiques de contrôle d'accès configurées.

2.2 La CSE hôte détermine le serveur DAS avec lequel exécuter la procédure d'autorisation dynamique directe:

2.2.1 La CSE hôte examine toutes les règles de contrôle d'accès (*accessControlRules*) pour lesquelles la demande satisfait aux paramètres *accessControlOperations* et *accessControlContexts* dans les ressources <*accessControlPolicy*> liées à la ressource demandée. La CSE hôte collecte l'ensemble de tous les *Role-ID* contenus dans le paramètre *accessControlOperators* de ces *accessControlRules.* Les *Role-ID* sont regroupés en fonction de l'AE-ID du serveur DAS identifié par le *Role-ID*.

NOTE 1 – En ce qui concerne le paramètre Role-ID(s): l'expéditeur est autorisé à accéder à la ressource demandée si un ou plusieurs jetons associant l'expéditeur à un ou plusieurs Role-ID(s) sont générés. La fourniture de cette liste au serveur DAS permet à ce dernier de sélectionner un ensemble approprié d'un ou plusieurs Role-ID(s) à associer au(x) jeton(s) de l'expéditeur, autorisant ainsi l'expéditeur à accéder aux ressources demandées. Ce sont les politiques configurées dans le serveur DAS qui dictent quels identificateurs de rôle, le cas échéant, sont inclus dans le ou les jetons remis à l'expéditeur.

2.2.2 La CSE hôte doit également récupérer l'ensemble des ressources <*dynamicAuthorizationConsultation*> liées à la ressource demandée, et regrouper ces dernières en fonction de l'attribut *dynamicAuthorizationPoA* du serveur DAS de la ressource <*dynamicAuthorizationConsultation*>.

2.3 La CSE hôte sélectionne un serveur DAS (dans l'ensemble déterminé à l'étape 2.2) et envoie un message de demande oneM2M contenant les informations décrites au Tableau 7.3.2.21. Le transport des paramètres est abordé à l'étape 2.3, § 11.5.2 de la Recommandation [UIT-T Y.4500.1].

Tableau 7.3.2.2-1 – Informations envoyées par la CSE hôte au serveur DAS lors de l'autorisation dynamique directe

Paramètre	Description	Obligatoire/ Facultatif
Originator	Identificateur de l'expéditeur de la demande reçue par le destinataire	O
Originator Resource Type	Type de la ressource sollicitée par la demande initiale reçue par le destinataire	O
Operation	Type d'opération indiqué dans la demande initiale reçue par le destinataire	O
Originator IP Address	Adresse IP de l'expéditeur de la demande reçue par le destinataire	F
Originator Location	Emplacement géographique de l'expéditeur de la demande reçue par le destinataire	F
Originator Role IDs	Identificateur de rôle de l'expéditeur de la demande reçue par le destinataire	F
Request Timestamp	Horodatage de réception de la demande initiale par le destinataire	F
targeted Resource ID	Identificateur de la ressource ciblée par la demande initiale reçue par le destinataire	F
Proposed Privileges Lifetime	Proposition de durée de vie des privilèges d'autorisation demandés par le destinataire	F
Role IDs From ACPs	Ensemble de rôles d'accès dynamiques dans les paramètres *accessControlDynAuthRole* associés à l'AE-ID du serveur DAS	F
Token IDs	Ensemble d'identificateurs de jeton associés à l'expéditeur	F

3 Traitement par le serveur DAS:

3.1 Le serveur DAS traite les paramètres reçus. Le serveur DAS peut décider de fournir les *Token(s)* et/ou *dynamicACPInfo* qui seront utilisés par la CSE hôte pour créer une ressource <*accessControlPolicy*> dynamique. Le serveur DAS applique les politiques avec lesquelles il est configuré pour décider des actions appropriées.

NOTE 2 – Les détails de cette décision sont spécifiques au système d'autorisation dynamique utilisé; ces détails ne sont pas visibles par le système oneM2M et ne sont pas traités dans la présente Recommandation.

Le ou les jetons, le cas échéant, doivent être conformes au § 7.3.3.1 "Structure des jetons", et présenter le profil suivant:

– Le paramètre "holder" (émetteur) doit contenir le CSE-ID ou l'AE-ID absolu de l'expéditeur reçu de la CSE hôte; ce paramètre peut contenir d'autres CSE-ID et AE-ID.

– Le paramètre "audience" doit contenir uniquement le CSE-ID de la CSE hôte.

Le serveur DAS doit appliquer une option de protection ESData aux jetons individuels, en respectant les exigences ci-dessous:

– Le serveur DAS doit chiffrer le jeton pour que ce dernier puisse être déchiffré par la CSE hôte.

– La CSE hôte doit être en mesure de vérifier que le jeton provient bien du serveur DAS.

Le traitement ESData entraîne la création d'une enveloppe ESData, appelée *ESData-protected Token* pour les besoins du présent flux de messages.

Si le serveur DAS décide d'autoriser la CSE hôte à créer une ressource <*accessControlPolicy*> dynamique, il doit composer un paramètre *dynamicACPInfo* contenant les informations énumérées ci-après dans le Tableau 7.3.2.2-2.

Tableau 7.3.2.2-2 – Informations incluses dans le paramètre dynamicACPInfo

Paramètre	Description	Obligatoire/Facultatif
Granted Privileges	Liste des privilèges octroyés	F
Privileges Lifetime	Durée de vie des privilèges octroyés	F
Tokens	Liste des jetons émis	F

3.2 Le serveur DAS doit envoyer, via son AE, les éventuels jetons protégés par ESData et le paramètre facultatif *dynamicACPInfo* à la CSE hôte. Le transport des paramètres est abordé à l'étape 2.3, § 11.5.2 de la Recommandation [UIT-T Y.4500.1].

4 Traitement par la CSE hôte:

4.1 La CSE hôte traite les éventuels jetons protégés par ESData et, le cas échéant, le paramètre facultatif *dynamicACPInfo*:

4.1.1 La CSE hôte doit effectuer les vérifications suivantes pour chaque jeton protégé par ESData:

4.1.1.1 La CSE hôte doit appliquer le traitement ESData au jeton protégé par ESData, afin d'extraire le jeton authentifié.

4.1.1.2 La CSE hôte doit effectuer les vérifications suivantes:

4.1.1.2.1 Le paramètre "issuer" (émetteur) du jeton doit correspondre exactement à l'identité du serveur DAS.

4.1.1.2.2 L'identificateur de la CSE hôte doit correspondre au CSE-ID indiqué dans le paramètre "audience" du jeton.

4.1.1.2.3 Le paramètre "holder" (détenteur) du jeton doit correspondre exactement au CSE-ID ou AE-ID absolu de l'expéditeur de la demande.

4.1.1.2.4 La CSE hôte doit vérifier que le jeton n'a pas expiré, en comparant l'heure actuelle au paramètre "notAfter" (pas après) du jeton.

4.1.1.3 La CSE hôte doit placer le jeton vérifié dans le cache, puis le supprimer quand il arrive à expiration (comme défini à l'étape 4.1.1.2.4).

4.1.2 Si le serveur DAS a fourni un paramètre *dynamicACPInfo*, la CSE hôte doit créer une ressource <*accessControlPolicy*> dynamique correspondant à *dynamicACPInfo*.

4.2 La CSE hôte répète le mécanisme de décision d'accès décrit au § 7.1.4 "Décision de contrôle d'accès".

4.3 Si l'accès est accordé, la CSE hôte effectue l'opération requise dans la demande de l'expéditeur, ce qui entraîne l'envoi d'un message à l'expéditeur par la CSE hôte.

7.3.2.3 Autorisation dynamique indirecte

La présente Recommandation décrit les paramètres échangés et le traitement associé au niveau de l'expéditeur et de la CSE hôte. Le transport des paramètres est abordé au § 11.5.3 de la Recommandation [UIT-T Y.4500.1].

Les échanges de messages pour l'autorisation dynamique indirecte sont illustrés à la Figure 7.3.2.3-1, et décrits à la suite de la figure.

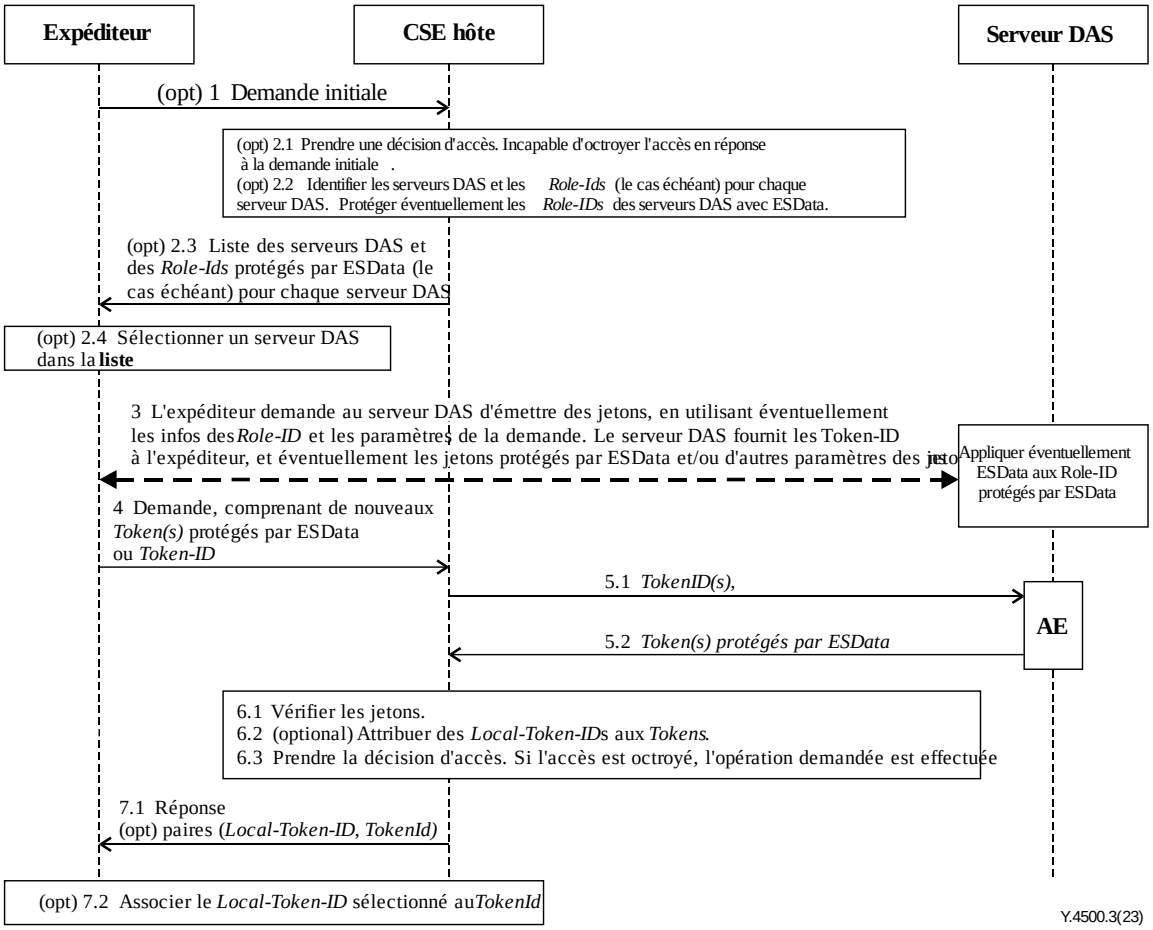


Figure 7.3.2.3-1 – Flux de messages lors d'une autorisation dynamique indirecte

1 (Étape facultative) L'expéditeur envoie une demande à la CSE hôte. Il inclut une information indiquant qu'il est préparé à demander des jetons au serveur DAS pour cette demande. La demande peut contenir une combinaison d'informations de type *Token*, *tokenID*, *Local-Token-ID*, mais le présent trajet de message suppose que celles-ci ne fournissent pas de permissions suffisantes pour accéder à la ressource demandée.

2 (Étape facultative) Traitement initial par la CSE hôte:

2.1 La CSE hôte détermine la décision d'accès en réponse à la demande de l'expéditeur. Ce flux de messages suppose que la demande de l'expéditeur est rejetée à la suite de la décision d'accès. La CSE hôte prend note du fait que l'expéditeur est préparé à demander des jetons au serveur DAS pour cette demande.

2.2 La CSE hôte établit une liste de serveurs DAS et d'identificateurs de rôles associés (le cas échéant), comme décrit à l'étape 2.2.1 de la procédure d'autorisation dynamique directe détaillée au § 7.3.2.3 "Autorisation dynamique directe".

Pour chaque serveur DAS, la CSE hôte peut appliquer une protection ESData à l'ensemble d'identificateurs de rôle, que le serveur DAS déchiffrera. Par exemple, la protection ESData peut chiffrer l'ensemble d'identificateurs de rôle pour que ceux-ci ne soient pas visibles par l'expéditeur.

2.3 La CSE hôte doit envoyer un message de refus d'accès à l'expéditeur, en y incluant la liste des serveurs DAS et l'ensemble d'identificateurs de rôle associé, éventuellement protégé par un chiffrement ESData.

2.4 L'expéditeur sélectionne un serveur DAS dans la réponse.

3 L'expéditeur demande à ce serveur DAS d'émettre un jeton (*Token*). L'expéditeur peut fournir au serveur DAS l'ensemble des identificateurs de rôle associés, éventuellement protégés par un chiffrement ESData, ainsi que les paramètres que contenait la demande d'accès initiale. Si les identificateurs de rôle sont protégés par ESData, le serveur DAS applique la procédure ESData pour les extraire. Le serveur DAS émet un ou plusieurs jetons puis il renvoie ces jetons et les identificateurs de jeton associés, éventuellement protégés par ESData, à l'expéditeur. Le serveur DAS peut également fournir d'autres paramètres du jeton à l'expéditeur, par exemple la durée de validité du jeton. Cette interaction est spécifique à la technologie du système d'autorisation dynamique employé.

4 L'expéditeur ajoute à sa demande d'origine le jeton protégé par ESData fourni par le serveur DAS, ou le *token-ID* (si aucun jeton protégé par ESData n'a été fourni) si le jeton correspondant protégé par ESData n'a pas été fourni par le serveur DAS. En particulier, si la demande de l'étape 1 n'a pas obtenu de réponse positive à l'étape 2.3, l'expéditeur peut renvoyer sa demande en y incluant de nouveaux paramètres *Token* et/ou *token-ID*. Un même jeton peut être utilisé dans plusieurs demandes.

L'expéditeur doit envoyer la demande à la CSE hôte.

5 (Étape facultative) Si la demande contient un ou plusieurs *token-ID*, la CSE hôte identifie, pour chacun d'eux, l'AE de serveur DAS correspondante, auprès de laquelle il convient de demander le jeton associé:

5.1 La CSE hôte envoie les identificateurs de jeton au serveur DAS, via l'AE du serveur DAS.

5.2 Le serveur DAS renvoie à la CSE hôte, via l'AE du serveur DAS, les jetons valides correspondants protégés par ESData.

6 Traitement par la CSE hôte:

6.1 Traitement du jeton:

– L'identificateur de la CSE hôte doit correspondre à l'un des CSE-ID absolus (comprenant éventuellement des caractères de troncation) dans le paramètre "audience" du jeton.

– Le paramètre "holder" (émetteur) du jeton doit correspondre exactement au CSE-ID ou AE-ID absolu de l'expéditeur à l'origine de la demande.

– La CSE hôte doit vérifier que le jeton n'a pas expiré, en comparant l'heure en cours aux paramètres "notBefore" (pas avant) et "notAfter" (pas après) du jeton. Un jeton mis en cache qui a expiré peut être supprimé du cache.

6.1.1 La CSE hôte doit appliquer le déchiffrement ESData aux jetons protégés par ESData, inclus dans le message de demande ou récupérés auprès du serveur DAS, afin d'extraire les jetons authentifiés.

6.1.2 Si la demande contient un *Local-Token-ID*, la CSE hôte tente récupérer le jeton en question dans le cache.

6.1.3 La CSE hôte doit effectuer les vérifications suivantes, pour chaque jeton authentifié mis en cache associé à la demande:

– L'identificateur de la CSE hôte doit correspondre à l'un des CSE-ID absolus (comprenant éventuellement des caractères de troncation) dans le paramètre "audience" du jeton.

– Le paramètre "holder" (émetteur) du jeton doit correspondre exactement au CSE-ID ou à l'AE-ID absolu de l'expéditeur à l'origine de la demande.

– La CSE hôte doit vérifier que le jeton n'a pas expiré, en comparant l'heure en cours aux paramètres "notBefore" et "notAfter" du jeton. Un jeton mis en cache qui a expiré peut être retiré du cache.

6.1.4 S'il n'a pas été possible de récupérer un jeton identifié à l'étape 5 ou 6.1.2, ou si un identificateur de jeton protégé par ESData échoue aux vérifications de l'étape 6.1.1, ou encore si un jeton échoue aux vérifications de l'étape 6.1.3, la CSE hôte doit renvoyer un message d'erreur.

6.1.5 La CSE hôte peut mettre en cache tout nouveau jeton.

6.2 La CSE hôte peut attribuer des *Local-Token-ID* aux jetons mis en cache.

6.3 La CSE hôte doit prendre la décision d'accès comme décrit au § 7.1.4, en incluant dans le processus les informations des jetons identifiés dans la demande. Si l'accès est autorisé, l'opération demandée doit être exécutée.

7 Réponse:

7.1 La CSE hôte envoie une réponse à l'expéditeur. Pour chaque nouveau *Local-Token-ID* qui a été assigné, la CSE hôte fournit le *Local-Token-ID* et le *token-ID* correspondant dans les paramètres de la réponse.

7.2 L'expéditeur associe le *Local-Token-ID* au *token-ID*. Lors des demandes ultérieures, l'expéditeur pourra utiliser ce *Local-Token-ID* au lieu du *Token* ou du *token-ID*.

7.3.2.4 Structure des jetons

Un jeton sert à transporter des informations d'autorisation, qui peuvent être des rôles assignés au détenteur du jeton, ou des politiques de contrôle d'accès applicables au détenteur du jeton. La structure d'un jeton, illustrée à la Figure 7.3.2.4-1, contient les champs de données suivants:

version: version du jeton.

tokenID: identificateur unique du jeton.

holder: identificateur du détenteur du jeton.

issuer: identificateur de l'émetteur du jeton.

notBefore: date/heure de début de validité du jeton.

notAfter: date/heure d'expiration du jeton.

tokenName: (facultatif) nom lisible en clair du jeton.

audience: (facultatif) liste des CSE-ID des CSE censées accepter le jeton.

permissions: permissions associées au jeton. Son format est décrit au § 9.6.39 de la Recommandation [UIT-T Y.4500.1].

extension: champ servant à stocker d'autres informations, qui peuvent par exemple être spécifiques à une application.

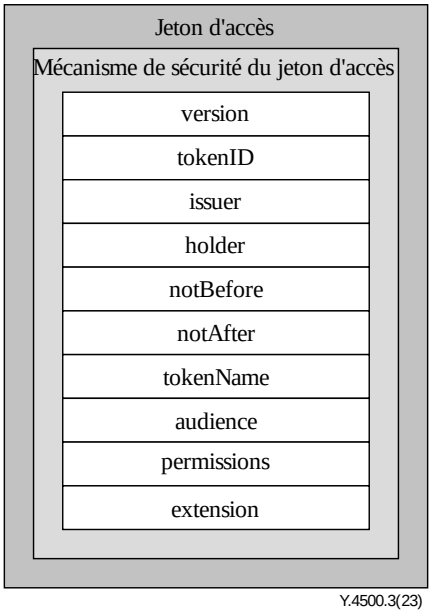


Figure 7.3.2.4-1 – Structure d'un jeton

Un jeton doit être protégé par le mécanisme de sécurité ESData. Un jeton doit être signé, chiffré, ou signé et chiffré.

7.3.2.5 Évaluation des jetons

Le processus générique d'évaluation d'un jeton peut être décrit comme suit:

Validation de la sécurité du jeton: suivant le mécanisme de sécurité employé par un jeton, la validation peut consister à:

– vérifier un jeton signé;
– déchiffrer un jeton chiffré; ou
– déchiffrer et vérifier un jeton chiffré et signé.

Lorsque le jeton passe avec succès la validation de sécurité, le texte en clair qu'il contient peut servir à effectuer une validation complémentaire.

Validation du contenu du jeton: suivant le contenu du jeton, la validation peut consister à vérifier:

– si l'identité de l'expéditeur correspond au détenteur indiqué dans le champ *holder*;
– si l'émetteur du jeton indiqué dans le champ *issuer* est valide;
– si le jeton n'a pas expiré, en fonction des champs *notBefore* et *notAfter*;
– si l'identificateur de la CSE hôte figure dans la liste des CSE-ID indiquée dans le champ *audience* (si ce champ est renseigné).

Lorsque le jeton passe avec succès la validation du contenu, les permissions qu'il contient doivent être utilisées pour le contrôle d'accès.

Évaluation des permissions du jeton: cette opération consiste à vérifier un par un chaque élément de la liste des permissions, jusqu'à ce que la demande d'accès soit autorisée par l'une des permissions, ou jusqu'à la fin de la liste. Pour chaque élément de la liste de permissions, l'évaluation se déroule de la manière suivante:

On vérifie l'élément *resourceIDs*. Si cet élément est présent, les informations d'autorisation décrites dans les éléments *privileges* et/ou *roleID* s'appliquent uniquement aux ressources nommées par *resourceIDs*. Si l'élément *privileges* est présent, l'élément *resourceIDs* doit également être présent.

Si l'élément *privileges* est présent, les règles de contrôle d'accès qu'il contient doivent être utilisées en tant que politique de contrôle d'accès applicable pendant le processus de décision d'accès en cours.

Si l'élément *roleIDs* est présent, les identificateurs de rôle qu'il contient doivent être utilisés en tant que rôles valides pendant le présent processus de décision d'accès.

7.3.2.6 Jetons web JSON oneM2M (JWT)

7.3.2.6.1 Introduction aux jetons web JSON oneM2M (JWT)

La spécification oneM2M décrit une représentation de jetons web JSON (JWT) [IETF RFC 7519] applicable aux jetons utilisés dans le système oneM2M. Un jeton JWT conforme au présent paragraphe est appelé *jeton JWT oneM2M.*

Contexte: un jeton JWT utilise soit la représentation compacte de la signature web JSON (JWS), soit la représentation compacte du chiffrement web JSON (JWE), lesquelles sont abordées respectivement dans les documents [IETF RFC 7515] et [IETF RFC 7519]. La spécification des jetons JWT [IETF RFC 7519] définit également un jeton JWT non sécurisé, qui est une représentation JWS dans laquelle le paramètre d'en-tête "alg" a la valeur "none" et la signature JWS a comme valeur une chaîne vide.

La spécification JWT définit un élément JSON comme étant la structure de la charge utile de la signature JWS ou du chiffrement JWE employés comme jeton JWT. Cette charge utile comprend un jeu de déclarations JWT, une liste initiale de noms de déclarations JWT étant normalisée dans le document [IETF RFC 7519]. L'IANA tient à jour un registre de noms de déclarations JWT [b-IANA JWT].

7.3.2.6.2 Profil des jetons JWT oneM2M

Déclarations oneM2M: le Tableau 7.3.2.6.2-1 présente les correspondances entre les noms de déclaration JWT d'un jeton JWToneM2M et les éléments du type de données complexe m2m:tokenClaimSet décrit dans la Recommandation [UIT-T Y.4500.4]. On a utilisé, quand ils étaient disponibles, les noms de déclaration JWT enregistrés par l'IANA [b-IANA JWT]. La Recommandation [UIT-T Y.4500.4] précise les éléments obligatoires et les éléments facultatifs.

Tableau 7.3.2.6.2-1 – Liste des déclarations des jetons oneM2M JWT et leur correspondance avec les éléments de m2m:tokenClaimSet

Chemin de l'élément Token Claimset Object	Nom court de l'élément Token Claimset Object	Nom de déclaration JWT oneM2M	Où est défini ce nom de déclaration JWT?	Détails supplémentaires pour le mappage des valeurs Token Claimset Object aux valeur JWT Claim
version	***tkvr***	"tkvr"	Noms courts [UIT-T Y.4500.4]	Les valeurs doivent être identiques
tokenID	***tkid***	"jti"	[IETF RFC 7519]	Les valeurs doivent être identiques
issuer	***tkis***	"iss"	[IETF RFC 7519]	Les valeurs doivent être identiques
holder	***tkhd***	"azp"	OpenID Connect Core 1.0 [OIDF CC]	Les valeurs doivent être identiques
notBefore	***tknb***	"nbf"	[IETF RFC 7519]	L'élément Token Claimset Object "notBefore" est au format basique de l'ISO 8601, voir [UIT-T Y.4500.4]. Cet élément doit être mappé à la déclaration JWT "nbf", qui utilise le format NumericDate [RFC7519].

Tableau 7.3.2.6.2-1 – Liste des déclarations des jetons oneM2M JWT et leur correspondance avec les éléments de m2m:tokenClaimSet

Chemin de l'élément Token Claimset Object	Nom court de l'élément Token Claimset Object	Nom de déclaration JWT oneM2M	Où est défini ce nom de déclaration JWT?	Détails supplémentaires pour le mappage des valeurs Token Claimset Object aux valeur JWT Claim
notAfter	***tkna***	"exp"	[IETF RFC 7519]	L'élément Token Claimset Object "notAfter" est au format basique de l'ISO 8601, voir [UIT-T Y.4500.4]. Cet élément doit être mappé à la déclaration JWT "exp", qui utilise le format NumericDate [RFC7519].
tokenName	***tknm***	"tknm"	Noms courts [UIT-T Y.4500.4]	Les valeurs doivent être identiques
audience	***tkau***	"aud"	[IETF RFC 7519]	L'élément Token Claimset Object "audience" est une liste d'identificateurs m2m:ID. Cette liste doit être mappée à la déclaration JWT "aud" qui comprend un tableau de chaînes de caractères sensibles à la casse, contenant chacune une valeur StringOrURI [RFC7519].
permissions	***tkps***	"tkps"	Noms courts [UIT-T Y.4500.4]	Les valeurs doivent être identiques
extension	***tkex***	"tkex"	Noms courts [UIT-T Y.4500.4]	Les valeurs doivent être identiques

Profil de sécurité oneM2M JWT: la représentation compacte JWS et la représentation compacte JWE sont toutes deux prises en charge par ESData (voir le § 8.5.3). Un jeton JWT oneM2M peut utiliser n'importe quelle classe de sécurité ESData: *Encryption-only* (chiffrement seul), *Signature-only* (signature seule) ou *Nested-Sign-then-encrypt* (signature puis chiffrement imbriqués). Un jeton JWT oneM2M peut utiliser n'importe quel algorithme pris en charge par ESData pour la représentation compacte JWS et la représentation compacte JWE.

Un jeton JWT oneM2M peut être un jeton JWT non sécurisé, auquel cas on considèrera que le jeton JWT oneM2M utilise la classe de sécurité ESData non sécurisée.

Le document [IETF RFC 7519] aborde des points relatifs à la sécurité des jetons JWT, et les opérateurs de systèmes d'émission de jetons (serveurs d'autorisation dynamique et organismes d'autorisation) devraient consulter ce texte lors du choix de la classe de sécurité et des algorithmes ESData.

Paramètres d'en-tête JOSE des jetons JWT oneM2M: lorsque la classe de sécurité "chiffrement seul" du cadre ESData est utilisée:

– l'en-tête JOSE de la représentation JWE doit contenir les paramètres "typ" définis sur "JWT";

– l'en-tête JOSE de la représentation JWE ne doit pas contenir le paramètre "cty".

Lorsque la classe de sécurité "signature seule" du cadre ESData est utilisée:

– l'en-tête JOSE de la représentation JWS doit contenir les paramètres "typ" définis sur "JWT";

– l'en-tête JOSE de la représentation JWS ne doit pas contenir le paramètre "cty".

Lorsque la classe de sécurité "signature puis chiffrement imbriqués" du cadre ESData est utilisée, les déclarations JWT constituent la charge utile d'une représentation JWS, et la représentation JWS devient la charge utile d'une représentation JWE. Dans ce cas:

– l'en-tête JOSE de la représentation JWS et de la représentation JWE doivent contenir les paramètres "typ" définis sur "JWT";
– l'en-tête JOSE de la représentation JWE doit contenir les paramètres "cty" définis sur "JWT", pour indiquer qu'un jeton JWT imbriqué est transporté dans ce JWT;
– l'en-tête JOSE de la représentation JWS ne doit pas contenir le paramètre "cty".

7.3.2.6.3 Procédures relatives aux jetons JWT oneM2M

Configuration des CSE pour qu'elles vérifient les jetons émis par un émetteur de jetons: pour qu'une CSE vérifie des jetons JWT oneM2M émis par un émetteur de jetons donné, cette CSE doit recevoir les informations suivantes de façon sécurisée:

– les combinaisons des classes de sécurité ESData et des algorithmes permis par l'émetteur de jetons;
– des justificatifs d'identité de vérification des jetons conformes à ces classes de sécurité et ces algorithmes, en notant qu'aucun justificatif d'identité n'est requis pour vérifier des jetons utilisant la classe non sécurisée du cadre ESData.

La présente Recommandation ne décrit par les mécanismes permettant de fournir ces informations à la CSE. Elle n'aborde pas non plus les structures de données permettant de stocker ces informations au niveau de la CSE. Le niveau de sécurité à appliquer sur chaque CSE particulière doit être déterminé par une évaluation des risques spécifiques à l'application.

Création d'un jeton JWT oneM2M: lorsqu'un émetteur de jetons est invité à créer un jeton, il doit effectuer les étapes suivantes:

– L'émetteur de jetons doit former un élément Token Claimset Object conforme au type de données m2m:tokenClaimSet, dans lequel l'élément *permission* utilise la sérialisation JSON.
– L'émetteur de jetons doit créer le jeu de déclarations de jeton JWT oneM2M correspondant, en s'appuyant sur les correspondances présentées au Tableau 7.3.2.6.2-1.
– L'émetteur de jetons doit choisir une classe de sécurité ESData, les algorithmes et les justificatifs d'identité correspondants. Il est possible d'effectuer cette étape avant l'Étape 1), ou entre les Étapes 1) et 2).
– L'émetteur de jetons doit créer un jeton JWT oneM2M à partir des déclarations de jeton JWT oneM2M, de la classe de sécurité ESData, des algorithmes et des justificatifs d'identité correspondants. Cette étape suit la procédure décrite pour les jetons JWT dans le document [IETF RFC 7519].

Le jeton JWT oneM2M résultant est du type de données m2m:dynAuthJWT.

Validation d'un jeton JWT oneM2M: lorsqu'une CSE reçoit un jeton JWT oneM2M à utiliser dans une décision d'accès, elle doit effectuer les étapes suivantes:

– La CSE doit vérifier que le jeton JWT oneM2M est conforme au type de données m2m:dynAuthJWT.
– La CSE doit valider la sécurité du jeton JWT oneM2M, comme décrit au § 7.3.2.5, en se basant sur les détails spécifiques aux JWT indiqués dans le document [IETF RFC 7519] et, le cas échéant, sur les justificatifs d'identité configurés. Une CSE doit écarter un jeton JWT oneM2M qui utilise une classe de sécurité ESData ou des algorithmes non autorisés par l'émetteur de jetons.

– La CSE doit créer un élément Token Claimset Object à partir de la liste de déclarations JWT oneM2M, en inversant les correspondances présentées au Tableau 7.3.2.6.2-1.

– La CSE doit valider l'élément Token Claimset Object comme décrit au § 7.3.2.5.

L'élément *permissions* du Token Claimset Object peut à présent être traité comme décrit au § 7.3.2.5.

7.4 Contrôle d'accès fondé sur les rôles

7.4.1 Architecture du contrôle d'accès fondé sur les rôles

La Figure 7.4.1-1 donne un aperçu de haut niveau de l'architecture de contrôle d'accès fondée sur les rôles dans le système oneM2M. Les entités impliquées dans l'attribution des rôles et le contrôle d'accès fondé sur les rôles sont décrites ci-après:

– Autorité d'autorisation: autorité chargée d'attribuer les rôles aux expéditeurs par le biais de la création de ressources <*role*> et/ou <*token*> dans des référentiels de rôles et/ou de jetons.

– Référentiel de rôles: CSE chargée de stocker les ressources <*role*>.

– Référentiel de jetons: CSE chargée de stocker les ressources <*token*>.

NOTE 1 – Les flèches sur la Figure 7.4.1-1 représentent les relations logiques entre entités, et non les relations d'enregistrement qui constituent les chemins réels des données.

NOTE 2 – Les fonctions du PDP et du PIP sont combinées dans la CSE hôte pour la version actuelle de oneM2M.

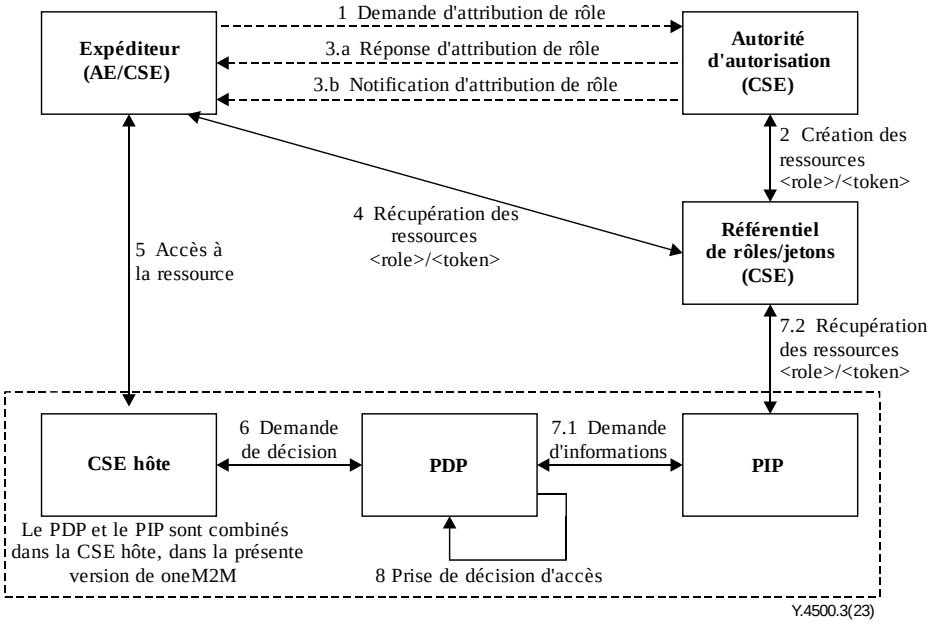


Figure 7.4.1-1 – Architecture de contrôle d'accès fondée sur les rôles

La procédure générique associée à cette architecture est décrite ci-après:

Étape 001: Un expéditeur peut émettre une demande de rôle auprès d'une autorité d'autorisation. Cette étape peut ne pas exister dans certaines situations, par exemple lorsque l'autorité d'autorisation attribue directement un rôle à un expéditeur.

Cette étape n'est pas décrite dans la présente Recommandation.

Étape 002: L'autorité d'autorisation doit vérifier si le privilège appliqué peut être attribué à l'expéditeur. Si la réponse est oui, l'autorité d'autorisation doit créer une ressource <*role*> qui indique l'attribution du rôle dans un référentiel de rôles, ou émettre un jeton qui contient le rôle attribué à l'expéditeur. Le jeton émis peut être stocké dans une ressource <*token*>, au sein d'un référentiel de jetons.

Étape 003: L'autorité d'autorisation informe l'expéditeur du résultat de l'attribution de rôle. Parmi les informations renvoyées peuvent figurer l'identificateur de rôle, l'identificateur de jeton, le jeton ou les informations relatives aux ressources <*role*> ou <*token*> créées. On envisagera deux cas:

a) Dans le cas où l'expéditeur envoie une requête d'attribution de rôle à l'autorité d'autorisation, cette dernière renvoie l'attribution de rôle via une réponse d'attribution de rôle.

b) Dans le cas où l'autorité d'autorisation attribue directement un rôle à un expéditeur, l'autorité d'autorisation renvoie l'information via une notification d'attribution de rôle.

Cette étape n'est pas décrite dans la présente Recommandation.

Étape 004: L'expéditeur peut récupérer les rôles attribués et/ou les jetons auprès des référentiels de rôles et de jetons au moyen des informations fournies par une autorité d'autorisation, afin d'obtenir des informations détaillées à propos des rôles attribués et/ou des jetons.

Étape 005: L'expéditeur envoie une demande d'accès à la ressource cible dans la CSE hôte. La demande peut contenir les informations de rôle, qui peuvent être les identificateurs de rôle, les jetons ou les identificateurs de jetons.

Étape 006: La CSE hôte peut envoyer une demande de décision de contrôle d'accès à un PDP.

Étape 007: Le PDP peut avoir besoin de récupérer les informations d'attribution de rôle de l'expéditeur, en fonction des identificateurs de rôle, et/ou des identificateurs de jeton auprès des référentiels de rôles et de jetons.

Étape 008: Le PDP vérifie les rôles et/ou les jetons de l'expéditeur, puis il prend une décision de contrôle d'accès en fonction des politiques de contrôle d'accès et des rôles.

7.4.2 Procédure d'attribution des rôles

7.4.2.1 Introduction

L'attribution de rôles à un expéditeur peut s'effectuer de deux manières.

La première consiste à créer une ressource <*role*>, qui décrit le rôle attribué à l'expéditeur. Les ressources <*role*> sont stockées dans des référentiels de rôles auprès desquels les AE et les CSE peuvent obtenir les ressources <*role*> afin de récupérer les informations relatives à l'attribution de rôles à un expéditeur.

La seconde consiste à émettre un jeton qui décrit le rôle attribué au détenteur du jeton (c'est-à-dire l'expéditeur). Le jeton émis peut également être stocké dans une ressource <*token*> dans un référentiel de jetons, auprès duquel on peut récupérer les informations relatives au jeton émis.

Une ressource <*role*> peut également pointer vers une ressource <*token*> dans laquelle est stocké le jeton contenant le rôle attribué, par le biais d'un attribut *tokenLink*.

7.4.2.2 Procédure d'attribution de rôle

La procédure générale d'attribution de rôle à un expéditeur est illustrée à la Figure 7.4.2.2-1 et décrite ci-après:

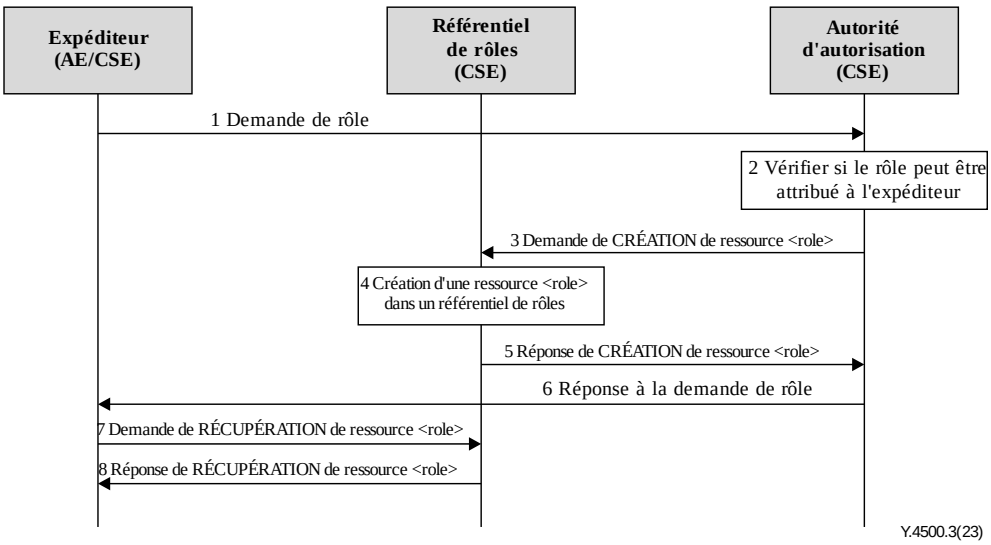


Figure 7.4.2.2-1 – Procédure d'attribution de rôle

La procédure d'attribution de rôle est la suivante:

1 L'expéditeur peut envoyer une demande d'attribution de rôle à l'autorité d'autorisation. La description de cette étape est cependant hors du champ d'application de la présente Recommandation.

2 L'autorité d'autorisation doit vérifier si le rôle demandé peut être assigné à l'expéditeur. Une fois cette vérification effectuée avec succès, l'autorité d'autorisation attribue le rôle à l'expéditeur.

3 L'autorité d'autorisation doit envoyer une demande de création de ressource <*role*> au référentiel de rôles.

4 Le référentiel de rôles doit créer une ressource <*role*>, conformément à la demande de création.

5 Le référentiel de rôles doit renvoyer le résultat de la création de la ressource <*role*> à l'autorité d'autorisation.

6 L'autorité d'autorisation doit à son tour renvoyer le résultat de l'attribution de rôle à l'expéditeur. La description de cette étape est cependant hors du champ d'application de la présente Recommandation.

7 L'expéditeur doit envoyer la demande de récupération de la ressource <*role*> au référentiel de rôles, afin d'obtenir les informations d'attribution de rôle.

8 Le référentiel de rôles doit renvoyer le contenu de la ressource <*role*> à l'expéditeur.

7.4.2.3 Émission d'un jeton associé à un rôle

La procédure générale d'émission d'un jeton de rôle (jeton associé à un rôle assigné) à un expéditeur est illustrée à la Figure 7.4.2.3-1 et décrite ci-après:

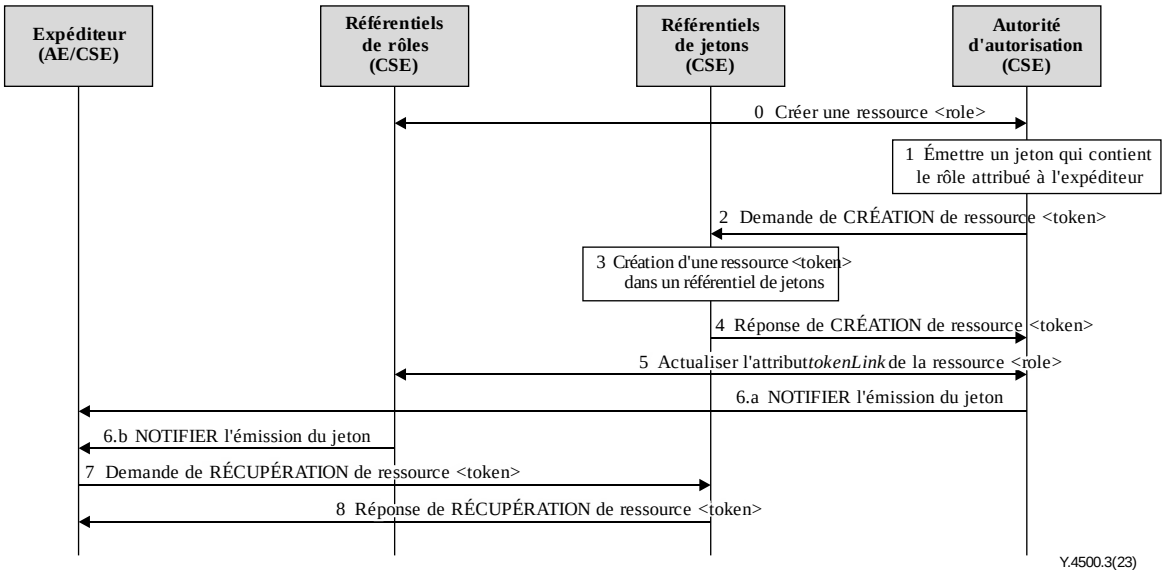


Figure 7.4.2.3-1 – Procédure d'émission de jeton de rôle

La procédure d'émission d'un jeton de rôle est la suivante:

0 L'autorité d'autorisation crée une ressource <*role*> dans un référentiel de rôles en vue de l'attribution d'un rôle.

1 L'autorité d'autorisation émet un jeton qui contient le rôle attribué à l'expéditeur.

2 L'autorité d'autorisation doit envoyer une demande de création de ressource <*token*> à un référentiel de rôles.

3 Le référentiel de rôles doit créer une ressource <*token*>, conformément à la demande de création.

4 Le référentiel de rôles doit renvoyer le résultat de la création de la ressource <*token*> à l'autorité d'autorisation.

5 L'autorité d'autorisation doit mettre à jour l'attribut *tokenLink* de la ressource <*role*> en y ajoutant l'adresse de la ressource <*token*>.

6 Les informations relatives à l'émission d'un jeton doivent être transmises à l'expéditeur. Cette notification peut s'effectuer de deux manières:

a) L'autorité d'autorisation informe l'expéditeur par le biais de l'opération NOTIFY.

b) Le référentiel de jetons informe (NOTIFY) l'expéditeur en vertu de l'abonnement de l'expéditeur à la ressource <*role*>.

7 L'expéditeur peut envoyer une demande de récupération de la ressource <*token*> au référentiel de jetons, afin d'obtenir les informations d'émission de jeton.

8 Le référentiel de jetons doit renvoyer le contenu de la ressource <*token*> à l'expéditeur.

7.4.3 Procédure de contrôle d'accès fondée sur les rôles

La procédure générale d'utilisation d'un rôle dans un processus d'autorisation est illustrée à la Figure 7.4.3-1 et décrite ci-après:

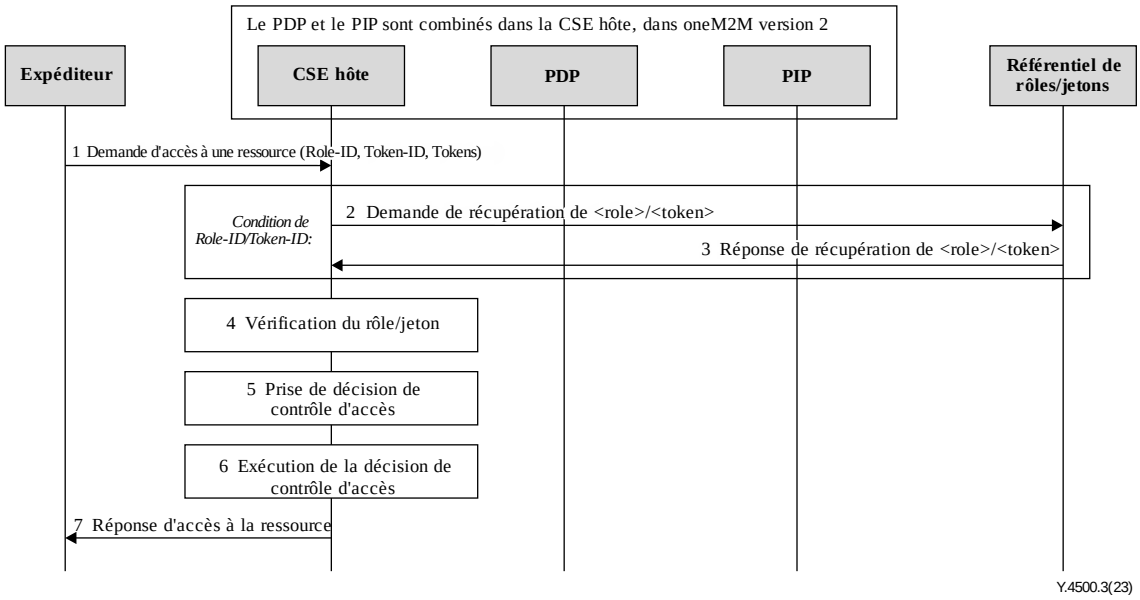


Figure 7.4.3-1 – Procédure de contrôle d'accès fondée sur les rôles

1 L'expéditeur doit inclure les identificateurs de rôle, identificateurs de jeton ou jetons applicables dans la requête envoyé à la CSE hôte.

2 Dans le cas d'identificateurs de rôle ou d'identificateurs de jetons: la CSE hôte, qui joue le rôle du PIP, doit envoyer une demande de récupération de ressource <*role*> ou <*token*> au référentiel de rôles ou de jetons.

3 Le référentiel de rôles ou de jetons doit renvoyer les attributs de la ressource <*role*> ou <*token*> à la CSE hôte.

4 La CSE hôte (qui joue le rôle du PDP) doit vérifier les rôles et/ou les jetons reçus, la vérification portant sur les trois points suivants: le rôle/jeton est émis par une autorité d'autorisation valide; le détenteur du rôle/jeton est bien l'expéditeur; et le rôle/jeton est toujours valide. Seuls des rôles et jetons valides doivent être utilisés pour le contrôle d'accès.

5 La CSE hôte (qui joue le rôle du PDP) doit examiner la demande d'accès de l'expéditeur en vertu des politiques de contrôle d'accès et/ou des identificateurs de rôle, afin de prendre une décision de contrôle d'accès comme décrit au § 7.1.

6 La CSE hôte doit appliquer la décision de contrôle d'accès, c'est-à-dire soit accéder à la ressource pour le compte de l'expéditeur, soit refuser l'accès à la ressource.

7 La CSE hôte doit renvoyer le résultat de l'accès à la ressource à l'expéditeur.

8 Cadres de sécurité

8.1 Introduction générale aux cadres de sécurité

8.1.0 Généralités

Pour tenir compte de la variété des scénarios de déploiement qui peuvent être rencontrés dans les applications M2M, la présente spécification prend en charge diverses méthodes visant à configurer et établir la sécurité dans les systèmes M2M.

8.1.1 Introduction générale aux cadres de sécurité à clé symétriques

Dans les cadres de sécurité à clés symétriques, chaque paire d'entités devant s'authentifier mutuellement reçoit sa propre clé symétrique partagée. Cette opération est effectuée par préconfiguration, par exemple pendant la fabrication ou le déploiement du dispositif, ou par le biais d'un cadre de configuration à distance de la sécurité.

8.1.2 Introduction générale aux cadres de sécurité fondés sur des certificats

8.1.2.0 Introduction

Ce paragraphe décrit la configuration des justificatifs d'identité et la vérification des certificats effectuées dans le cadre d'établissement d'association de sécurité fondé sur des certificats et le cadre de configuration à distance de la sécurité fondé sur des certificats.

8.1.2.1 Variantes de certificats de clé publique

La présente Recommandation définit des procédures qui s'appuient sur les variantes suivantes de certificats de clé publique:

Certificats de clé publique brute:

Description: un certificat de clé publique brute ([IETF RFC 7250]) contient uniquement la clé publique brute, sans les autres informations habituellement fournies par les certificats. Le certificat de clé publique brute est échangé lors de la prise de contact TLS, à la place d'un certificat classique (voir [IETF RFC 7250]).

Utilisation: un certificat de clé publique brute peut être utilisé pour authentifier une CSE ou une AE pendant la phase de prise de contact de sécurité de l'association (de la procédure d'établissement d'association de sécurité fondée sur des certificats), ou pendant la phase de prise de contact d'inscription d'amorçage (du cadre de configuration à distance de la sécurité fondé sur des certificats).

Certificats de dispositifs:

Description: ces certificats sont reliés, par une chaîne de certificats, à une ancre de confiance et ils contiennent au moins un identificateur d'instance matérielle unique mondialement (comme par exemple les identificateurs de dispositif M2M fondés sur un identificateur d'objet abordés à l'Annexe H "Identificateur de dispositif M2M fondé sur un identificateur d'objet" de la Recommandation [UIT-T Y.4500.1]) dans l'extension subjectAltName du certificat. Un certificat de dispositif peut permettre de vérifier l'identité de l'instance matérielle sur laquelle est exécutée l'entité.

Utilisation: un certificat de dispositif peut être utilisé pour authentifier une CSE ou une AE exécutée sur un dispositif M2M donné. Si le dispositif M2M est un nœud de type ASN ou MN (qui prend en charge une CSE), le certificat de dispositif est associé implicitement à la CSE exécutée sur ce dispositif. Si le dispositif est un nœud de type ADN (qui ne prend pas en charge une CSE), le certificat de dispositif n'est pas associé implicitement à une AE spécifique exécutée sur l'appareil. Un certificat de dispositif peut être utilisé pour authentifier une CSE de terrain pendant la phase de prise de contact de sécurité de l'association du cadre d'établissement d'association de sécurité fondé sur des certificats, ou pendant la phase d'amorçage du cadre de configuration à distance de la sécurité fondé sur des certificats.

Certificats d'identificateur de nœud:

Description: ces certificats sont reliés, par une chaîne de certificats, à une ancre de confiance et ils contiennent l'identificateur d'un nœud (voir la Recommandation [UIT-T Y.4500.1]) dans l'extension subjectAltName du certificat. Il est possible d'utiliser un certificat d'identificateur de nœud pour vérifier l'identité d'un nœud.

Utilisation: un certificat d'identificateur de nœud peut être utilisé pour authentifier un principal de sécurité sur un nœud agissant pour le compte de la CSE et/ou d'une ou plusieurs AE.

Certificats de CSE-ID:

Description: ces certificats sont reliés, par une chaîne de certificats, à une ancre de confiance et ils contiennent la représentation d'un CSE-ID sous la forme d'un nom de domaine public (voir la Recommandation [UIT-T Y.4500.1]) dans l'extension subjectAltName du certificat. Un certificat de CSE-ID vérifie que l'entité présentant le certificat a été assignée à un CSE-ID particulier.

Utilisation: un certificat de CSE-ID ne peut être utilisé que pour authentifier une CSE.

Certificats d'AE-ID:

Description: ces certificats sont reliés, par une chaîne de certificats, à une ancre de confiance et ils contiennent la représentation complète par URI d'un AE-ID dans l'extension subjectAltName du certificat. Un certificat d'AE-ID vérifie que l'entité présentant le certificat a été assignée à un AE-ID particulier.

Utilisation: un certificat d'AE-ID ne peut être utilisé que pour authentifier une AE.

Certificats de FQDN:

Description: ces certificats sont reliés, par une chaîne de certificats, à une ancre de confiance et ils contiennent le FQDN d'une fonction MEF dans l'extension subjectAltName du certificat. Un certificat de FQDN vérifie que l'entité présentant le certificat a été assignée à un nom de domaine complet particulier.

Utilisation: un certificat de FQDN est utilisé pour authentifier une fonction d'inscription M2M auprès d'une entité inscrite pendant la phase de prise de contact d'inscription d'amorçage d'un cadre de configuration à distance de la sécurité fondé sur des certificats.

NOTE – Les versions et les détails spécifiques à chacune sont abordés dans le but de prendre en charge une variété de modèles de déploiement, tout en assurant que les entités oneM2M bénéficient de procédures claires pour authentifier les autres entités oneM2M au moyen de certificats.

Les profils de ces certificats sont détaillés au § 10.1.1 "Profils de certificats".

8.1.2.2 Validation du chemin de certification et vérification du statut d'un certificat

Si une entité doit authentifier une autre entité au moyen d'un certificat de dispositif, d'un certificat de CSE-ID, d'un certificat d'AE-ID ou d'un certificat de FQDN, la première entité doit procéder à une validation basique du chemin de certification (voir le § 6.1 de [IETF RFC 5280]) dans le cadre de la vérification du certificat de l'autre entité (voir le § 8.1.2.5 "Vérification d'un certificat").

Les certificats de CA doivent comprendre les extensions de contraintes de nom (§ 4.2.1.10 "Contraintes de nom" du document [IETF RFC 5280]) et doivent contraindre les noms (identificateur de dispositif M2M fondé sur un identificateur d'objet selon l'Annexe H "Identificateur de dispositif M2M fondé sur un identificateur d'objet" [UIT-T Y.4500.1], représentation du CSE-ID sous forme d'un nom de domaine public, de l'AE-ID absolu ou FQDN) susceptibles de figurer dans le certificat subséquent utilisé pour authentifier l'entité (respectivement le certificat de dispositif, le certificat de CSE-ID, le certificat d'AE-ID ou le certificat de FQDN).

Le paragraphe 4.2.1.10 "Contraintes de nom" de [IETF RFC 5280] décrit comment utiliser l'extension de contrainte de nom pour contraindre les URI et les FQDN.

Le paragraphe 10.4.1.4.2 "Profil pour l'autorité de certificat en vue des certificats de dispositifs" décrit la façon dont est utilisée l'extension de contrainte de nom pour contraindre les identificateurs de dispositifs M2M fondés sur des identificateurs d'objet.

Les informations relatives à l'ancre de confiance (§ 6.1.1 de [IETF RFC 5280]) sont fournies à l'entité pendant la configuration des justificatifs d'identité, la configuration de l'association, la configuration des justificatifs d'identité d'amorçage ou la configuration des instructions d'amorçage.

NOTE 1 – Le paragraphe 6.1.1 de [IETF RFC 5280] déclare que "les informations relatives à l'ancre de confiance sont fiables car elles ont été transmises à la procédure de traitement du chemin par une procédure hors bande digne de confiance". La configuration des justificatifs d'identité, la configuration de l'association, la configuration des justificatifs d'identité d'amorçage et la configuration des instructions d'amorçage sont des procédures hors bande digne de confiance.

Vérification du statut des certificats: dans le cas où une entité du domaine de l'infrastructure reçoit un certificat MEF, l'entité doit vérifier le statut du certificat en utilisant une liste de révocation de certificat, comme décrit dans [IETF RFC 5280]. Il est possible d'utiliser un mappage du protocole OCSP (*Online Certificate Status Protocol*) sur HTTP, comme décrit à l'Annexe A de [IETF RFC 6960], toutefois le mappage d'OCSP sur CoAP n'est pas défini actuellement. De surcroît, le protocole OCSP peut ne pas être facilement applicable dans tous les environnements. Une autre approche peut consister à utiliser l'extension de demande de statut de certificat TLS (paragraphe 8 de [IETF RFC 6066]; méthode également appelée "agrafage OCSP"), ou, mieux encore, l'extension de statut de certificat multiple ([IETF RFC 6961]), lorsqu'elle est disponible.

NOTE 2 – La majeure partie de ce qui précède s'appuie sur le texte quasi identique de la spécification du protocole CoAP [b-IETF RFC 7252], lequel contient des considérations similaires, voire identiques, relatives aux déploiements oneM2M.

8.1.2.3 Configuration des justificatifs d'identité pour le cadre de sécurité fondé sur des certificats

Si une entité doit s'authentifier à l'aide d'un cadre de sécurité fondé sur des certificats, elle doit être munie au préalable des informations suivantes:

– la clé de signature privée de l'entité;

NOTE – Une entité s'authentifie auprès d'autres entités en prouvant qu'elle connaît la clé de signature privée correspondant à une clé de vérification publique particulière.

– le certificat de l'entité (et, le cas échéant, la chaîne de certificats), comme décrit au § 10.1.1 "Profils de certificat".

Dans le cas d'un certificat de CSE-ID, l'entité doit être configurée avec le CSE-ID de l'entité.

Dans le cas d'un certificat d'AE-ID, l'entité doit être configurée avec l'AE-ID de l'entité.

8.1.2.4 Informations requises pour l'authentification par certificat d'une autre entité

L'entité A doit être configurée pour faire confiance aux informations suivantes afin d'authentifier l'entité B au moyen d'un cadre SAEF fondé sur des certificats:

Une indication de la variante de certificat de clé publique du certificat de l'autre entité B (c'est-à-dire, s'il s'agit d'un certificat de clé publique brute, d'un certificat de dispositif, d'un certificat de CSE-ID ou d'un certificat de FQDN).

Dans le cas où le certificat de l'entité B est un certificat de clé publique brute:

Un identificateur de clé publique pour la clé publique brute que contient le certificat (voir le § 10.1.2 "Identificateur de clé publique").

Dans le cas où le certificat de l'entité B est un certificat de dispositif, un certificat de CSE-ID ou un certificat de FQDN:

Un identificateur unique mondialement: l'identificateur unique mondialement de l'entité qui est également présent dans l'extension subjectAltName du certificat de l'autre entité:

Certificat de dispositif: identificateur d'instance matérielle unique mondialement (tel qu'un identificateur de dispositif M2M fondé sur un identificateur d'objet selon l'Annexe H "Identificateur de dispositif M2M fondé sur un identificateur d'objet" [UIT-T Y.4500.1]) qui est présent dans le certificat de dispositif.

Certificat de CSE-ID: représentation du CSE-ID sous forme d'un nom de domaine public, comme défini dans la Recommandation [UIT-T Y.4500.1].

Informations relatives à l'ancre de confiance: pour les certificats de l'ancre de confiance de la chaîne de certificats de l'entité B (voir le § 8.1.2.2 "Validation du chemin et vérification du statut des certificats").

L'entité B doit être configurée pour faire confiance aux informations suivantes afin d'authentifier l'entité A au moyen du cadre SAEF fondé sur des certificats:

Une indication de la variante de certificat de clé publique du certificat de l'entité A (c'est-à-dire, s'il s'agit d'un certificat de clé publique brute, d'un certificat de dispositif, d'un certificat de CSE-ID ou d'un certificat de AE-ID).

Dans le cas où le certificat de l'entité A est un certificat de clé publique brute:

Un identificateur de clé publique pour la clé publique brute que contient le certificat (voir le § 10.1.2 "Identificateur de clé publique").

Dans le cas où le certificat de l'entité A est un certificat de dispositif, un certificat de CSE-ID ou un certificat d'AE-ID:

Informations relatives à l'ancre de confiance: pour les certificats de l'ancre de confiance de la chaîne de certificats de l'entité A (voir le § 8.1.2.2 "Validation du chemin et vérification du statut des certificats").

Pour pouvoir authentifier la fonction d'inscription M2M au moyen d'un cadre RSPF fondé sur des certificats, une entité inscrite doit être configurée pour faire confiance aux informations de l'ancre de confiance de la chaîne de certificats de la fonction d'inscription M2M.

Une fonction d'inscription M2M doit être configurée pour faire confiance aux informations suivantes afin d'authentifier l'entité inscrite au moyen du cadre RSPF fondé sur des certificats:

Une indication de la variante de certificat de clé publique du certificat de l'entité B (c'est-à-dire, s'il s'agit d'un certificat de clé publique brute ou d'un certificat de dispositif).

Dans le cas où le certificat de l'entité inscrite est un certificat de clé publique brute:

Un identificateur de clé publique pour la clé publique brute que contient le certificat (voir le § 10.1.2 "Identificateur de clé publique").

Dans le cas où le certificat de l'entité inscrite est un certificat de dispositif, un certificat de CSE-ID ou un certificat d'AE-ID:

Identificateur unique mondialement: l'identificateur unique mondialement de l'entité qui est également présent dans l'extension subjectAltName du certificat de l'entité inscrite:

Certificat de dispositif: identificateur d'instance matérielle unique mondialement (tel qu'un identificateur de dispositif M2M fondé sur un identificateur d'objet selon l'Annexe H "Identificateur de dispositif M2M fondé sur un identificateur d'objet" [UIT-T Y.4500.1]) qui est présent dans le certificat de dispositif.

Certificat de CSE-ID: représentation du CSE-ID sous forme d'un nom de domaine public, comme défini dans la Recommandation [UIT-T Y.4500.1].

Certificat d'AE-ID: AE-ID absolu attribué à l'AE.

Informations relatives à l'ancre de confiance: pour la certification de l'ancre de confiance de la chaîne de certificats de l'entité inscrite (voir le § 8.1.2.2 "Validation du chemin et vérification du statut des certificats").

8.1.2.5 Vérification des certificats

Ce paragraphe décrit la façon dont une entité authentifie l'autre entité lors de la prise de contact de sécurité d'un cadre de sécurité fondé sur des certificats.

Le certificat de l'autre entité est reçu pendant la prise de contact de sécurité.

La vérification du certificat de l'autre entité se déroule comme suit:

Si les informations de sécurité configurées pendant la configuration de l'association ou la configuration des instructions d'amorçage indiquent que le certificat de l'autre entité est un certificat de clé publique brute, l'entité vérifie que l'identificateur de la clé publique (reçu pendant la configuration de l'association ou la configuration des instructions d'amorçage) correspond bien au certificat de clé publique brute (reçu pendant la prise de contact de sécurité) en suivant la procédure décrite au § 10.1.2 "Identificateur de clés publiques".

Si les informations de certificat configurées pendant la configuration de l'association ou la configuration des instructions d'amorçage indiquent que le certificat de l'autre entité est un certificat de dispositif, un certificat de CSE-ID, un certificat d'AE-ID ou un certificat de FQDN, l'entité doit effectuer les vérifications suivantes:

L'entité doit rechercher une correspondance entre l'identificateur unique mondialement décrit au § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité" (reçu pendant la configuration de l'association ou la configuration des instructions d'amorçage) et les valeurs dans l'extension subjectAltName du certificat de l'autre entité (reçu pendant la prise de contact de sécurité). En l'absence de correspondance exacte, l'entité doit abandonner la prise de contact (D)TLS.

Dans le cas d'un certificat de dispositif, l'identificateur unique mondialement est un identificateur d'instance matérielle unique à l'échelle mondiale (tel que l'identificateur de dispositif M2M fondé sur un identificateur d'objet selon l'Annexe H "Identificateur de dispositif M2M fondé sur un identificateur d'objet" [UIT-T Y.4500.1]). Dans cas, la notion de "correspondance" dépend de la façon dont est représenté l'identificateur d'instance matérielle unique mondialement dans l'extension subjectAltName.

Dans le cas d'un certificat de CSE-ID, l'identificateur unique mondialement est la représentation du CSE-ID sous la forme d'un nom de domaine public, comme défini dans la Recommandation [UIT-T Y.4500.1], et on appellera correspondance une égalité exacte entre un FQDN dans l'extension subjectAltName du certificat de l'autre entité et la représentation du CSE-ID sous forme d'un nom de domaine public.

Dans le cas d'un certificat d'AE-ID, l'identificateur unique mondialement est l'AE-ID, et on appellera correspondance une égalité exacte entre un URI dans l'extension subjectAltName du certificat de l'autre entité et l'AE-ID absolu.

Dans le cas d'un certificat de FQDN, l'identificateur unique mondialement est le FDQN de la fonction d'authentification M2M ou de la fonction d'inscription M2M, et on appellera correspondance une égalité exacte entre un URI, un FQDN ou un dNSName dans l'extension subjectAltName du certificat de l'autre entité et le FDQN de la fonction d'authentification M2M ou de la fonction d'inscription M2M.

L'entité doit valider le chemin et vérifier le statut des certificats en utilisant le certificat de l'ancre de confiance, comme décrit au § 8.1.2.2 "Validation du chemin et vérification du statut des certificats"). En cas d'échec de cette vérification, l'entité doit abandonner la prise de contact (D)TLS.

NOTE – Après une prise de contact de sécurité réussie, au cours de laquelle l'autre entité fournit une chaîne de certificats, l'identité de l'autre entité (reçue pendant la configuration de l'association ou la configuration des instructions d'amorçage) peut être associée à des informations supplémentaires extraites de la chaîne de certificats de l'autre entité (par exemple le fabricant de l'autre entité, son propriétaire, ou encore des critères de conformité). Ces détails ne sont pas décrits dans la présente Recommandation.

8.1.3 Introduction générale au cadre de sécurité fondé sur une architecture d'amorçage générique (GBA)

Une architecture d'amorçage générique ou GBA est un cadre susceptible d'être utilisé pour la configuration à distance de la sécurité. La Figure 8.1.3-1 présente un cadre GBA.

On pourrait faire appel à une procédure GBA dans un scénario où le fournisseur de services M2M et les opérateurs de réseau sous-jacent se sont mis d'accord pour utiliser les justificatifs d'identité du réseau sous-jacent comme base de la sécurité entre un service d'application M2M/nœud intermédiaire et un nœud d'infrastructure (y compris dans le cas où le fournisseur de services M2M et l'opérateur d'un réseau sous-jacent sont dans les faits la même entité).

Il est important que cette fonction soit utilisée uniquement dans le cadre d'un accord approprié entre le fournisseur de services M2M et l'opérateur du réseau sous-jacent. Le texte normatif pour le cadre d'établissement d'association de sécurité fondé sur une GBA (§ 8.2.2.2) et le cadre d'amorçage de sécurité fondé sur une GBA (§ 8.3.2.2) suppose implicitement qu'un tel accord est déjà en place. La présente Recommandation étant une spécification technique, elle ne traite pas des détails d'un tel accord.

Une introduction générale aux architectures d'amorçage génériques est proposée dans le rapport technique oneM2M TR-0008 [b-oneM2M TR0008].

Une fois l'amorçage GBA effectué avec succès, le service d'application M2M/le nœud intermédiaire et la BSF partagent une association de sécurité constituée d'un identificateur de transaction d'amorçage (B-TID) et de données de clé (Ks d'amorçage GBA).

Cette association de sécurité peut être utilisée par le service d'application M2M/le nœud intermédiaire pour dériver les clés NAF (Ks_(ext/int)_NAF) partagées entre un service d'application M2M/nœud intermédiaire et un nœud d'infrastructure M2M ou une fonction d'authentification M2M.

Il existe deux modes de GBA: la GBA reposant sur des équipements mobiles (GBA_ME) et la GBA reposant sur des cartes UICC (GBA_U). Dans le cas de la GBA_ME, on génère une clé spécifique à la fonction d'application réseau: la clé Ks_NAF. Dans le cas de la GBA_U, on génère deux clés spécifiques à la fonction d'application réseau: Ks_ext_NAF (qui est mise à disposition sur l'équipement mobile) et Ks_int_NAF (qui reste à l'intérieur de la carte UICC).

Le mode GBA_U ne peut être exécuté que lorsque la carte UICC a connaissance de l'architecture GBA.

Le BSF détermine quel mode exécuter, en fonction des capacités de la carte UICC indiquées dans les réglages de sécurité utilisateur de la GBA (GUSS).

Le mode GBA_U est recommandé car il offre un niveau de sécurité plus élevée que le mode GBA_ME. La conséquence de cette recommandation est que l'entité, AE ou CSE, qui utilise des clés NAF basées sur la GBA_U devrait résider dans la carte UICC.

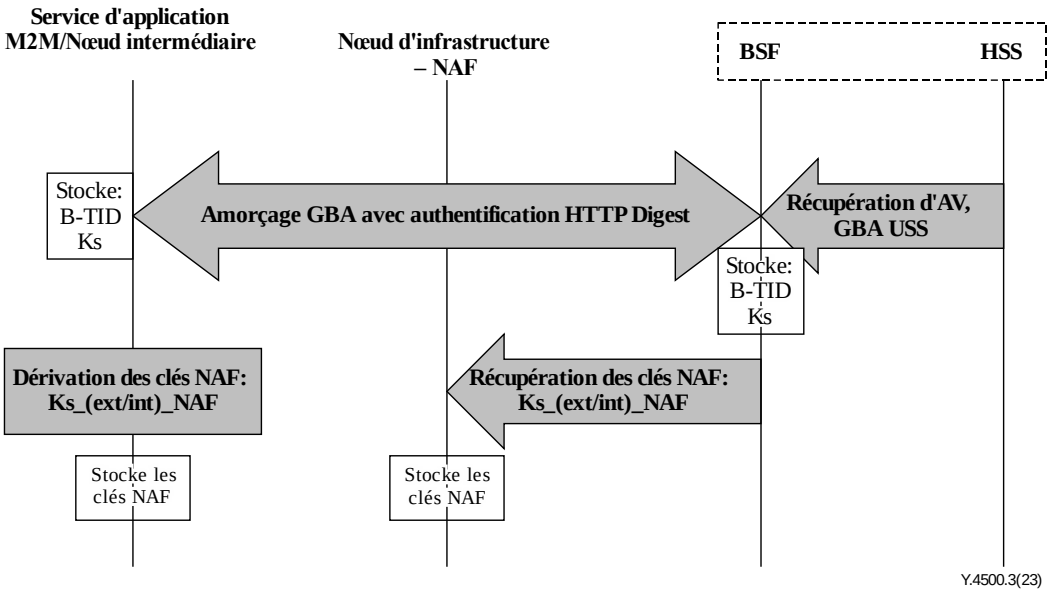


Figure 8.1.3-1 – Cadre GBA

Il est à noter que la fonction d'application réseau (NAF) peut être un nœud d'infrastructure ou une fonction d'authentification M2M.

8.2 Cadres d'établissement d'association de sécurité

8.2.1 Vue d'ensemble des cadres d'établissement d'association de sécurité

Les cadres d'établissement d'association de sécurité (SAEF) décrits dans la présente Recommandation s'appliquent aux connexions directes sur les points de référence Mcc, Mcc' ou Mca.

Les points d'extrémité de l'établissement d'association de sécurité sont désignés comme suit:

– L'entité A, qui peut être une AE ou une CSE. Cette entité agit toujours comme le client de l'association de sécurité (session TLS/DTLS).

– L'entité B, qui est obligatoirement une CSE. Cette entité agit toujours comme le serveur de l'association de sécurité (session TLS/DTLS).

Le système oneM2M prend en charge les cadres d'établissement d'association de sécurité suivants:

Cadres d'établissement d'association de sécurité

Établissement d'association de sécurité fondé sur la configuration d'une clé: une clé symétrique, nommée clé symétrique M2M et notée Kpsa, est configurée sur les entités. Les entités s'authentifient mutuellement en vérifiant les codes d'intégrité des messages (MIC) de la prise de contact de sécurité, codes qui ont été générés en utilisant la clé symétrique. Consulter le § 8.2.2.1 pour plus d'informations.

Établissement d'association de sécurité fondé sur des certificats: chaque entité reçoit:

– une clé de signature privée, connue uniquement de l'entité concernée;

– un certificat contenant la clé de vérification publique correspondante; et

– une chaîne de certificats reliant le certificat de l'entité à un certificat racine.

Chaque entité valide le certificat de l'autre avant de faire confiance aux clés de vérification publiques du certificat. Dans le cadre de la prise de contact de sécurité, l'entité A génère une signature numérique des paramètres de session en utilisant sa clé de signature privée et l'entité B vérifie la signature numérique en utilisant la clé de vérification publique de l'entité A. Les rôles sont ensuite

inversés: l'entité B génère une signature numérique, que vérifie l'entité A. Consulter le § 8.2.2.2 pour plus d'informations.

Établissement d'association de sécurité fondé sur une fonction d'authentification M2M (MAF): ce cadre d'établissement d'association de sécurité utilise l'authentification mutuelle de l'entité A et la fonction d'authentification M2M (MAF) pour générer une clé de connexion sécurisée M2M (Kc) que la fonction MAF transmet à l'entité B (via une communication distincte mutuellement authentifiée). Les entités s'authentifient alors mutuellement en utilisant la clé de connexion sécurisée M2M (Kc). Chacune de l'entité A et de l'entité B peut utiliser, au choix, les justificatifs d'identité de la clé symétrique ou les certificats pour s'authentifier mutuellement avec la fonction MAF.

Pour une description plus détaillée des cadres d'établissement d'association de sécurité ci-dessus, il est utile de comparer les aspects suivants de ces cadres:

Configuration des justificatifs d'identité: pour les cadres d'établissement d'associations de sécurité fondés sur la configuration d'une clé symétrique configurée, l'entité A et l'entité B reçoivent la clé symétrique M2M configurée, qu'elles utilisent ensuite pour s'authentifier mutuellement par le biais d'une préconfiguration ou d'une configuration à distance.

Pour les cadres d'établissement d'associations de sécurité fondés sur des certificats, l'entité A et l'entité B reçoivent au préalable le justificatif d'identité que l'entité utilise ensuite pour s'authentifier auprès de l'autre entité par le biais d'une préconfiguration ou d'une configuration à distance.

Pour les cadres d'établissement d'associations de sécurité fondés sur une fonction MAF, la procédure de configuration des justificatifs d'identité MAF (§ 8.8.3.1) est exécutée deux fois: une première fois pour configurer les justificatifs d'identité pour l'authentification mutuelle de l'entité A avec la fonction MAF, et une deuxième fois pour configurer les justificatifs d'identité pour l'authentification mutuelle de l'entité B avec la fonction MAF.

Configuration des identités: la configuration des identités peut avoir lieu en même temps que la configuration des justificatifs d'identité, ou ultérieurement.

Pour le cadre d'établissement d'association de sécurité fondé sur une fonction MAF, la fonction MAF est configurée avec des informations sur les identités de l'entité B et, éventuellement, de l'entité A. Le paragraphe 8.2.2.3 fournit des détails supplémentaires.

NOTE 1 – Les spécifications actuelles ne décrivent pas la façon dont ces informations sont configurées au niveau de la fonction MAF.

La connaissance par l'entité A de son identité (IdA) n'a aucun impact sur l'établissement de l'association de sécurité.

L'entité B doit être configurée avec son CSE-ID (IdB) avant la configuration de l'association.

Configuration de l'association: l'entité A doit recevoir l'IdB, le CSE-ID pour l'entité B.

NOTE 2 – La présente spécification ne décrit pas la façon dont l'entité A reçoit l'IdB. Des exemples de mécanismes pourraient inclure une configuration par gestion à distance et des mécanismes de découverte pris en charge par le(s) réseau(x) sous-jacent(s).

Dans le cas d'un cadre d'authentification fondé sur des certificats: chaque entité (entité A et entité B) est en outre configurée avec les informations de certificat qu'elle utilisera ensuite pour vérifier l'autre entité. Les informations de certificat nécessaires dépendent de la version des certificats. Pour plus de détails, consulter le § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité".

Dans le cas du cadre d'établissement d'association de sécurité fondé sur une fonction MAF:

La fonction MAF reçoit l'identité de l'entité B pour laquelle la MAF est autorisée à établir une association de sécurité avec l'entité A.

L'entité A et la fonction MAF interagissent, en utilisant la procédure d'enregistrement de clé MAF (§ 8.8.2.7) pour générer la clé de connexion sécurisée M2M (Kc) et l'identificateur de clé de connexion sécurisée M2M (KcID) et autoriser l'entité A à établir une association de sécurité avec l'entité B. Cette étape comprend l'authentification mutuelle à l'aide de la procédure de prise de contact MAF (§ 8.8.2.2). Cette étape comprend la fourniture, par l'entité A, de l'IdB à la fonction MAF. Voir la Note 2 ci-dessus.

Prise de contact de sécurité de l'association: identification, authentification et établissement d'un contexte de sécurité.

Dans le cas du cadre d'établissement d'association de sécurité fondé sur une fonction MAF:

L'entité A fournit l'identificateur de clé de connexion sécurisée M2M (KcID) à l'entité B.

L'entité B et la fonction MAF interagissent au moyen de la procédure de récupération de clé MAF (§ 8.8.2.8). Cette étape comprend l'authentification mutuelle via la prise de contact MAF (§ 8.8.2.2). L'entité B transfère le KcID à la MAF puis, si l'entité B est autorisée, la fonction MAF envoie en retour la clé de connexion sécurisée M2M (Kc) et soit l'IdA, soit un identificateur unique mondialement pour le justificatif d'identité utilisé par la MAF pour authentifier l'entité A pendant la configuration de l'association.

La clé de connexion sécurisée M2M (Kc) est ensuite utilisée dans la prise de contact de sécurité en vue de l'authentification mutuelle entre l'entité A et l'entité B.

L'entité A associe le contexte de sécurité résultant à l'IdB, à savoir l'AE-ID ou le CSE-ID pour l'entité B établi pendant la configuration de l'association.

L'entité B associe le contexte de sécurité à l'un des éléments suivants:

– un CSE-ID absolu, et une indication que l'entité A est une CSE;
– un AE-ID absolu, et une indication que l'entité A est une AE; ou
– une liste des valeurs des AE-ID absolus, et une indication que l'entité A est une AE. Ce cas de figure s'applique uniquement lorsque l'entité A présente un certificat de dispositif.

La présente Recommandation propose les approches suivantes pour que l'entité B détermine le CSE-ID ou le(s) AE-ID applicables avant l'enregistrement:

Si l'entité A est authentifiée au moyen d'un certificat de CSE-ID (ou un certificat d'AE-ID), l'entité B extrait le CSE-ID (ou, respectivement, l'AE-ID) du certificat et il associe le contexte de sécurité à ce CSE-ID (ou, respectivement, cet AE-ID), comme décrit dans le profil de certificat au § 10.1.1 "Profils de certificat".

Dans tous les autres cas, l'entité B forme un identificateur de justificatif d'identité unique mondialement (voir le § 10.4 "Détails concernant l'identificateur de justificatif d'identité") qui identifie le justificatif d'identité utilisé par l'entité A dans le mécanisme d'établissement d'association de sécurité. L'identificateur de justificatif d'identité identifie soit une Kpsa (dans le cas d'un SAEF fondé sur une PSK), soit un certificat (dans le cas d'un SAEF fondé sur des certificats), soit une Km (dans le cas d'un SAEF fondé sur une fonction MAF). L'entité B détermine ensuite le CSE-ID ou le(s) AE-ID applicables à cet identificateur de justificatif d'identité.

Si l'entité B a attribué le ou les AE-ID correspondant à cet identificateur de justificatif d'identité, c'est à l'entité B qu'il incombe de déterminer les AE-ID correspondant à cet identificateur.

À défaut, le CSE-ID ou les AE-ID peuvent être mis à disposition de l'entité B de l'une des façons suivantes. Le fournisseur de service M2M doit s'assurer que l'une de ces approches fournira avec succès le CSE-ID ou les AE-ID de l'entité A.

– Si la procédure d'établissement de l'association de sécurité est facilitée par une fonction d'authentification M2M, celle-ci peut recevoir le CSE-ID ou l'AE-ID et le transmettre à l'entité B en même temps que la clé Kc. La fonction d'authentification M2M peut avoir reçu

le CSE-ID ou l'AE-ID pendant la configuration, y compris dans le cas où la fonction MAF reçoit le CSE-ID ou l'AE-ID pendant configuration à distance par une fonction MEF (qui est similaire au cas décrit au point suivant).

– Si la procédure d'établissement de l'association de sécurité utilise une clé symétrique configurée à distance sur l'entité A et l'entité B, la fonction d'inscription M2M peut fournir à l'entité B un CSE-ID ou un AE-ID pendant la procédure de configuration à distance de la sécurité.

– Si le fournisseur de services M2M attribue le ou les identificateurs d'entité de l'entité A, le CSE-ID ou le(s) AE-ID peuvent être configurés de manière sécurisée par le fournisseur de services M2M auprès de l'entité B avant la prise de contact de sécurité de l'association. Par exemple, le CSE-ID ou le(s) AE-ID peuvent être configurés pour faire partie de la configuration des justificatifs d'identité ou de la configuration de l'association. La présente spécification permet d'utiliser d'autres mécanismes, en supposant que le mécanisme assure l'authentification, la protection de l'intégrité et éventuellement la confidentialité.

EXEMPLE 1: Si le fournisseur de services M2M a l'opportunité de configurer l'entité B avant le déploiement, il pourrait en profiter pour configurer en même temps le CSE-ID ou le(s) AE-ID sur l'entité B.

EXEMPLE 2: Un protocole de gestion à distance sécurisé pourrait être utilisé pour configurer le CSE-ID ou le(s) AE-ID sur l'entité B. Cependant, il ne s'agit pas actuellement d'une fonctionnalité interopérable car il n'existe pas d'objet de gestion normalisé facilitant cette gestion.

Dans le cas où l'entité A est une AE et l'entité B une CSE, il est possible d'obtenir le(s) AE-ID applicables en récupérant les ressources <*serviceSubscribedAppRule*> applicables, le lien vers ces ressources étant assuré par l'attribut *ruleLinks* du <*serviceSubscribedNode*> de l'entité B sur la CSE du nœud d'infrastructure (IN-CSE), comme décrit au § 10.1.1.2.2 "Procédure d'enregistrement d'une entité d'application" dans la Recommandation [UIT-T Y.4500.1].

La Figure 8.2.1-1 résume les trois cadres d'établissement d'association de sécurité définis ci-dessus.

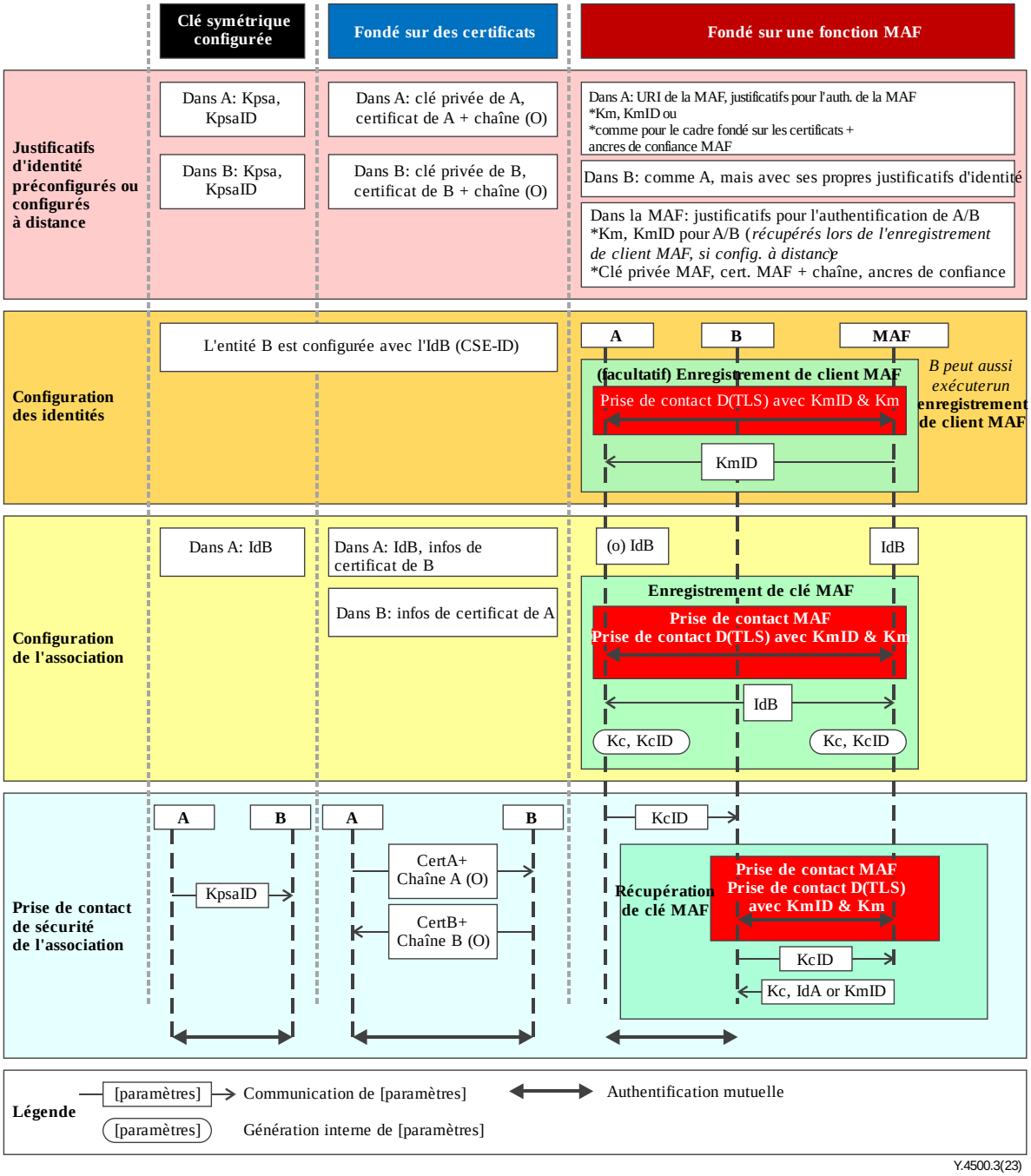


Figure 8.2.1-1 – Vue d'ensemble des cadres d'établissement d'association de sécurité pris en charge par oneM2M

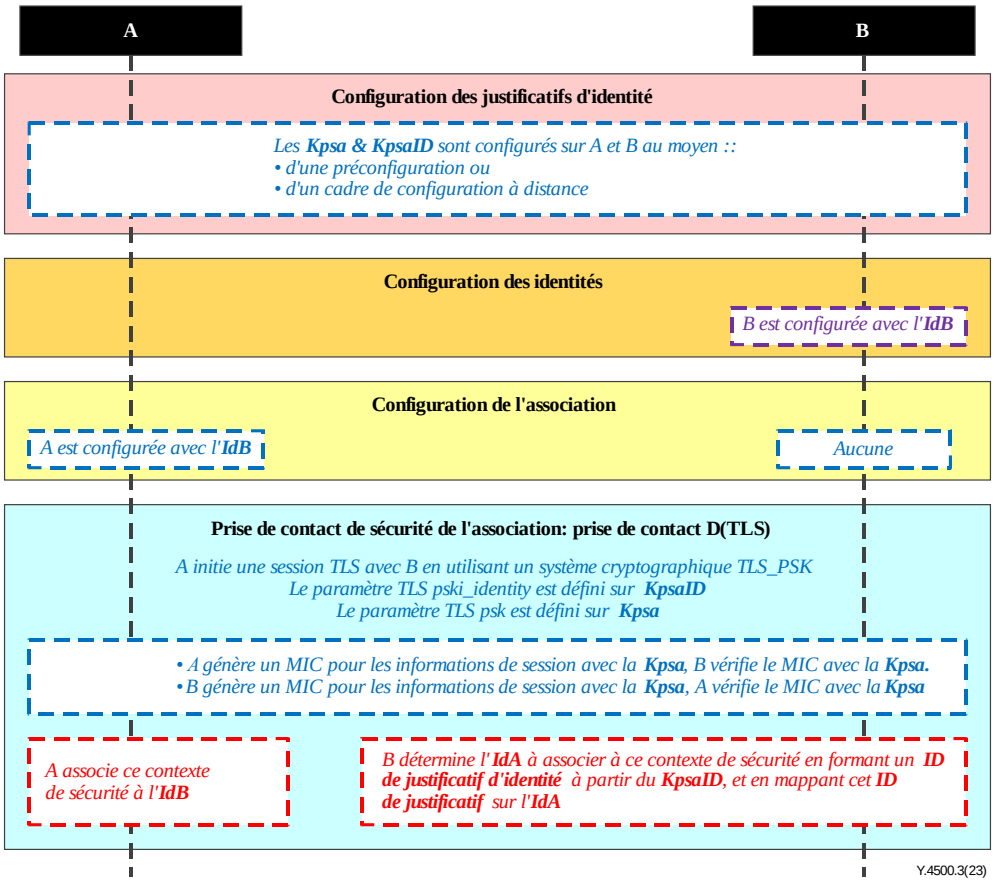
8.2.2 Présentation détaillée des cadres d'établissement d'associations de sécurité

8.2.2.1 Cadres d'établissement d'associations de sécurité fondés sur une clé symétrique configurée

Ce paragraphe décrit le cadre d'établissement d'association de sécurité fondé sur la configuration d'une clé symétrique. Ce cadre permet l'authentification mutuelle de deux entités, qui peuvent être deux CSE, ou une CSE et une AE. Le justificatif d'identité de ce cadre est une clé symétrique à longue durée de validité, qui a été configurée sur les entités à authentifier. Cette clé, notée Kpsa, est appelée clé de connexion sécurisée configurée. La Kpsa peut être fournie aux entités par le biais d'une préconfiguration ou d'une configuration à distance grâce aux cadres de configuration à distance de la sécurité décrits au § 8.3. Les entités s'authentifient mutuellement en vérifiant les codes

d'authentification des messages de la prise de contact de sécurité de l'association, codes qui ont été générés en utilisant la clé de connexion sécurisée configurée.

La Figure 8.2.2.1-1 représente la séquence d'événements survenant lors de l'utilisation du cadre d'établissement d'association de sécurité fondé sur une clé symétrique configurée. Dans cette figure, "Entité A" et "Entité B" représentent respectivement deux CSE, ou bien une CSE et une AE, ou encore une AE et CSE.



NOTE – Les couleurs de police suivantes permettent d'identifier les différents sujets auquel se rapporte le texte:
Le texte en italique bleu souligne les détails spécifiques à ce cadre particulier d'établissement d'une association de sécurité.
Le texte en italique violet souligne les actions techniques qui peuvent comprendre des étapes non décrites par la spécification oneM2M.
Le texte en italique rouge souligne les propriétés relatives à la sécurité.

Figure 8.2.2.1-1 – Séquence d'événements liés à l'utilisation du cadre d'établissement d'association de sécurité fondé sur une clé symétrique configurée

Configuration des justificatifs d'identité: la clé de connexion sécurisée configurée (Kpsa) et l'identificateur de clé de connexion sécurisée configurée correspondant, noté KpsaID, sont fournis aux deux entités via une préconfiguration ou une configuration à distance. Le format du KpsaID est défini au § 10.5 "KpsaID". Si l'entité A est une CSE, elle doit également être configurée avec le CSE-ID correspondant (événement non représenté sur la figure).

Configuration des identités: voir le § 8.2.1

Configuration de l'association: l'entité A doit être configurée avec l'identité de l'entité B (IdB) avant la prise de contact de sécurité de l'association. L'entité A doit utiliser cette identité pour authentifier l'entité B au moyen des arguments ci-dessus. Cette identité est également utilisée pour acheminer l'échange (D)TLS. L'entité A doit associer l'identité de l'entité B à des messages sécurisés dans des contextes de sécurité établis en utilisant la clé de connexion sécurisée configurée (Kpsa) associée à l'identificateur de clé de connexion sécurisée configurée (KpsaID).

Si l'entité A est une CSE, il faut configurer le CSE-ID de cette entité A sur l'entité B avant d'initier la prise de contact de sécurité de l'association. Si l'entité A est une AE, l'entité B peut soit être configurée avec l'identité de l'entité A (IdA) avant la prise de contact de sécurité de l'association, ou bien elle peut déterminer l'IdA pendant l'enregistrement (création de la ressource <AE>). L'entité B doit utiliser cette identité pour authentifier l'entité A au moyen des arguments ci-dessus. L'entité B doit associer l'identité de l'entité A configurée à des messages sécurisés dans des contextes de sécurité établis en utilisant la clé de connexion sécurisée configurée (Kpsa) associée à l'identificateur de clé de connexion sécurisée configurée (KpsaID).

Prise de contact de sécurité de l'association: les entités doivent effectuer une prise de contact (D)TLS-PSK [IETF RFC 4279] pour établir une session sécurisée.

Le paramètre "psk_identity" [IETF RFC 4279] doit être défini sur la valeur de l'identificateur de la clé de connexion sécurisée configurée KpsaID.

Les entités définissent le paramètre "psk" [IETF RFC 4279] sur la valeur de la clé de connexion sécurisée configurée Kpsa.

Le système cryptographique (D)TLS pour le cadre d'établissement d'association de sécurité fondé sur une clé de connexion sécurisée configurée doit être conforme au § 10.2.2.

Lorsque l'entité B a été authentifiée avec succès, l'entité A doit associer le contexte de sécurité à l'IdB (identificateur de l'entité B) configuré sur l'entité A pendant la configuration de l'association.

Lorsque l'entité A a été authentifiée avec succès, l'entité B doit associer le contexte de sécurité à un CSE-ID ou un AE-ID:

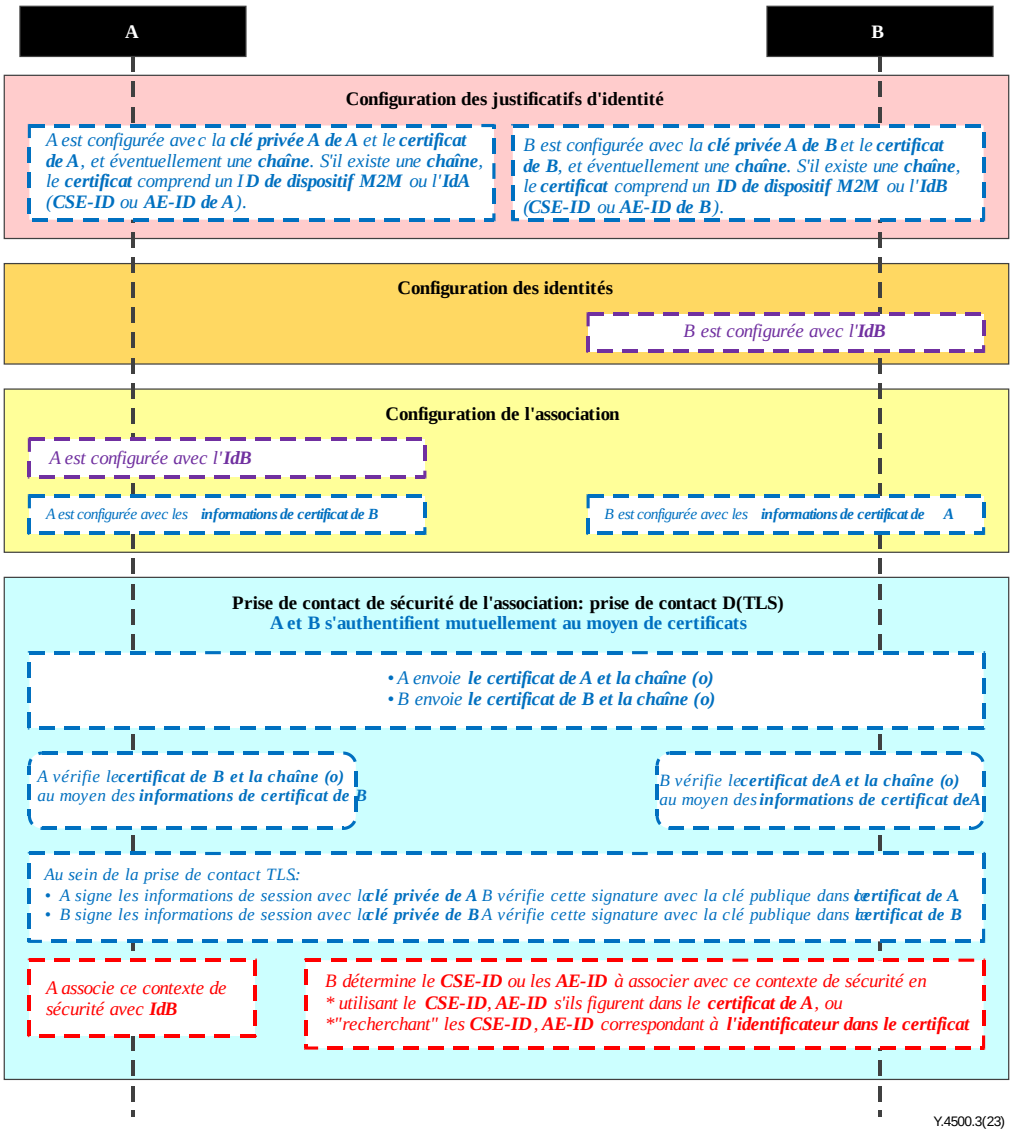
– Si l'entité B a déjà reçu le CSE-ID ou l'AE-ID correspondant au KpsaID, l'entité B doit associer le contexte de sécurité à ce CSE-ID ou AE-ID.

Sinon, l'entité B associe le contexte de sécurité à l'identificateur de justificatif d'identité formé à partir du KpsaID, comme décrit au § 10.4 "Détails concernant l'identificateur de justificatif d'identité ". L'entité B doit ensuite déterminer le CSE-ID ou l'AE-ID à partir de cet identificateur de justificatif d'identité, comme décrit au § 8.2.1 "Vue d'ensemble des cadres d'établissement d'association de sécurité".

8.2.2.2 Cadres d'établissement d'association de sécurité fondés sur des certificats

Ce paragraphe décrit le cadre d'établissement d'association de sécurité fondé sur des certificats.

La Figure 8.2.2.2-1 représente la séquence d'événements survenant lors de l'utilisation du cadre d'établissement d'association de sécurité fondé sur des certificats. Dans cette figure, "Entité A" et "Entité B" représentent respectivement deux CSE, ou bien une CSE et une AE.



NOTE – Les couleurs de police suivantes permettent d'identifier les différents sujets auquel se rapporte le texte:
Le texte en italique bleu souligne les détails spécifiques à ce cadre particulier d'établissement d'association de sécurité.
Le texte en italique violet souligne les actions techniques qui peuvent comprendre des étapes non décrites par la spécification oneM2M.
Le texte en italique rouge souligne les propriétés relatives à la sécurité.

Figure 8.2.2.2-1 – Séquence d'événements liés à l'utilisation du cadre d'établissement d'une association de sécurité fondé sur des certificats

Configuration des justificatifs d'identité: les clés privées et certificats pour chaque entité doivent être préconfigurés comme décrit au § 8.1.2.3 "Configuration des justificatifs d'identité pour le cadre de sécurité fondé sur des certificats". Si l'entité A est une CSE, elle A doit également être configurée avec le CSE-ID correspondant (événement non représenté sur la figure).

Configuration des identités: voir le § 8.2.31

Configuration de l'association: il faut configurer, sur l'entité A et l'entité B, les informations nécessaires pour l'authentification et l'identification (lors de la prise de contact de sécurité de l'association) de l'entité B et de l'entité A, respectivement.

Les informations configurée sur l'entité A doivent inclure les arguments suivants:

– les informations de certificat de l'entité B, comme décrit au § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité";

– l'identité de l'entité B (IdB). L'entité A doit utiliser cette identité pour authentifier l'entité B au moyen des arguments ci-dessus. Ces informations sont utilisées pour acheminer l'échange (D)TLS.

NOTE – L'entité A associe l'identité de l'entité B aux messages sécurisés au sein des contextes de sécurité établis conformément aux informations de certificat de l'entité B configurées.

Les informations configurées sur l'entité B doivent inclure les arguments suivants:

les informations de certificat de l'entité A, comme décrit au § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité".

Prise de contact de sécurité de l'association

Chaque entité doit vérifier le certificat de l'autre, comme décrit au § 8.1.2.5 "Vérification des certificats".

Les entités doivent s'authentifier l'une l'autre en utilisant les certificats validés, comme décrit dans les spécifications TLS 1.2 [IETF RFC 5246] et DTLS 1.2 [IETF RFC 6347].

Le système cryptographique (D)TLS pour le cadre d'établissement d'association de sécurité fondé sur des certificats doit être conforme au § 10.2.3.

Lorsque l'entité B a été authentifiée avec succès, l'entité A doit associer le contexte de sécurité à l'IdB (identificateur de l'entité B) configuré sur l'entité A pendant la configuration de l'association.

Lorsque l'entité A a été authentifiée avec succès, l'entité B doit associer le contexte de sécurité à un CSE-ID, un AE-ID ou une liste des AE-ID autorisés:

– Si l'entité A établit un contexte de sécurité en présentant un certificat de CSE-ID, l'entité B doit associer le contexte de sécurité au CSE-ID indiqué dans le certificat.
– Si l'entité A établit un contexte de sécurité en présentant un certificat d'AE-ID, l'entité B doit associer le contexte de sécurité à l'AE-ID absolu indiqué dans le certificat.
– Si l'entité A établit un contexte de sécurité en présentant un certificat de dispositif, l'entité B doit associer le contexte de sécurité à l'identificateur de justificatif d'identité formé à partir de l'identificateur d'instance matérielle unique mondialement indiqué dans le certificat, comme décrit au § 10.4 "Détails concernant les justificatifs d'identité". L'entité B doit ensuite utiliser cet identificateur de justificatif d'identité pour déterminer le CSE-ID, l'AE-ID ou la liste des AE-ID autorisés, comme décrit au § 8.2.1 "Vue d'ensemble des cadres d'établissement d'association de sécurité".
– Si l'entité A établit un contexte de sécurité en présentant un certificat de clé publique brute, l'entité B doit associer le contexte de sécurité à l'identificateur de justificatif d'identité formé à partir de l'identificateur de clé publique correspondant, comme décrit au § 10.1.2 "Identificateur de clés publiques". L'entité B doit alors utiliser cet identificateur de justificatif d'identité pour déterminer le CSE-ID ou l'AE-ID, comme décrit au § 8.2.1 "Vue d'ensemble des cadres d'établissement d'association de sécurité".

8.2.2.3 Cadres d'établissement d'associations de sécurité à clé symétrique fondés sur une fonction MAF

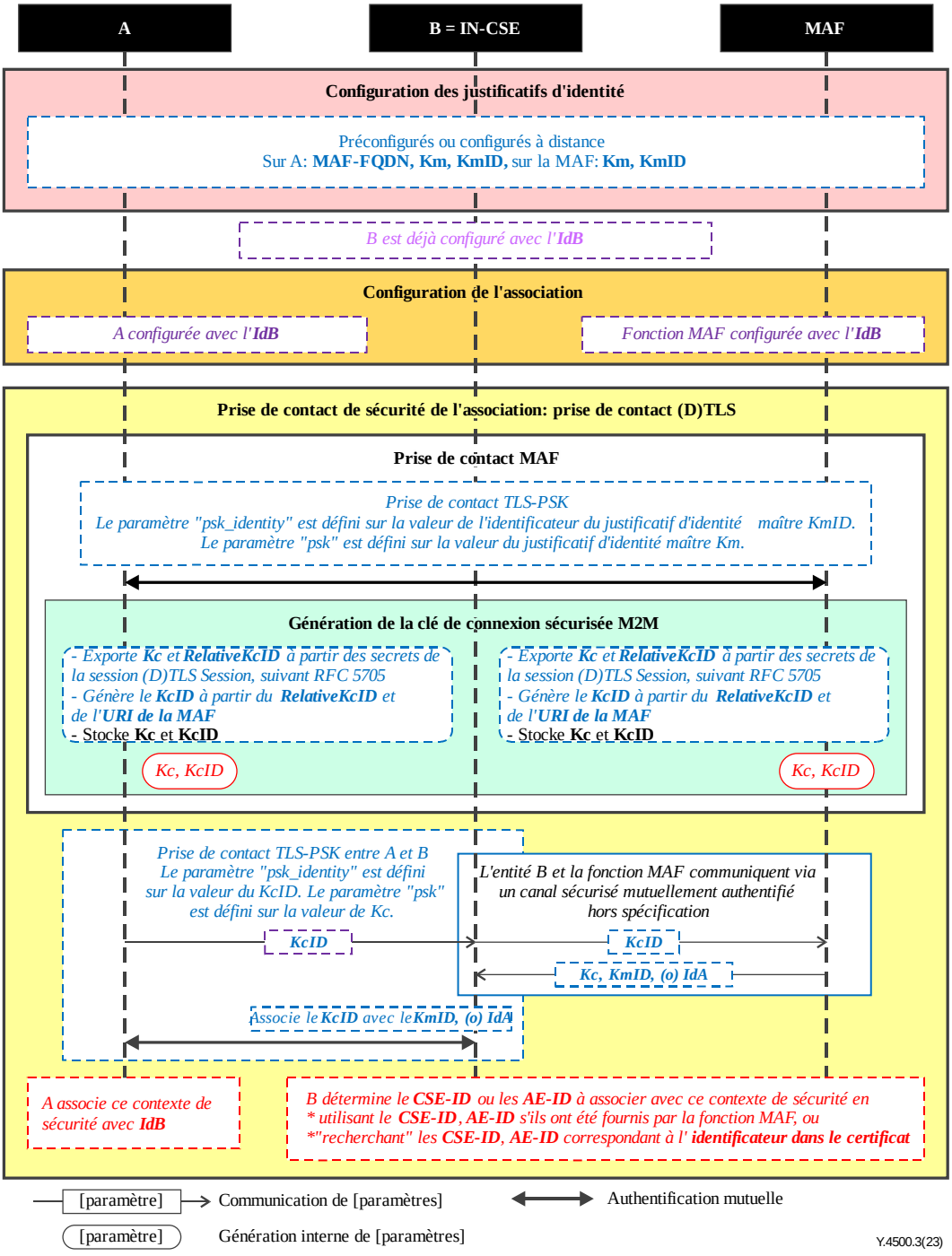
Ce paragraphe décrit le cadre d'établissement d'association de sécurité fondé sur une fonction MAF.

Un tel cadre utilise les procédures du cadre de sécurité MAF du § 8.8, en appliquant les correspondances suivantes pour les rôles fonctionnels:

– L'entité A joue le rôle du point d'extrémité source.
– L'entité B joue le rôle du point d'extrémité cible.

Le présent paragraphe désigne les entités uniquement par les noms d'entité A et d'entité B.

La Figure 8.2.2.3-1 représente la séquence d'événements survenant lors de l'utilisation du cadre d'établissement d'association de sécurité fondé sur une fonction MAF.



NOTE 1 – Les couleurs de police suivantes permettent d'identifier les différents sujets auquel se rapporte le texte:
Le texte en italique bleu souligne les détails spécifiques à ce cadre particulier d'établissement d'association de sécurité.
Le texte en italique violet souligne les actions techniques qui peuvent comprendre des étapes non décrites par la spécification oneM2M.
Le texte en italique rouge souligne les propriétés relatives à la sécurité.

Figure 8.2.2.3-1 – Séquence d'événements liés à l'utilisation du cadre d'établissement d'une association de sécurité fondé sur une fonction MAF

Configuration des justificatifs d'identité: l'entité A (et, respectivement, l'entité B) doit être configurée de manière individuelle avec les justificatifs d'identité permettant l'authentification mutuelle avec la fonction MAF, comme décrit dans le paragraphe relatif à la configuration des justificatifs d'identité MAF (§ 8.8.3.1). Il est possible, pour ce faire, de procéder par préconfiguration ou par configuration à distance. Dans le cas d'une configuration à distance de clés symétriques, la fonction MAF obtient les clés symétriques de la fonction MEF au cours de la procédure d'enregistrement de client MAF, lors de la configuration des identités.

Configuration des identités: la fonction MAF est censée être autorisée à fournir des services à l'entité A et à l'entité B.

NOTE 1 – Les spécifications oneM2M actuelles ne décrivent pas la façon dont cette autorisation est octroyée à la fonction MAF.

La fonction MAF est configurée avec des informations d'identité sur l'entité B et, éventuellement, sur l'entité A:

Si l'entité A est une CSE, la fonction MAF est censée être configurée avec le CSE-ID de l'entité A (noté IdA).

Si l'entité A est une AE, la fonction MAF est censée être configurée avec l'AE-ID de l'entité A (noté IdA).

La fonction MAF est censée être configurée avec le CSE-ID attribué par le fournisseur de services M2M de l'entité B (noté IdB).

NOTE 2 – Les spécifications oneM2M actuelles ne décrivent pas la façon dont ces informations sont configurées au niveau de la fonction MAF.

Si l'entité A (ou, respectivement, l'entité B) reçoit à distance une clé symétrique destinée à être utilisée avec la fonction MAF, l'entité A (ou, respectivement, l'entité B) doit exécuter à titre individuel la procédure d'enregistrement de client MAF (§ 8.8.2.3) auprès de la fonction MAF. Cette procédure indique à la fonction MAF (a) de récupérer la Km auprès de la fonction MEF, et (b) de fournir au point d'extrémité le KmID à utiliser pour procéder ensuite à l'authentification avec la fonction MAF à l'étape 5.

Configuration de l'association: il faut configurer, sur l'entité A et au niveau de la fonction B, les informations nécessaires pour l'authentification et l'identification lors de la prise de contact MAF et de la prise de contact de sécurité de l'association.

Autorisation du cadre SAEF: l'entité A doit recevoir l'IdB, c'est-à-dire le CSE-ID pour l'entité B. Voir la Note 3 au § 8.2.1.

La fonction MAF est censée être configurée avec l'identité de l'Entité B (IdB) pour laquelle la fonction est autorisée à fournir une clé Kc pour un cadre SAEF avec l'entité A.

L'entité A et la fonction MAF doivent établir un canal de communication sécurisé mutuellement authentifié au moyen de la procédure de prise de contact MAF (§ 8.8.2.2), en utilisant les justificatifs fournis pendant la configuration des justificatifs d'identité.

L'entité A doit initier la procédure d'enregistrement de clé MAF (§ 8.8.2.7) auprès de la fonction MAF. L'enregistrement de clé MAF doit comprendre l'identificateur d'utilisation de sécurité (SUID) associé au cadre SAEF fondé sur la fonction MAF et l'IdB. Cette procédure produit les effets suivants:

L'entité A et la fonction MAF établissent une clé de connexion sécurisée M2M (Kc) et l'identificateur associé (KcID), lesquels correspondent à la clé symétrique de sortie et à l'identificateur de clé établi par la procédure d'enregistrement de clé MAF.

La fonction MAF indique la durée de vie de la clé de connexion sécurisée M2M (Kc).

Le SUID limite le périmètre au sein duquel l'utilisation de la clé Kc est autorisée. Dans le cas présent, le SUID permet de s'assurer que l'entité A ne peut utiliser la clé Kc que dans le cadre SAEF fondé sur la fonction MAF.

Prise de contact de sécurité de l'association: l'entité A doit initier une prise de contact (D)TLS-PSK avec l'entité B, conformément au § 10.2.2.

L'entité A doit envoyer le KcID à l'entité B (nœud d'infrastructure) sous la forme du paramètre "psk_identity" dans une prise de contact (D)TLS-PSK.

L'entité B reconnaît la partie MAF-FQDN du KcID dans le paramètre "psk_identity", et elle détermine que la clé de connexion sécurisée M2M (Kc) correspondante doit être récupérée auprès de la fonction MAF correspondante. L'entité B doit définir le RelativeKeyID comme étant la partie relative du KcID.

L'entité B et la fonction MAF doivent exécuter la procédure de récupération de clé MAF décrite au § 8.8.2.8.

NOTE 3 – La procédure de récupération de clé MAF comprend l'établissement d'un canal de communication sécurisé mutuellement authentifié au moyen de la procédure de prise de contact MAF (décrite au § 8.8.2.2), en utilisant les justificatifs fournis pendant la configuration des justificatifs d'identité.

L'entité B doit fournir l'identificateur RelativeKeyID à la fonction MAF. La fonction MAF envoie en retour à l'entité B la valeur de la clé symétrique de sortie, le délai d'expiration (expirationTime), l'identificateur d'utilisation de sécurité (SUID) et l'identité de l'entité A. La valeur de la clé Kc doit être définie sur la valeur de la clé symétrique de sortie. La durée de vie de la clé Kc doit être définie sur le délai expirationTime. Le SUID limite le périmètre au sein duquel la clé Kc sera utilisée. Dans le cas présent, le SUID permet de s'assurer que l'entité B ne peut utiliser la clé Kc que dans le cadre SAEF fondé sur la fonction MAF.

NOTE 4 – L'attribution de la durée de vie de la clé Kc incombe à la fonction MAF.

L'entité A et l'entité B doivent mener à bien la prise de contact (D)TLS-PSK avec le paramètre "psk" défini sur la clé de connexion sécurisée M2M (Kc).

Lorsque l'entité B a été authentifiée avec succès, l'entité A doit associer le contexte de sécurité à l'IdB (identificateur de l'entité B) configuré sur l'entité A pendant la configuration de l'association.

Lorsque l'entité A a été authentifiée avec succès, l'entité B doit associer le contexte de sécurité à un CSE-ID ou un AE-ID:

Si la fonction MAF a déjà fourni un CSE-ID ou un AE-ID à l'entité B, celle-ci doit associer le contexte de sécurité à ce CSE-ID ou AE-ID.

Sinon, l'entité B associe le contexte de sécurité à l'identificateur de justificatif d'identité formé à partir du KmID (fourni par la fonction MAF) comme décrit au § 10.4 "Détails concernant l'identificateur de justificatif d'identité". L'entité B doit ensuite déterminer le CSE-ID ou l'AE-ID à partir de l'identificateur de justificatif d'identité, comme décrit au § 8.2.1 "Vue d'ensemble des cadres d'établissement d'association de sécurité".

L'entité A et l'entité B peuvent établir à tout moment une nouvelle prise de contact (D)TLS-PSK fondée sur la clé Kc pendant la durée de vie de la clé Kc. À l'expiration de la clé Kc, l'entité B doit échouer à établir une prise de contact (D)TLS-PSK, ce qui lui indique qu'une nouvelle prise de contact MAF est requise.

8.3 Cadres de configuration à distance de la sécurité

8.3.1 Vue d'ensemble des cadres de configuration à distance de la sécurité

8.3.1.1 Objectif des cadres de configuration à distance de la sécurité

Les cadres de configuration à distance de la sécurité (RSPF) fournissent des justificatifs d'identité à une entité inscrite, laquelle est un principal de sécurité dans un nœud, une CSE ou une AE, dans le cadre de l'inscription de ladite entité auprès d'un fournisseur de services M2M ou d'un générateur de confiance M2M. La fonction MEF assure ses services pour le compte de *parties prenantes administratrices,* par exemples des fournisseurs de service M2M ou des générateurs de confiance M2M (MTE) tiers. Une partie prenante administratrice autorise le fournisseur de services MEF à fournir des services à des clients MEF, et supervise l'autorisation de la gestion des justificatifs d'identité.

Ces justificatifs peuvent être:

Une clé symétrique partagée par l'entité inscrite et une cible d'inscription, laquelle peut être une fonction MAF, un nœud, une CSE ou une AE.

Si la cible d'inscription est une fonction MAF, le justificatif d'identité peut être utilisé dans un cadre SAEF et des mécanismes de sécurité ESPrim et ESData fondés sur une fonction MAF, la clé symétrique configurée servant à l'authentification mutuelle de l'entité inscrite et de la fonction MAF.

Si la cible d'inscription est un nœud, une CSE ou une AE, le justificatif d'identité peut être utilisé pour une seule des trois options parmi le cadre SAEF et les protocoles ESPrim et ESData fondés sur une clé pré-partagée (PSK).

La clé symétrique configurée fournie aux fins de l'authentification mutuelle de l'entité inscrite et du nœud ou de la CSE ou de l'AE.

NOTE 1 – Ce scénario ne devrait être employé que dans les cas où l'entité inscrite est censée avoir besoin d'une clé symétrique avec un nombre réduit de CSE ou d'AE.

Le ou les certificats dont l'entité inscrite connaît la clé privée correspondante, et un ensemble d'ancres de confiance destinées à authentifier le fournisseur de services M2M ou la fonction MAF du générateur de confiance M2M (MTE) ou d'autres entités inscrites auprès du fournisseur de services M2M ou du MTE. Ces justificatifs d'identité peuvent servir à:

Sécuriser la communication directement avec d'autres nœuds, CSE ou AE au moyen d'un cadre SAEF fondé sur des certificats, d'un échange ESCertKE (établissement direct de clés de bout en bout au moyen de certificats), et des options de protection ESData fondés sur des certificats. Les autres nœuds, CSE ou AE s'authentifieraient au moyen de leurs propres certificats, reliés à un certificat configuré émanant d'une CA constituant une ancre de confiance, dans ces cadres de sécurité.

Un cadre SAEF fondé sur une fonction MAF et des options de protection ESPrim et ESData fondées sur une fonction MAF, le certificat étant utilisé pour authentifier l'entité inscrite auprès de la fonction MAF. La fonction MAF s'authentifierait au moyen de son propre certificat, relié à un certificat configuré émanant d'une CA constituant une ancre de confiance.

Les spécifications oneM2M prennent aussi en charge la configuration des justificatifs d'identité via les mécanismes de configuration des dispositifs abordés dans la Recommandation [UIT-T Y.4500.22] et la préconfiguration; c'est-à-dire par des moyens autres qu'un cadre de configuration à distance de la sécurité. La méthode de préconfiguration peut dépendre du déploiement. L'Annexe D décrit un cadre de préconfiguration interopérable reposant sur des cartes UICC.

NOTE 2 – Les cadres RSPF sont décrits de façon à offrir une interface interopérable permettant aux entités de terrain d'interagir avec une fonction MEF. L'utilisation, par les entités de terrain, des cadres RSPF décrits est recommandée car ces cadres ont été passés en revue par des experts en sécurité oneM2M. Les cadres RSPF peuvent également être utilisés par des entités du domaine de l'infrastructure (nœuds, AE, CSE et fonctions MAF) pour interagir avec une fonction MEF. Il est prévu que la fonction MEF puisse inclure des interfaces

"backend" (d'extrémité) supplémentaires, non abordées par la spécification oneM2M, pour la coordination de l'information avec les parties prenantes administratrices et les fournisseurs de services MAF.

8.3.1.2 Circulation de haut niveau

Un principal de sécurité situé sur un nœud, une AE ou une CSE, et qui nécessite d'être configuré à distance, est appelé *entité inscrite* ou *client MEF source*. Lors de la configuration d'une clé, les nœuds, les CSE, les AE ou la fonction d'authentification M2M avec lesquels l'entité inscrite doit établir la clé symétrique sont appelés *cible d'inscription* ou *client MEF cible*.

Le système oneM2M prend en charge les méthodes d'authentification suivantes pour les cadres de configuration à distance de la sécurité:

Cadre de configuration à distance de la sécurité fondé sur une clé symétrique d'entité inscrite préconfigurée: une clé symétrique est préconfigurée pour l'entité inscrite et la fonction d'inscription M2M en vue de l'authentification mutuelle de ces entités. Consulter le § 8.3.2.1 pour plus d'informations.

NOTE 1 – La version actuelle de la présente spécification prend en charge uniquement les clés symétriques préconfigurées. Il est prévu d'ajouter, dans les versions à venir, la prise en charge de l'authentification au moyen de clés symétriques configurées par une autre fonction MEF au moyen d'un cadre RSPF, ou par d'autres mécanismes.

Cadre de configuration à distance de la sécurité fondé sur des certificats: L'entité inscrite et la fonction MEF reçoivent chacune:

– une clé de signature privée, connue uniquement de l'entité concernée;

– un certificat contenant la clé de vérification publique correspondante; et

– (dans le cas d'un certificat de dispositif, d'un certificat de CSE-ID ou d'un certificat d'AE-ID) une chaîne de certificats reliant le certificat de l'entité au certificat d'une ancre de confiance.

Le certificat peut être préconfiguré ou configuré au sein d'un cadre RSPF au moyen des procédures de configuration de certificat décrites au § 8.3.6. Si une fonction MEF configure un client MEF, ce dernier doit s'authentifier auprès de la MEF au moyen du certificat configuré le plus récent reçu de la MEF.

L'entité inscrite et la fonction MEF doivent valider mutuellement leurs certificats avant de faire confiance aux clés de vérification publiques que contiennent ces certificats. Dans le cadre de la prise de contact de sécurité, la fonction MEF génère une signature numérique des paramètres de session en utilisant sa clé de signature privée et l'entité inscrite vérifie la signature numérique en utilisant la clé de vérification publique de la fonction MEF. Les rôles sont ensuite inversés: l'entité inscrite génère une signature numérique, que vérifie la fonction MEF.

Consulter le § 8.3.2.2 pour plus d'informations.

Cadre de configuration à distance de la sécurité fondé sur une GBA: dans ce cas de figure, le rôle de la fonction MEF est assuré par une fonction de serveur d'amorçage d'une architecture GBA. Ce cadre utilise des clés symétriques 3GPP ou 3GPP2 pour authentifier l'entité inscrite et la fonction MEF (qui est également une fonction de serveur d'amorçage de la GBA). Les documents 3GPP TS 33.220 [ETSI TS 133 220] et 3GPP2 S.S0109-A [TIA-1098-A] précisent les détails de ce cadre de configuration. Consulter le § 8.3.2.3 pour plus d'informations.

Les cadres de configuration à distance de la sécurité comprennent les phases suivantes:

Configuration du justificatif d'identité du client MEF: le client MEF et la fonction MEF reçoivent le justificatif d'identité d'amorçage qu'utilisera l'entité pour s'authentifier auprès de l'autre entité. Cette phase est également appelée configuration du justificatif d'identité d'amorçage.

Fréquence: si le justificatif d'identité est une clé symétrique, cette opération se produit une fois par association entre le client MEF et la fonction MEF. Si le justificatif d'identité est un certificat, cette opération se produit une fois par client MEF.

Configuration du service du client MEF: le client MEF reçoit l'URI de la fonction MEF (ce qui lui permettra d'acheminer les messages (D)TLS à la fonction MEF).

De plus, dans le cas d'un cadre de configuration à distance de la sécurité fondé sur des certificats:

Le client MEF est configuré avec les certificats de la CA constituant l'ancre de confiance, qu'il utilisera pour vérifier la fonction MEF.

La fonction MEF est configurée avec les informations de certificat du client MEF qu'elle utilisera pour vérifier le certificat du client MEF. Les informations de certificat nécessaires dépendent de la version du certificat du client MEF. Pour plus de détails, consulter le § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité".

Fréquence: cette opération se produit une fois par association entre le client MEF et la fonction MEF.

NOTE 2 – Dans le cas du cadre RSPF fondé sur une clé symétrique pré-partagée et du cadre RSPF fondé sur une architecture GBA, la configuration du justificatif d'identité du client MEF et l'attribution du client MEF se produisent généralement simultanément. Dans le cas du cadre RSPF fondé sur des certificats, l'attribution du client MEF peut se faire séparément de la configuration du justificatif d'identité du client MEF.

Coordination de la partie prenante administratrice avec la fonction MEF (détails non couverts par la présente spécification). Une partie prenante administratrice autorise la fonction MEF à fournir des services aux clients MEF, supervise l'autorisation de distribution des clés symétriques et contrôle la gestion des objets MO liés à la sécurité au niveau du client MEF. Cette coordination intervient généralement avant la prise de contact MEF.

Fréquence: à la demande de la partie prenante administratrice.

Instructions de la procédure de configuration: le client MEF peut déterminer de manière implicite qu'il doit exécuter des procédures de configuration spécifiques, ou bien recevoir des instructions explicites. Dans les deux cas, le client MEF exécute la prise de contact MEF et initie les procédures de configuration lors de l'échange d'inscription.

Fréquence: à chaque fois que le client MEF doit lancer un ensemble de procédures de configuration.

L'ensemble constitué par la configuration du service du client MEF, la coordination de la partie prenante administratrice avec la fonction MEF et les instructions des procédures de configuration est également dénommé "configuration des instructions d'amorçage".

Prise de contact MEF: identification, authentification et établissement du contexte de sécurité entre le client MEF et la fonction MEF.

Fréquence: une prise de contact MEF est effectuée à chaque fois que le client MEF est sollicité par des instructions de procédure de configuration.

Cette phase est également appelée prise de contact d'inscription d'amorçage.

Échange d'inscription: lorsque la prise de contact MEF a été réalisée avec succès dans le cadre RSPF fondé sur l'architecture GBA, le client MEF et la fonction MEF ont établi une clé symétrique à partir de laquelle il est possible de générer des clés qui pourront être utilisées avec d'autres AE, CSE ou fonctions MAF. Aucune autre interaction ne se produit entre le client MEF et la fonction MEF jusqu'à l'expiration de la clé symétrique, qui donne lieu à une nouvelle prise de contact.

Après une prise de contact MEF dans un cadre RSPF fondé sur une clé symétrique prépartagée ou sur des certificats, le client MEF et la fonction MEF ont établi un canal sécurisé qui servira à protéger l'échange d'inscription au cours duquel les justificatifs d'identité seront configurés. L'échange d'inscription est décrit plus en détail au § 8.3.4. Cet échange peut comprendre différents types de procédures: enregistrement de client MEF, configuration de clé symétrique, configuration de

certificat et configuration de dispositif. La séquence des procédures d'échange d'inscription peut être pilotée par la procédure de commande des clients MEF, laquelle est présentée dans les grandes lignes au § 8.3.4 et décrite plus en détail au § 8.3.9.

Fréquence: cet échange se produit chaque fois que les instructions de la procédure de configuration le déclenchent.

Utilisation des justificatifs d'identité configurés: les justificatifs d'identité peuvent ensuite être utilisés dans les cadres de sécurité suivants:

SAEF fondé sur des certificats, ESPrim et ESData: les certificats et ancres de confiance configurés sont utilisés directement dans les cadres de sécurité fondés sur les certificats avec les autres nœuds, AE ou CSE. Les ancres de confiance peuvent également être configurées séparément, par exemple conformément à la Recommandation [UIT-T Y.4500.22].

SAEF fondé sur une clé pré-partagée, ESPrim et ESData: le client MEF source et la fonction MEF ont établi une clé symétrique à usage limité, un identificateur de clé correspondant et une liste de clients MEF autorisés. Le client MEF source fournit l'identificateur de clé aux clients MEF cibles dans le cadre du protocole de sécurité. Le ou les clients MEF cibles établissent une connexion sécurisée avec la fonction MEF et exécutent la procédure de récupération de clé MEF (§ 8.3.5.2.8) afin de récupérer la clé symétrique sujette à autorisation au niveau de la fonction MEF.

SAEF fondé sur une fonction MAF, ESPrim et ESData: s'il est prévu d'utiliser des certificats pour l'authentification auprès de la fonction MAF, le certificat et les ancres de confiance configurés au moment de l'inscription du certificat sont utilisés pour l'authentification mutuelle du client MEF et de la fonction MAF. Si l'authentification mutuelle repose sur une clé symétrique, le client MEF et la fonction MEF ont établi une clé symétrique et un identificateur de clé correspondant, avec une contrainte d'usage pour une fonction MAF spécifique. Le client MEF (qui agit à présent en tant que client MAF) exécute la procédure d'enregistrement de client MAF, lors de laquelle le client MEF/client MAF fournit l'identificateur de clé à la fonction MAF. La fonction MAF établit une connexion sécurisée avec la fonction MEF et exécute la procédure de récupération de clé MEF (§ 8.3.5.2.8) afin de récupérer la clé symétrique sujette à autorisation au niveau de la fonction MAF. La fonction MAF transmet au client MEF/client MAF un KmID qui sera utilisé lors des prises de contact MAF suivantes.

NOTE 3 – Si la cible d'inscription héberge une ressource <*ServiceSubscribedAppRule*>, les justificatifs d'identité récupérés auprès de la fonction MEF ou de la fonction MAF doivent être stockés après que la cible d'inscription a établi une connexion sécurisée avec l'entité inscrite. Une valeur d'identificateur de justificatif d'identité au format mentionné au § 10.4 est générée à partir du justificatif d'identité ayant servi à l'établissement de la connexion sécurisée; cette valeur est ajoutée dans l'attribut *applicableCredIDs* de la ressource <*ServiceSubscribedAppRule*>.

NOTE 4 – Si l'identificateur de l'entité inscrite est récupéré auprès de la fonction MEF ou de la fonction MAF, il est enregistré dans l'attribut *allowedAEs* de la ressource <*ServiceSubscribedAppRule*>.

La Figure 8.3.1.2-1 illustre les différentes phases des cadres de configuration à distance de la sécurité.

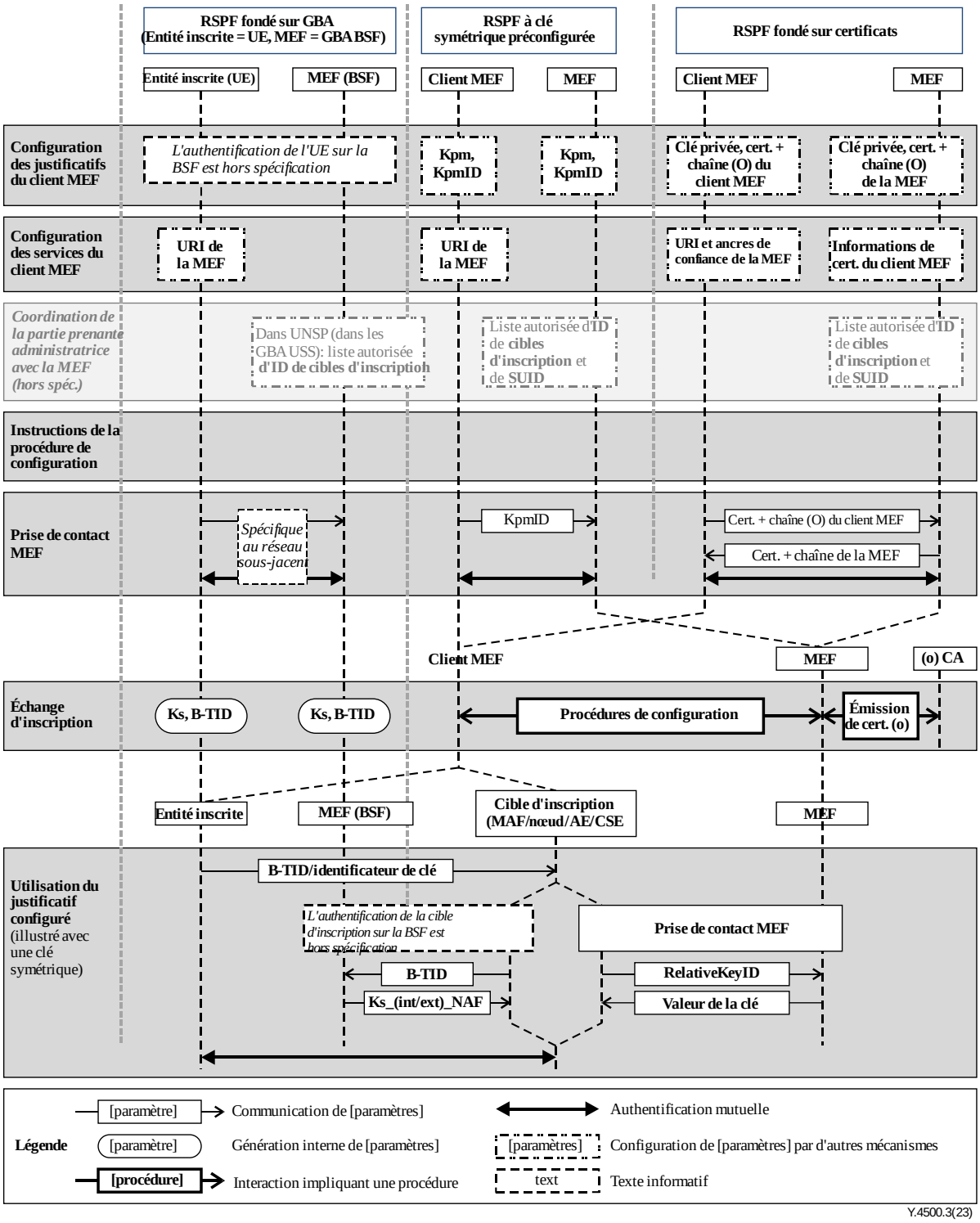


Figure 8.3.1.2-1 – Vue d'ensemble des cadres de configuration à distance de la sécurité pris en charge par le système oneM2M

8.3.2 Description détaillée des cadres de configuration à distance de la sécurité

8.3.2.1 Cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée

Ce paragraphe décrit le cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée. Le justificatif d'identité d'amorçage pour ce cadre est une clé symétrique à longue durée de validité que l'on a préconfigurée sur l'entité inscrite et la fonction d'inscription M2M; cette clé est appelée clé symétrique d'entité inscrite préconfigurée et notée Kpm.

La Figure 8.3.2.1-1 représente la séquence d'événements survenant lors de l'utilisation du cadre de configuration à distance de la sécurité fondé sur une clé symétrique d'entité inscrite préconfigurée.

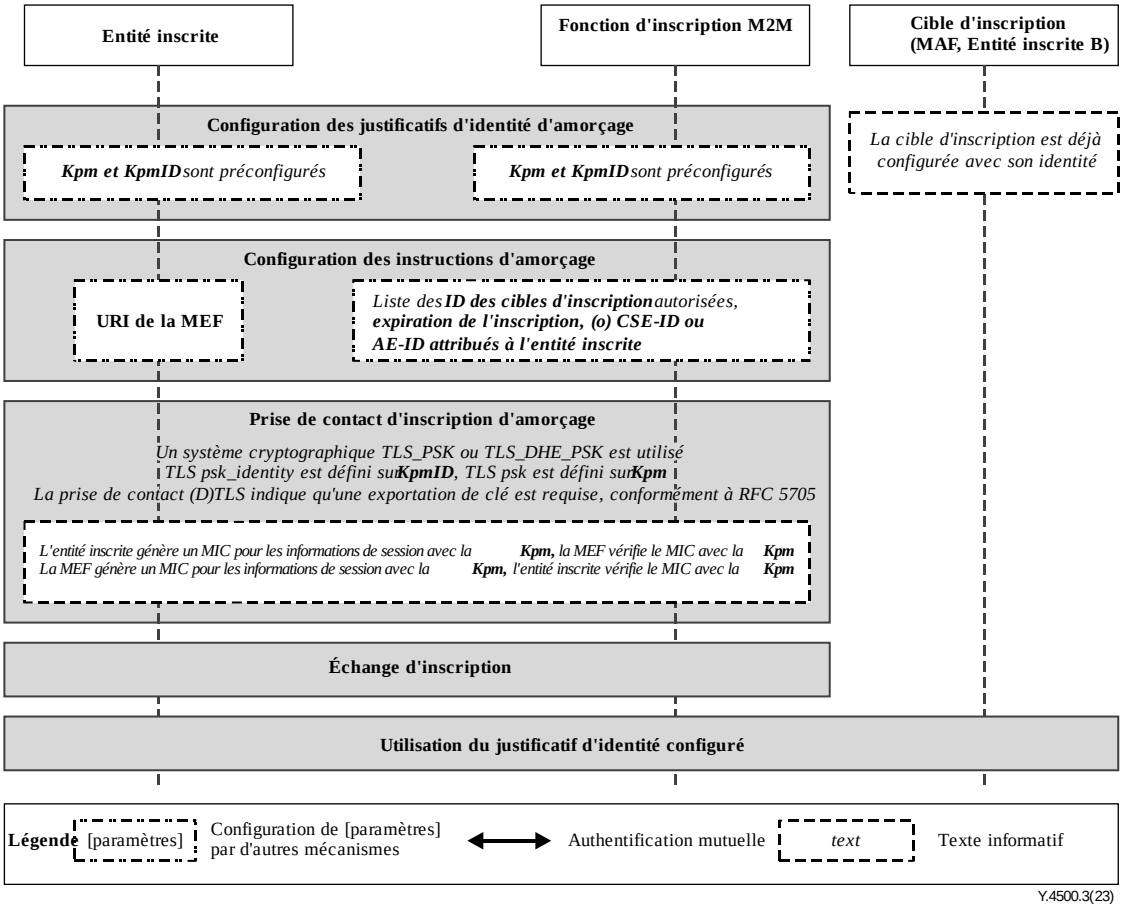


Figure 8.3.2.1-1 – Séquence d'événements liés à l'utilisation du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée

Configuration des justificatifs d'identité d'amorçage: la clé symétrique d'entité inscrite préconfigurée (Kpm) et l'identificateur de clé associé, noté KpmID, sont préconfigurés sur les deux entités. L'URI de la fonction d'inscription M2M (MEF URI) est également préconfiguré sur l'entité inscrite, afin de pouvoir acheminer l'échange (D)TLS.

NOTE 1 – Cette préconfiguration (par définition) emploie des mécanismes ne relevant pas des spécifications oneM2M.

Configuration des instructions d'amorçage: les informations nécessaires pour autoriser la configuration à distance sont configurées sur l'entité inscrite et la fonction d'inscription M2M:

L'entité inscrite reçoit par configuration (ou obtient d'une autre manière) les arguments suivants permettant d'initier la configuration à distance:

L'identité de la cible d'inscription: identifie la cible d'inscription pour laquelle l'entité inscrite doit être configurée.

L'entité inscrite associe ces arguments à la fonction d'inscription M2M. La fonction d'inscription M2M peut être identifiée, auprès de l'entité inscrite, au moyen de l'identificateur de clé symétrique d'entité inscrite préconfigurée (KpmID) ou de l'URI de la fonction d'inscription M2M.

Expiration de l'inscription: la durée de vie de la clé générée, c'est-à-dire la clé d'inscription (Ke) comme indiqué au § 10.7.

Les arguments suivants sont configurés sur la fonction d'inscription M2M pour autoriser celle-ci à configurer à distance une cible d'inscription pour l'entité inscrite:

L'identité de la cible d'inscription: identifie la cible d'inscription pour laquelle l'entité inscrite doit être configurée.

Le CSE-ID ou l'AE-ID ("identificateur d'entité inscrite") attribué à l'entité inscrite. La fonction d'inscription M2M doit fournir cette identité, ainsi que la clé Km ou Kpsa, à la cible d'inscription quand celle-ci le demande.

La fonction d'inscription M2M associe ces arguments à une entité inscrite. L'entité inscrite peut être identifiée auprès de la fonction d'inscription M2M au moyen de l'identificateur de clé symétrique d'entité inscrite préconfigurée (KpmID).

Expiration de l'inscription: la durée de vie de la clé générée, c'est-à-dire la clé d'inscription (Ke). La fonction d'inscription M2M peut fournir cette durée de vie en même temps que la Km ou la Kpsa à la cible d'inscription.

Prise de contact de sécurité d'amorçage: l'entité inscrite et la fonction d'inscription M2M doivent effectuer une prise de contact (D)TLS-PSK [IETF RFC 4279] pour établir une session sécurisée.

Le paramètre "psk_identity" [IETF RFC 4279] est défini sur la valeur de l'identificateur de clé symétrique d'entité inscrite préconfigurée (KpmID).

Le paramètre "psk" [IETF RFC 4279] est défini sur la valeur de la clé symétrique d'entité inscrite préconfigurée (Kpm).

Le système cryptographique (D)TLS est décrit au § 10.2.2 "Systèmes cryptographiques TLS and DTLS destinés aux cadres de sécurité TLS-PSK".

Génération des clés d'inscription: la clé d'inscription (Ke), l'identificateur RelativeKeID et la clé de réauthentification d'inscription (Ker) sont générés à partir des secrets de la session (D)TLS par l'entité inscrite et la fonction d'inscription M2M, au moyen d'une exportation de clés TLS ([IETF RFC 5705]). Ce processus est décrit au § 10.3.1 "Détails concernant l'exportation de clés à l'aide du protocole TLS".

L'identificateur de clé d'inscription (KeID) est généré à partir du RelativeKeID et du nom de domaine complet de la fonction MEF par l'entité inscrite et la fonction d'inscription M2M, comme décrit au § 10.3.4 "Génération de l'identificateur KeID".

L'entité inscrite et la fonction d'inscription M2M stockent la clé d'inscription (Ke) et l'identificateur de clé d'inscription (KeID), ainsi que la clé de réauthentification (Ker).

NOTE 2 – Le processus de génération de la clé d'inscription est identique pour le cadre de configuration à distance de la sécurité fondé sur une clé symétrique d'entité inscrite et pour le cadre fondé sur des certificats.

Échange d'inscription: l'entité inscrite compose une demande dont la charge utile contient les paramètres et les valeurs indiqués au Tableau 8.3.2.1-1. Il est possible de sérialiser ces paramètres en utilisant, par exemple, les formats XML ou JSON.

Tableau 8.3.2.1-1 – Demande initiale de l'entité inscrite à la fonction MEF

Nom du paramètre	Valeur du paramètre
Certificate Enrolment Indication	True/False
MAF Enrolment Indication	True/False
Remote Management Indication	True/False

Ces paramètres indiquent si l'entité inscrite est préparée, ou non, à exécuter les procédures concernées quand elle en reçoit l'instruction de la fonction MEF. L'entité inscrite doit envoyer la demande à l'URI d'échange d'inscription de la fonction MEF.

La fonction MEF doit traiter la demande en fonction des préférences pour l'entité inscrite (voir les conditions préalables), afin de déterminer comment doit être configurée cette dernière.

NOTE 3 – La présente Recommandation ne définit pas ce traitement.

Si l'entité inscrite ne requiert pas d'être configurée à distance en vue d'une authentification fondée sur des certificats avec des cibles d'inscription, la fonction MEF passe à l'étape 10. Pour configurer à distance l'entité inscrite en vue d'une authentification fondée sur des certificats avec des cibles d'inscription, la fonction MEF doit composer une réponse dont la charge utile contient les paramètres et les valeurs indiqués au Tableau 8.3.2.1-3. Il est possible de sérialiser ces paramètres en utilisant, par exemple, les formats XML ou JSON.

Tableau 8.3.2.1-2 – Réponse de la fonction MEF à l'entité inscrite qui déclenche l'inscription des certificats

Nom du paramètre	Valeur du paramètre
Instruction Type	<Indique l'inscription des certificats>
URI	<URI de base d'inscription des certificats>

La fonction MEF doit envoyer la réponse à l'entité inscrite.

Si la fonction MEF donne l'instruction à l'entité inscrite de procéder à l'inscription des certificats, cette dernière doit exécuter la procédure d'inscription des certificats décrite au § 8.3.3.1.

Une fois l'inscription des certificats terminée, l'entité inscrite envoie un message de réussite à la fonction MEF.

Si l'entité inscrite ne requiert pas d'être inscrite à distance auprès d'une fonction d'authentification M2M (MAF), la fonction MEF passe à l'étape 13. Pour inscrire à distance l'entité inscrite auprès d'une fonction MAF, la fonction MEF doit composer une réponse dont la charge utile contient les paramètres et les valeurs indiqués au Tableau 8.3.2.1-3. Il est possible de sérialiser ces paramètres en utilisant, par exemple, les formats XML ou JSON.

Tableau 8.3.2.1-3 – Réponse de la fonction MEF à l'entité inscrite qui déclenche l'inscription auprès d'une fonction MAF

Nom du paramètre	Valeur du paramètre
Instruction Type	<Indique l'inscription sur la fonction MAF>
Credential Type	<Indique s'il faut utiliser des certificats ou une clé symétrique pour s'authentifier auprès de la fonction MAF>
MAF Key Registration URI	<URI où sont enregistrées les clés MAF>
MAF Key Retrieval URI	<URI où sont récupérées les clés MAF>
(facultatif) MAF Client Registration URI	<URI où le KmID assigné par la fonction MAF peut être obtenu>
(facultatif) trust anchors	<Certificats de CA constituant des ancres de confiance pour le certificat MAF>
(facultatif) Lifetime	<Durée de vie au bout de laquelle la clé symétrique partagée avec la fonction MAF expire>

La fonction MEF doit envoyer la réponse à l'entité inscrite.

À réception du message, l'entité inscrite exécute la procédure d'enregistrement de client MAF, comme décrit au § 8.8.2.4. Cette procédure implique l'utilisation d'un "justificatif d'identité configuré à distance".

Une fois l'inscription du client MAF terminée, l'entité inscrite envoie un message de réussite à la fonction MEF. La fonction MEF peut revenir à l'étape 10, s'il faut configurer l'entité inscrite pour une autre fonction MAF.

Si l'entité inscrite ne requiert pas d'être configurée à distance pour un serveur de gestion distant à contacter pour des configurations ultérieures, la fonction MEF passe à l'étape 15. Pour configurer à distance l'entité inscrite sur un serveur de gestion distant, la fonction MEF doit composer une réponse dont la charge utile contient les paramètres et les valeurs indiqués au Tableau 8.3.2.1-4. Il est possible de sérialiser ces paramètres en utilisant, par exemple, les formats XML ou JSON.

Tableau 8.3.2.1-4 – Réponse de la fonction MEF pour configurer l'entité inscrite sur un serveur de gestion distant

Nom du paramètre	Valeur du paramètre
Instruction Type	<Indique le serveur de gestion distant>
URI	<URI de base du serveur de gestion distant >

La fonction MEF doit envoyer la réponse à l'entité inscrite.

L'entité inscrite envoie une confirmation de réception de l'instruction à la fonction MEF. L'entité inscrite prend contact avec le serveur de gestion distant lorsque la session TLS/DTLS avec la fonction MEF est clôturée.

La fonction MEF doit envoyer un message pour signaler la fin de l'échange d'inscription.

C'est la fonction MEF qui doit clôturer la session TLS/DTLS.

Utilisation d'un justificatif d'identité configuré à distance: lorsque la cible d'inscription est une fonction MAF, l'entité inscrite est invitée à contacter une fonction MAF spécifique avec laquelle elle doit effectuer l'inscription.

Si l'entité inscrite reçoit à distance un certificat et des ancres de confiance pendant l'échange d'inscription, elle peut les utiliser dans les protocoles de sécurité avec la clé d'inscription. Sinon, l'entité inscrite doit utiliser le KeID dans les protocoles de sécurité avec la cible d'inscription, comme décrit dans les étapes suivantes.

L'entité inscrite doit fournir le KeID comme identificateur de clé symétrique pendant le protocole de sécurité.

La cible d'inscription vérifie si elle dispose des justificatifs d'identité associés à ce KeID; si ce n'est pas le cas, elle se met en condition pour les récupérer auprès de la fonction MEF.

La cible d'inscription ayant été préconfigurée avec le nom de domaine complet/l'URL de la fonction MEF, et en vue d'établir une connexion sécurisée, elle utilise le certificat ou les justificatifs d'identité PSK qui ont également été préconfigurés entre la cible d'inscription et la fonction MEF. Dans le cas d'une entité inscrite B, il est possible d'utiliser pour l'authentification une clé de réauthentification d'inscription (Ker) établie avec la fonction MEF, ou un certificat configuré par la fonction MEF.

Si la cible d'inscription souhaite obtenir des informations de justificatif d'identité de la fonction MEF, elle doit adresser à cette dernière une demande de récupération dans laquelle l'URI cible est défini sur /fetchCredentials/<KeID>/<identificateur-d'utilisation-de sécurité>, où <identificateur-d'utilisation-de sécurité> est le SUID correspondant à l'usage particulier de la clé symétrique. Le champ Originator

de la demande (par exemple l'en-tête X-M2M-Origin si HTTPS est utilisé) contient l'identificateur (AE-ID/CSE-ID/MAF-ID) de la cible d'inscription/MAF.

À réception de la demande, la fonction MEF effectue les opérations suivantes:

La fonction MEF extrait le KeID de l'URI cible. Elle doit récupérer la clé d'inscription (Ke) pour le KeID correspondant, conformément au processus de génération de clé d'inscription décrit au § 8.3.1.2. Si la fonction MEF est dans l'incapacité de récupérer cette information, elle doit renvoyer une réponse d'erreur, conformément à l'étape f)vii.

L'identificateur de la cible d'inscription est extrait du champ Originator inclus dans la demande, tandis que l'identificateur d'utilisation de sécurité (SUID) est extrait de l'URI cible. La fonction MEF doit vérifier si la cible d'inscription concernée est autorisée à obtenir des justificatifs d'identité pour l'entité inscrite avec le SUID indiqué.

En cas d'échec de cette vérification, la fonction MEF renvoie une réponse d'erreur, conformément à l'étape f)vii. Si la vérification est concluante, la clé doit être générée à partir de la clé d'inscription (Ke) obtenue à l'étape f)i, comme indiqué au § 10.3.7.

La fonction MEF détermine l'identificateur de l'entité inscrite correspondant au KeID.

La fonction MEF détermine le paramètre de durée de vie de l'entité inscrite pour le KeID. Il s'agit d'une valeur préconfigurée qui indique la période de validité des justificatifs d'identité fournis à la cible d'inscription/fonction MAF.

La fonction MEF compose une réponse dont la charge utile contient les paramètres et les valeurs indiqués au Tableau 8.3.2.1-5. Il est possible de sérialiser ces paramètres en utilisant, par exemple, les formats XML ou JSON.

Tableau 8.3.2.1-5 – Réponse de réussite renvoyée par la fonction MEF à la cible d'inscription

Nom du paramètre	Valeur du paramètre
Status	True
Credential	<Clé>
EnroleeID	<Valeur de l'identificateur de l'entité inscrite>
Lifetime	<Durée de vie de la clé produite>

En cas d'erreur lors de l'une des étapes ci-dessus, la fonction MEF doit composer une réponse contenant les paramètres indiqués au Tableau 8.3.2.1-6.

Tableau 8.3.2.1-6 – Réponse d'échec renvoyée par la fonction MEF à la cible d'inscription

Nom du paramètre	Valeur du paramètre
Status	False
ErrorString	<Raison de l'échec>

Une fois que la cible d'inscription a reçu les justificatifs d'identité, elle peut les utiliser dans le protocole de sécurité avec l'entité inscrite.

8.3.2.2 Cadre de configuration à distance de la sécurité fondé sur des certificats

Ce paragraphe décrit le cadre de configuration à distance de la sécurité fondé sur des certificats. Les justificatifs d'identité d'amorçage pour ce cadre sont des certificats reposant sur des paires de clé asymétriques.

NOTE 1 – Les clés privées asymétriques à longue durée de validité peuvent poser un risque de sécurité si elles ne sont pas protégées de façon adéquate. Il est recommandé de les stocker dans des environnements sécurisés.

La Figure 8.3.2.2-1 représente la séquence d'événements survenant lors de l'utilisation du cadre d'établissement d'association de sécurité fondé sur des certificats.

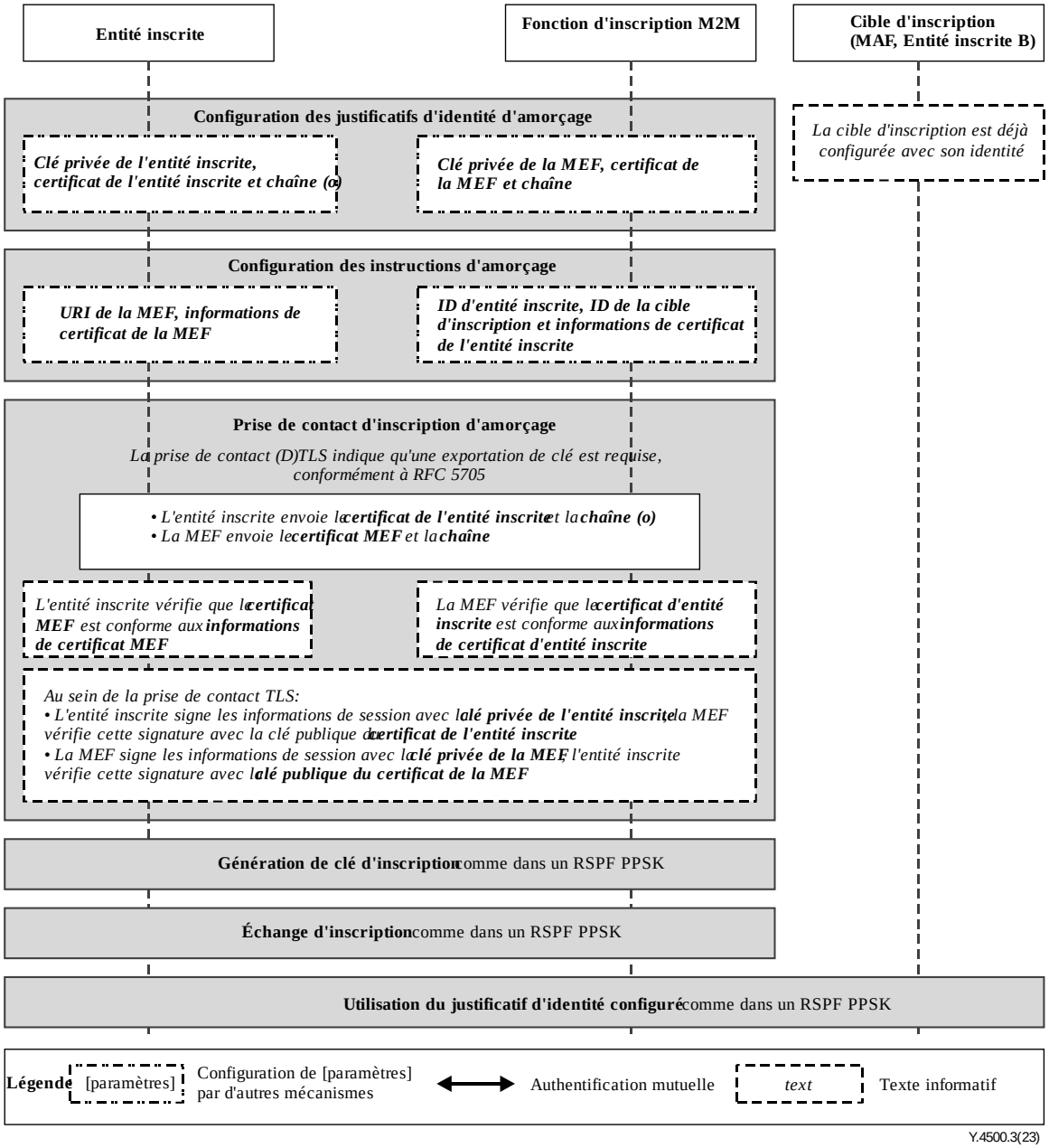


Figure 8.3.2.2-1 – Séquence d'événements liés à l'utilisation du cadre de configuration à distance de la sécurité fondé sur des certificats

Configuration des justificatifs d'identité d'amorçage: pour ce cadre de configuration à distance de la sécurité, l'entité inscrite et la fonction d'inscription M2M s'authentifient mutuellement au moyen d'un certificat de clé publique. Les justificatifs d'identité d'amorçage pour l'entité inscrite et la fonction d'inscription M2M doivent être préconfigurés comme décrit au § 8.1.2.3 "Configuration des justificatifs d'identité pour le cadre de sécurité fondé sur des certificats".

NOTE 2 – Les identités de la fonction d'inscription M2M et de la cible d'inscription sont supposées avoir été configurées avant cette phase.

Configuration des instructions d'amorçage: outre les informations identifiées au § 8.3.1.2, les informations nécessaires pour autoriser la configuration à distance sont configurées sur l'entité inscrite et la fonction d'inscription M2M:

L'entité inscrite reçoit par configuration (ou obtient d'une autre manière) les arguments suivants permettant d'initier la configuration à distance:

Informations requises pour l'authentification par certificat de la fonction d'inscription M2M au moyen d'un certificat MEF, décrites au § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité".

La fonction d'inscription M2M est configurée les arguments suivants, qui décrivent l'entité inscrite autorisée à effectuer la prise de contact de sécurité avec ladite fonction:

Informations requises pour l'authentification par certificat de l'entité inscrite, décrites au § 8.1.2.4 "Informations requises pour l'authentification par certificat d'une autre entité".

Prise de contact de sécurité d'amorçage: l'entité inscrite et la fonction d'inscription M2M exécutent une prise de contact (D)TLS comme décrit dans les spécifications TLS v1.2 [IETF RFC 5246] et DTLS 1.2 [IETF RFC 6347], afin d'établir une session sécurisée.

Chaque entité (entité inscrite et fonction MEF) doit vérifier le certificat de l'autre, comme décrit au § 8.1.2.5 "Vérification des certificats".

L'entité inscrite et la fonction MEF s'authentifient l'une l'autre en utilisant les certificats validés, comme décrit dans les spécifications TLS 1.2 [IETF RFC 5246] et DTLS 1.2 [IETF RFC 6347].

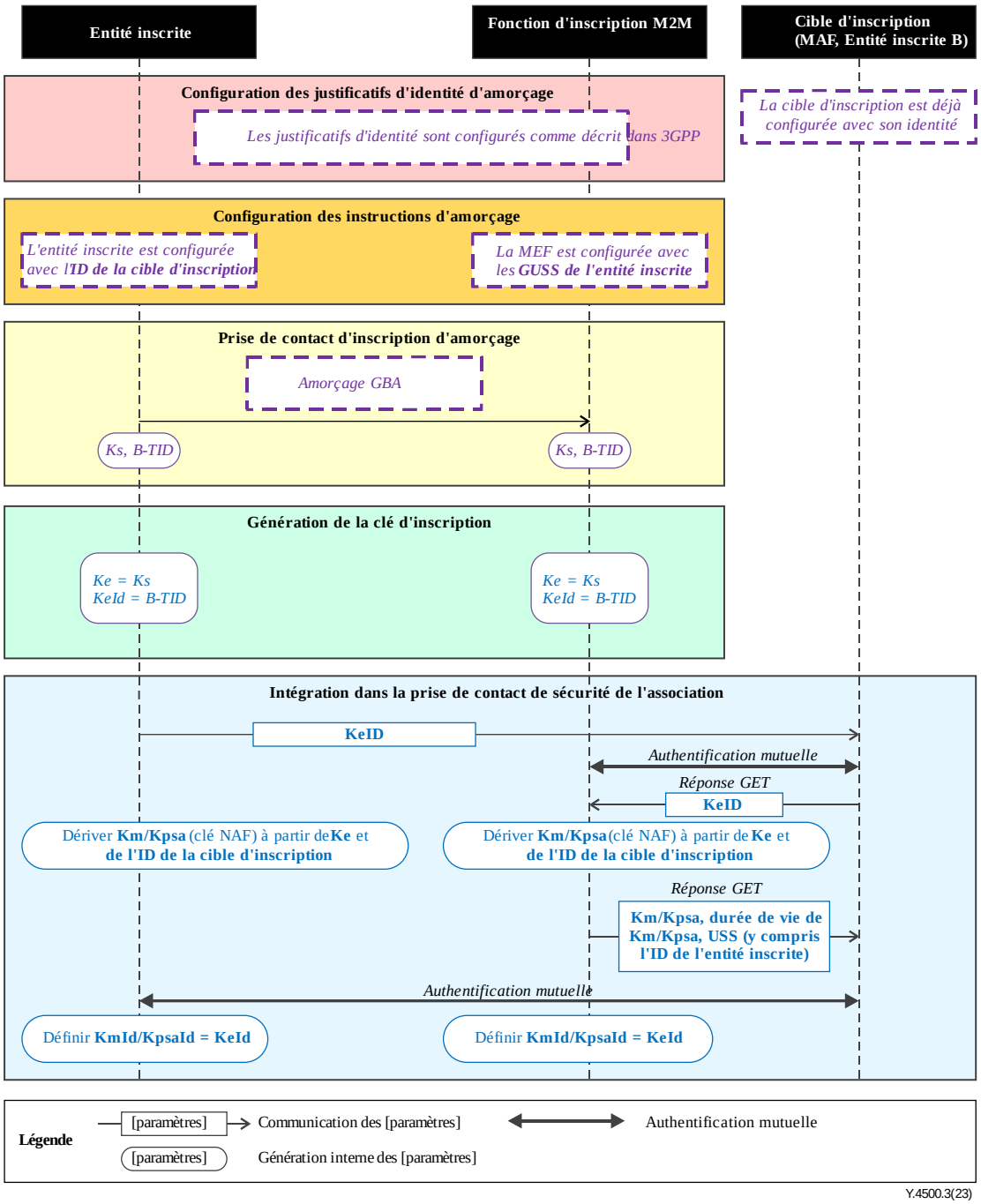
Le système cryptographique (D)TLS est abordé au § 10.2.3 "Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité TLS-PSK".

Génération des clés d'inscription, échange d'inscription et **utilisation des justificatifs d'identité configurés**: les étapes sont les mêmes qu'aux alinéas "Génération des clés d'inscription", "Échange d'inscription" et "Utilisation des justificatifs d'identité configurés" du § 8.3.2.1 "Cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée".

8.3.2.3 Cadre de configuration à distance de la sécurité fondé sur une architecture GBA

Pour pouvoir partager un justificatif d'identité maître (Km) à longue durée de validité ou une clé de connexion sécurisée configurée (Kpsa) entre un nœud de service d'application/nœud intermédiaire et une cible d'inscription, le nœud de service d'application/nœud intermédiaire M2M doit effectuer avec succès un amorçage GBA et générer une clé NAF (Ks_(ext/int)_NAF). Cette clé NAF constitue le justificatif d'identité maître (Km) ou la clé de connexion sécurisée configurée (Kpsa).

Voir la Figure 8.3.2.3-1.



NOTE – Les couleurs de police suivantes permettent d'identifier les différents sujets auquel se rapporte le texte: Le texte en noir décrit les détails indépendants du type de cadre de configuration à distance de la sécurité.
Le texte en italique bleu souligne les détails spécifiques à ce cadre particulier de configuration à distance de la sécurité.
Le texte en italique violet souligne les actions techniques qui peuvent comprendre des étapes non décrites par la spécification oneM2M.

Figure 8.3.2.3-1 – Séquence d'événements liés à l'utilisation du cadre de configuration à distance de la sécurité fondé sur une architecture GBA

Configuration des justificatifs d'identité d'amorçage: la configuration des justificatifs d'identité pour l'entité inscrite et la fonction d'inscription M2M (MEF) est décrite dans la spécification 3GPP TS 33.220 [ETSI TS 133 220]. La fonction MEF fait office de serveur d'amorçage (fonction BSF). Les justificatifs d'identité utilisés pour l'authentification mutuelle entre l'entité inscrite et la fonction MEF sont spécifiques au fournisseur de services de réseau sous-jacent (UNSP).

Configuration des instructions d'amorçage: les informations nécessaires pour autoriser la configuration à distance doivent être configurées sur l'entité inscrite, la fonction MEF et la cible d'inscription:

L'entité inscrite doit être configurée avec l'identité de la cible d'inscription, c'est-à-dire que la cible d'inscription doit être identifiée sur l'entité inscrite.

La fonction MEF doit être configurée avec l'identificateur de l'entité inscrite et l'identité de la cible d'inscription:

L'identité de la cible d'inscription identifie la cible d'inscription pour laquelle l'entité inscrite (authentifiée au moyen de l'architecture GBA) doit être configurée.

Le CSE-ID ou l'AE-ID attribué à l'entité inscrite, qui constitue l'identificateur de justificatif d'identité de l'entité inscrite. La fonction d'inscription M2M doit fournir cette identité, qui est celle de l'entité inscrite possédant la clé Km ou Kpsa, à la cible d'inscription quand celle-ci la demande.

Les réglages de sécurité utilisateur GBA (GUSS) de l'entité inscrite permettent d'indiquer si l'entité inscrite est autorisée à établir une clé spécifique à une fonction d'application réseau (NAF) avec la cible d'inscription, et/ou si la BSF peut distribuer une clé spécifique à la NAF à la cible d'inscription.

Prise de contact d'inscription d'amorçage: la prise de contact d'inscription d'amorçage autorise l'établissement d'une clé amorcée par la GBA (Ks) partagée entre l'entité inscrite et la fonction MEF, avec un identificateur de transaction d'amorçage (B-TID) associé et une durée de vie de clé, en exécutant la phase d'amorçage GBA décrite dans la spécification 3GPP TS 33.220 [ETSI TS 133 220].

Lorsqu'une clé amorcée Ks est déjà partagée entre l'entité inscrite et la fonction MEF, et que cette clé est toujours valide, la phase de prise de contact d'inscription d'amorçage n'est pas requise. La phase de génération des clés d'inscription peut se servir des données de clé Ks d'amorçage GBA existantes.

Phase de génération des clés d'inscription: la clé d'inscription (Ke) doit être la clé d'amorçage GBA (Ks) établie pendant la prise de contact d'inscription d'amorçage.

L'identificateur de clé d'inscription (Ke-ID) doit être l'identificateur de transaction d'amorçage (B-TID) généré pendant la prise de contact d'inscription d'amorçage.

Intégration dans la prise de contact de sécurité de l'association: l'entité inscrite et la cible d'inscription doivent établir le justificatif d'identité maître (Km) ou la clé de connexion sécurisée configurée (Kpsa) au moyen des procédures décrites dans la spécification 3GPP TS 33.220 [ETSI TS 133 220], en utilisant clé d'inscription (Ke) en tant que clé d'amorçage GBA Ks, et l'identificateur de clé d'inscription (Ke-ID) en tant que B-TID. La cible d'inscription fait office de fonction d'application réseau (fonction NAF).

La spécification [ETSI TS 133 220] décrit les réglages de sécurité utilisateur (USS) de l'entité inscrite provenant de la fonction MEF/BSF:

Le nom de domaine complet (FQDN) de la fonction d'application réseau (NAF), utilisé comme donnée d'entrée pour générer les données de clé Ks_(int/ext)_NAF, doit être établi comme suit:

– dans le cas où la cible d'inscription est une fonction d'authentification M2M (fonction MAF), le FQDN de la fonction NAF est défini sur le FQDN de la fonction MAF;
– dans le cas où la cible d'inscription est une CSE, le FQDN de la fonction NAF est défini sur la représentation du CSE-ID sous forme d'un nom de domaine public, comme défini dans la Recommandation [UIT-T Y.4500.1].

Dans le cas d'une architecture GBA_ME, la clé spécifique à la fonction d'application réseau est notée Ks_NAF.

Dans le cas d'une architecture GBA_U, les clés spécifiques à la fonction d'application réseau sont notées Ks_int_NAF et Ks_ext_NAF.

Le justificatif d'identité maître (Km) ou la clé de connexion sécurisée configurée (Kpsa) doit être la clé spécifique à la fonction d'application réseau:

Dans le cas d'une GBA_ME, Km/Kpsa = Ks_NAF.

Dans le cas d'une GBA_U, Km/Kpsa = Ks_int_NAF si l'application du client HTTP réside sur la carte UICC. Dans le cas contraire, Km/Kpsa = Ks_ext_NAF.

L'entité inscrite et la cible d'inscription doivent définir l'identificateur du justificatif d'identité maître (KmID) ou l'identificateur de clé de connexion sécurisée configurée (KpsaID) sur la valeur de KeID.

L'entité inscrite et la cible d'inscription doivent effectuer une prise de contact (D)TLS-PSK [IETF RFC 4279] avec le justificatif d'identité maître (Km) ou la clé de connexion sécurisée configurée (Kpsa) en tant que clé prépartagée, conformément au § 10.2.2 "Systèmes cryptographiques TLS et DTLS pour les cadres de sécurité TLS-PSK". Si une carte UICC est utilisée comme environnement sécurisé prenant en charge la configuration à distance de la sécurité, une architecture GBA-U pour laquelle Km/Kpsa = Ks_int_NAF doit être employée pour l'authentification et l'échange de clés.

8.3.3 Le présent paragraphe est intentionnellement laissé en blanc

8.3.4 Échange d'inscription

8.3.4.1 Procédures d'échange d'inscription

Les procédures suivantes peuvent se produire lors d'un échange d'inscription:

– procédures d'enregistrement de client MEF;

– procédures de configuration de clé symétrique;

– procédure de configuration de certificat;

– procédures de configuration de dispositif, conformément à la Recommandation [UIT-T Y.4500.22], la fonction MEF interagissant avec un serveur DM et le client MEF interagissant avec le client DM sur l'entité gérée;

– procédures de commande de client MEF (c'est-à-dire des procédures CRUD ciblant une ressource <*mefClientCmd*>), qui permettent à la fonction MEF de piloter la séquence des procédures d'échange d'inscription.

Le paragraphe ci-dessous décrit les mécanismes de déclenchement propres à chaque ensemble de procédures. D'autres mécanismes, non décrits par la spécification oneM2M, peuvent également être utilisés pour déclencher n'importe quelle procédure d'échange d'inscription, à condition que ces mécanismes assurent un niveau de sécurité suffisant. Une préconfiguration et une configuration manuelle sont des exemples de tels mécanismes.

8.3.4.2 Enregistrement des clients MEF

Les procédures d'enregistrement des clients MEF sont décrites aux § 8.3.5.2.3, 8.3.5.2.4, 8.3.5.2.5 et 8.3.5.2.6.

Ces procédures ne peuvent être exécutées que dans le cadre d'un échange d'inscription.

Elles peuvent être déclenchées par les mécanismes suivants, détaillés dans la spécification oneM2M:

Procédures déclenchées par une configuration de dispositif: une configuration de dispositif, décrite dans la Recommandation [ITU-T Y.4500.22], peut déclencher des procédures d'enregistrement de clients MEF:

L'ajout d'un objet MO [*MEFClientRegCfg*] amène le client MEF à exécuter la procédure d'enregistrement de client MEF décrite au § 8.3.5.2.3.

La suppression d'un objet MO [*MEFClientRegCfg*] amène le client MEF à cesser d'utiliser l'enregistrement de client MEF associé, à supprimer les éventuels justificatifs d'identité associés à cet enregistrement de client MEF et à mettre fin à l'enregistrement de client MEF associé au niveau de la fonction MEF. La fonction MEF se charge de l'étape finale, qui consiste à exécuter la procédure de désenregistrement de client MEF décrite au § 8.3.5.2.6.

8.3.4.3 Configuration des clés symétriques

Les procédures de configuration des clés symétriques sont décrites aux § 8.3.5.2.7, 8.3.5.2.8, 8.3.5.2.9 et 8.3.5.2.10.

Ces procédures ne peuvent être exécutées que dans le cadre d'un échange d'inscription.

Elles peuvent être déclenchées par les mécanismes suivants détaillés par la spécification oneM2M:

Procédures déclenchées par une commande de client MEF "MO_Node": une configuration de dispositif ([UIT-T Y.4500.22]) peut être utilisée pour configurer un client MEF avec un objet MO [*authenticationProfile*] qui dispose d'un nœud MO enfant [*MEFClientRegCfg*] de façon à informer le client MEF qu'une configuration de clé symétrique sera employée pour les justificatifs d'identité requis dans cet objet MO [*authenticationProfile*]. La réception par un client MEF d'une commande de client MEF "MO_NODE" correspondant au chemin d'un tel objet MO [*authenticationProfile*] peut déclencher la procédure de configuration de clé symétrique en fonction des informations dans la commande de client MEF et des valeurs actuelles des paramètres dans ces nœuds MO.

NOTE – Le recours à une configuration de dispositif pour actualiser ou supprimer l'objet MO [*authenticationProfile*] et/ou son nœud MO enfant [*MEFClientRegCfg*] ne déclenche pas implicitement une procédure de configuration de clé symétrique. L'actualisation ou la suppression ne prennent effet que lorsqu'une procédure de configuration de clé symétrique a été déclenchée par un autre mécanisme.

Procédures déclenchées par l'expiration de l'enregistrement d'une clé MEF: si le client MEF a préalablement exécuté (avec succès) une procédure d'enregistrement de clé MEF sous le contrôle d'un objet MO [*authenticationProfile*] sur le client MEF, que la date et heure actuelles dépassent la valeur *expirationTime* de la ressource [*authenticationProfile*], et que ces date et heure actuelles sont proches ou excèdent la valeur *expirationTime* de l'enregistrement de clé MEF le plus récent, le client MEF peut déclencher une procédure d'enregistrement de clé MEF. Le critère "proche de la valeur de *expirationTime*" est laissé à l'appréciation lors de la mise en œuvre du client MEF.

Procédures déclenchées par la réception, dans le cadre d'un protocole de sécurité oneM2M, d'un identificateur de clé symétrique dont le FQDN correspond au FQDN de la fonction MEF. La réception, par un client MEF cible pendant un protocole de sécurité oneM2M, d'un identificateur de clé symétrique dont le FQDN correspond à celui de la fonction MEF peut amener le client MEF à exécuter la procédure de récupération de clé MEF décrite au § 8.3.5.2.8. Voir les Étapes 6 et 7 du § 8.3.5.1.

8.3.4.4 Configuration des certificats

Les procédures de configuration des certificats sont décrites au § 8.3.6.

Ces procédures ne peuvent être exécutées que dans le cadre d'un échange d'inscription.

Elles peuvent être déclenchées par les mécanismes suivants détaillés par la spécification oneM2M:

Déclenchement des procédures par la réception d'une commande de client MEF: la réception, par le client MEF, d'une commande de client MEF identifiant une procédure de configuration de certificat amène le client MEF à exécuter cette procédure de configuration de certificat en utilisant les informations incluses dans la commande.

8.3.4.5 Configuration des dispositifs

La configuration des dispositifs est traitée dans la Recommandation [UIT-T Y.4500.22].

Il est possible d'exécuter une configuration de dispositif dans le cadre d'un échange d'inscription avec une fonction MEF, ou d'une session DM avec d'autres serveurs DM (session distincte de tout échange d'inscription). Le § 8.3.8 décrit le recours à une configuration de dispositif dans le cadre d'un échange d'inscription avec une fonction MEF.

Une configuration de dispositif peut être déclenchée par les mécanismes suivants détaillés par la spécification oneM2M:

Procédures déclenchées par la réception d'une commande de client MEF: la réception, par le client MEF, d'une commande de client MEF identifiant la procédure de configuration de dispositif amène le client MEF à exécuter une session de configuration de dispositif en utilisant les informations incluses dans la commande.

8.3.4.6 Commande des clients MEF

Les procédures de commande des clients MEF sont décrites au § 8.3.9.

Ces procédures ne peuvent être exécutées que dans le cadre d'un échange d'inscription.

Elles peuvent être déclenchées par les mécanismes suivants détaillés par la spécification oneM2M:

Procédures déclenchées suite à une procédure d'enregistrement de client MEF: une récupération de commande de client MEF doit être lancée à la suite d'une procédure d'enregistrement de client MEF (autre qu'un désenregistrement de client MEF).

Procédures déclenchées par *retryDuration*: lorsque la fonction MEF émet une commande de client MEF NO_MORE_COMMANDS, la référence *cmdArgs* comprend un attribut *retryDuration* indiquant la durée au bout de laquelle le client MEF tente une récupération de commande de client MEF. L'attribut *retryDuration* est annulé dès lors que le client MEF interagit avec succès avec la fonction MEF avant expiration du délai. Consulter le § 8.3.9.6 pour plus de détails.

Procédures déclenchées suite à une tentative d'exécuter une commande de client MEF reçue: lorsque le client MEF a tenté d'exécuter une commande de client MEF reçue, il doit exécuter la procédure d'actualisation de commande de client MEF pour informer la fonction MEF du statut de cette exécution. La fonction MEF peut envoyer une commande de client MEF dans sa réponse.

Un exemple de procédure de commande de client MEF est illustré à la Figure 8.3.4.6-1.

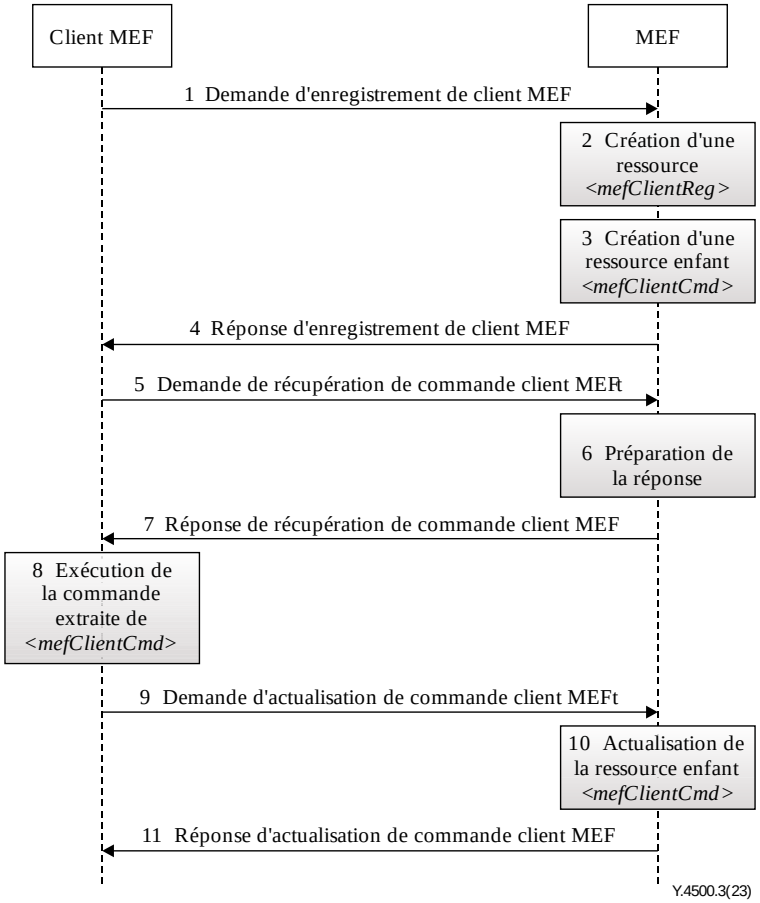


Figure 8.3.4.6-1 – Exemple de procédure de commande de client MEF

Le client MEF envoie une demande d'enregistrement de client MEF.

La fonction MEF créée une ressource <*mefClientReg*>.

Si la fonction MEF souhaite émettre une commande de client MEF, elle crée une ressource <*mefClientCmd*> enfant de la ressource <*mefClientReg*>.

La fonction MEF envoie la réponse d'enregistrement de client MEF, laquelle comprend une représentation de la ressource <*mefClientReg*>, notamment la référence *childResource*, dont la valeur correspond à l'identificateur d'une ressource <*mefClientCmd*>.

La présence de la référence *childResource* amène le client MEF à récupérer la ressource <*mefClientCmd*>. Le client MEF envoie une demande de récupération de commande de client MEF à la fonction MEF.

La fonction MEF compose une réponse.

La fonction MEF renvoie une réponse de récupération de commande de client MEF qui contient la ressource <*mefClientCmd*>.

Le client MEF analyse la réponse reçue et exécute la commande qu'elle contient.

Une fois la commande exécutée, le client MEF informe la fonction MEF du résultat par le biais d'une demande d'actualisation de commande de client MEF.

La fonction MEF actualise <*mefClientCmd*>. Si la fonction MEF dispose d'une nouvelle commande pour le client MEF, elle insère un déclencheur dans la représentation de la ressource <*mefClientCmd*>.

La fonction MEF envoie la réponse d'actualisation de commande de client MEF. Si la réponse reçue contient une autre commande de client MEF, les Étapes 8 à 11 sont répétées.

8.3.5 Détails concernant la configuration des clés symétriques

8.3.5.1 Introduction

Le paragraphe 8.3.5 décrit les détails et procédures communs à l'utilisation d'un cadre de configuration à distance de la sécurité pour configurer des clés symétriques.

Ces cadres font appel à une fonction MEF pour assurer l'authentification et la distribution de la clé symétrique destinée à être utilisée par le point d'extrémité source à l'origine de l'établissement de la clé symétrique, et par un ou plusieurs points d'extrémité cibles. Le Tableau 8.3.5.1-1 "Correspondance des rôles entre les cadres de sécurité fondés sur la fonction MEF" permet aux clients MEF de récupérer la clé symétrique produite auprès de la fonction MEF. La fonction MEF offre ses services pour le compte de parties prenantes administratrices, par exemples des fournisseurs de service M2M ou des générateurs de confiance M2M (MTE) tiers. Une partie prenante administratrice autorise la fonction MEF à fournir des services à des clients MEF, et supervise l'autorisation de la distribution des clés symétriques. Le Tableau 8.3.5.1-1 fait correspondre, au client MEF source et au client MEF cible, les rôles dans chaque cadre spécifique fondé sur la fonction MEF. Il indique également le nombre de clients MEF cibles autorisés.

Tableau 8.3.5.1-1 – Correspondance des rôles entre les cadres de sécurité fondés sur la fonction MEF

Cadre de sécurité fondé sur une fonction MEF	Client MEF source	Client MEF cible	Nombre de clients MEF cibles	Clé symétrique de sortie
Cadres de sécurité fondés sur une fonction MAF	Client MAF	Fonction MAF	1	Clé maître M2M (Km)
Cadre d'établissement d'association de sécurité (SAEF)	Entité A	Entité B	1	Clé de connexion sécurisée M2M (Kc)
Sécurité de bout en bout des primitives (ESPrim)	Expéditeur	Récepteur	1	Clé pairwiseESPrimKey
Sécurité de bout en bout des données (ESData)	Point d'extrémité ESData source	Point d'extrémité ESData cible	1 à n	Clé ESData

Le présent paragraphe 8.3.5 décrit les *procédures MEF* entre clients MEF, et les messages associés. Le fonctionnement et la gestion de la fonction MEF, au-delà des détails donnés pour les procédures MEF, ne sont pas abordés dans la présente Recommandation.

La séquence générale d'utilisation des procédures MEF est illustrée à la Figure 8.3.5.1-1 et décrite ci-après:

– Chaque client MEF doit établir séparément des justificatifs d'identité en vue de l'authentification mutuelle avec la fonction MEF, comme décrit à l'étape **Configuration des justificatifs d'identité des clients MEF** (§ 8.3.7.1).

– Chaque client MEF doit être configuré séparément pour s'enregistrer auprès de la fonction MEF avec une partie prenante administratrice donnée. L'étape **Configuration d'enregistrement des clients MEF** (§ 8.3.7.2) fournit les paramètres nécessaires.

– Chaque client MEF doit exécuter une **procédure d'enregistrement de client MEF** avec la fonction MEF, pour confirmer qu'il est prêt à utiliser les services de la fonction MEF, avec l'autorisation de la partie prenante administratrice. Le client MEF doit s'enregistrer séparément pour chaque partie prenante administratrice, même si l'enregistrement se fait sur une seule et même fonction MEF. Si le client MEF est configuré à distance en vue de l'authentification mutuelle avec la fonction MEF, cette dernière doit lui fournir le KmID qu'il devra utiliser par la suite pour s'authentifier auprès de la fonction MEF.

Ultérieurement, et indépendamment de cette séquence d'événements, la **procédure d'actualisation d'enregistrement de client MEF** peut être exécutée pour confirmer que le client MEF souhaite utiliser les services de la fonction MEF, et/ou pour établir de nouvelles données de clé Km et KmID; de même, la **procédure de désenregistrement de client MEF** peut être exécutée pour signaler que ce client MEF cesse d'utiliser les services de la fonction MEF.

Le client MEF source doit être configuré pour établir une communication sécurisée au moyen d'un cadre de sécurité (SAEF, ESPrim ou ESData) avec des clés symétriques établies par le biais de la fonction MEF. Les détails de cette configuration sont spécifiques au cadre de sécurité employé, mais elle doit comprendre une **Configuration d'enregistrement de clé MEF** (§ 8.4.4.3).

Le client MEF source doit exécuter une **procédure d'enregistrement de clé MEF** pour établir une clé symétrique et l'identificateur associé. Le client MEF doit également fournir l'identificateur d'utilisation de sécurité (SUID) qui limite la portée du justificatif d'identité en précisant le cadre de sécurité (SAEF, ESPrim ou ESData) dans lequel il peut être employé. Cette procédure doit comprendre la **procédure de prise de contact MEF** en vue de l'authentification mutuelle du client MEF source et de la fonction MEF.

Ultérieurement, et indépendamment de cette séquence d'événements, la **procédure d'actualisation d'enregistrement de clé MEF** peut être effectuée pour mettre à jour l'expiration de la clé enregistrée ou la liste des clients MEF cibles; de même, il est possible d'exécuter la **procédure de désenregistrement de clé MEF** pour supprimer l'enregistrement de la clé sur la fonction MEF.

Le client MEF source doit fournir, aux clients MEF cibles, l'identificateur de clé symétrique établi pendant la procédure d'enregistrement de clé MEF. Les détails de cette étape dépendent du cadre de sécurité, identifié par le SUID.

Le client MEF cible doit exécuter la **procédure de récupération de clé MEF** afin de récupérer la clé symétrique et les informations associées. Cette procédure doit comprendre la **procédure de prise de contact MEF** en vue de l'authentification mutuelle du client MEF cible et de la fonction MEF.

La clé symétrique doit être utilisée dans le protocole de sécurité entre le client MEF source et le client MEF cible. Si le protocole de sécurité requiert une seule clé symétrique, on utilisera la première moitié de la clé symétrique distribuée. Si le protocole de sécurité requiert deux clés symétriques (par exemple une clé de chiffrement et une clé d'intégrité distincte), on utilisera les deux moitiés de la clé symétrique distribuée, chacune constituant l'une des deux clés symétriques requises par le protocole de sécurité. Les détails de cette étape dépendent du cadre de sécurité.

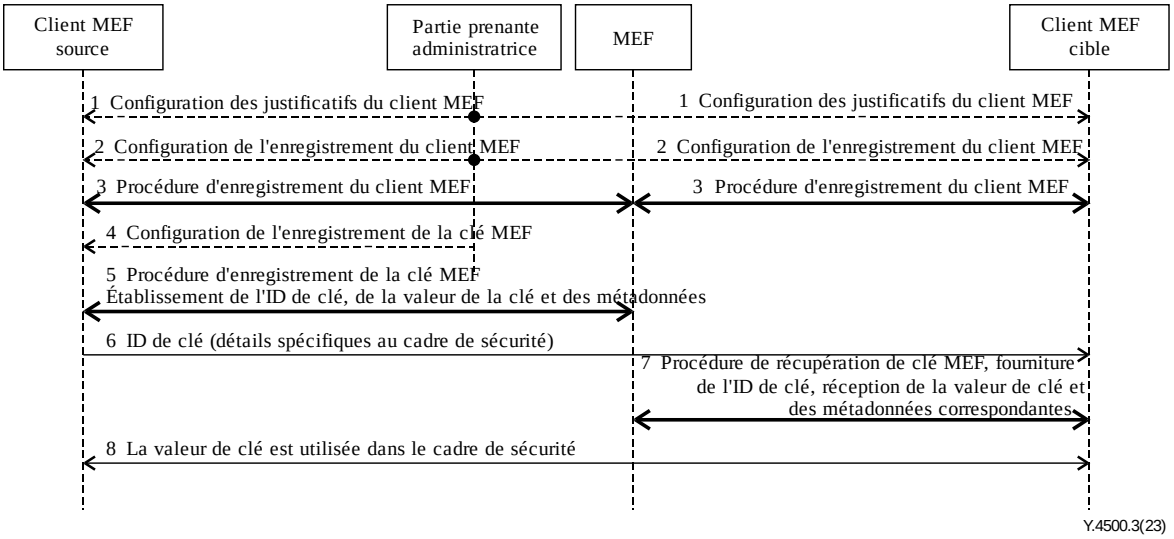


Figure 8.3.5.1-1 – Séquence d'événements liés à l'utilisation du cadre de sécurité MEF dans une fonctionnalité de sécurité

Le paragraphe 8.3.5.2 ci-après décrit le traitement et les flux d'information des procédures MEF. Le paragraphe 8.3.7 détaille les informations dans le justificatif d'identité du client MEF, la configuration d'enregistrement des clients MEF et la configuration d'enregistrement de clé MEF.

8.3.5.2 Traitement et flux des informations dans un cadre de sécurité MEF

8.3.5.2.1 Introduction

Le paragraphe 8.3.5.2 décrit le traitement et le flux des informations dans les procédures MEF.

8.3.5.2.2 Procédure de prise de contact MEF

Finalité: une procédure de prise de contact MEF établit une session TLS ou DTLS mutuellement authentifiée destinée à protéger les communications entre un client MEF et la fonction MEF. Dans le cas de la procédure d'enregistrement de clé MEF, la session TLS ou DTLS peut être utilisée par le client MEF source et la fonction MEF pour établir la valeur de la clé.

Conditions préalables: l'une des conditions ci-après est obligatoire:

Le client MEF et la fonction MEF ont été configurés avec des certificats, comme décrit au § 8.8.3.1 portant sur la configuration des justificatifs d'identité des clients MEF, ainsi qu'avec les certificats de CA destinés à valider les certificats, comme décrit au § 8.8.3.2 portant sur la configuration d'enregistrement des clients MEF.

Le client MEF et la fonction MEF ont établi un justificatif d'identité maître symétrique (Kpm) et l'identificateur correspondant (KpmID). Le Kpm et le KpmID peuvent avoir été préconfigurés, ou bien le Kpm peut avoir été fourni par un cadre de configuration à distance de la sécurité, le KpmID étant alors établi par la procédure d'enregistrement de client MEF.

NOTE – Dans le cas d'un établissement du Kpm par une configuration à distance, la prise de contact MEF ne peut pas avoir lieu pendant l'enregistrement du client MEF car (a) la fonction MEF ne connaît pas le Kpm avant l'enregistrement du client MEF et (b) le KpmID n'a pas été attribué avant l'enregistrement du client MEF.

Description de la procédure: si le client MEF et la fonction MEF ont établi un justificatif d'identité maître symétrique (Kpm) avec l'identificateur associé (KpmID), ils doivent établir une session TLS ou DTLS en exécutant une prise de contact TLS-PSK conformément au § 10.2.2, avec les détails suivants:

Le paramètre "psk_identity" [IETF RFC 4279] doit être défini sur la valeur de l'identificateur du justificatif d'identité maître (KpmID).

Le paramètre "psk" [IETF RFC 4279] doit être défini sur la valeur du justificatif d'identité maître (Kpm).

S'il est prévu que le client MEF et la fonction MEF s'authentifient au moyen de certificats, ils doivent établir la session TLS ou DTLS en exécutant une prise de contact TLS fondée sur des certificats conformément au § 10.2.2, avec les détails suivants:

Le certificat du serveur TLS doit être le certificat de la fonction MEF. Le client MEF doit vérifier le certificat de la fonction MEF par rapport aux ancres de confiance configurées pour ce certificat, comme décrit au § 8.1.2.5.

Le certificat du client TLS doit être le certificat du client MEF. La fonction MEF doit vérifier le certificat du client MEF par rapport aux ancres de confiance configurées pour ce certificat, comme décrit au § 8.1.2.5.

8.3.5.2.3 Procédure d'enregistrement de client MEF

Finalité: le client MEF s'enregistre auprès de la fonction MEF pour confirmer qu'il souhaite utiliser les services de cette dernière, avec l'autorisation de la partie prenante administratrice.

NOTE – La procédure d'enregistrement de client MEF est équivalente à l'enregistrement d'une CSE ou d'une AE, à la différence que le client MEF s'enregistre auprès de la fonction MEF, et non auprès de la CSE d'enregistrement.

Conditions préalables: le client MEF, la fonction MEF et (le cas échéant) la fonction MEF ont été configurés avec les paramètres décrits au § 8.3.7.

Description de la procédure: le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF en exécutant la procédure de prise de contact MEF décrite au § 8.3.5.2.2. Cette procédure fournit à la fonction MEF une identité authentifiée pour le client MEF.

Le client MEF doit envoyer une demande d'enregistrement de client MEF contenant les informations présentées au Tableau 8.3.5.2.3-1.

Tableau 8.3.5.2.3-1 – Contenu du message de demande d'enregistrement de client MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
expirationTime	Délai au terme duquel l'enregistrement n'est plus valable	1
labels	Libellés facilitant la découverte de l'enregistrement du client MEF	0 ou 1
adminFQDN	FQDN de la partie prenante administratrice, fourni par la configuration de l'enregistrement de client MEF	1

La fonction MEF doit traiter la demande dès sa réception. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. La fonction MEF peut attribuer des valeurs différentes aux paramètres reçus du client MEF, en fonction des instructions de la partie prenante administratrice. Lorsque la demande est traitée avec succès, la fonction MEF doit composer une réponse d'enregistrement de client MEF contenant les informations présentées au Tableau 8.3.5.2.3-2.

Tableau 8.3.5.2.3-2 – Contenu du message de réponse d'enregistrement de client MEF

Paramètre	Description	Multiplicité
MEFClientRegID	Identificateur du relevé d'enregistrement du nouveau client MEF	1
labels	Libellés facilitant la découverte du relevé d'enregistrement du client MEF	0 ou 1
expirationTime	Délai au terme duquel le relevé d'enregistrement du client MEF n'est plus valable	1
MEF Client ID	Identificateur du client MEF	1
adminFQDN	FQDN de la partie prenante administratrice	1

La fonction MEF doit envoyer la réponse au client MEF.

Le client MEF et la fonction MEF doivent stocker les paramètres.

8.3.5.2.4 Procédure de récupération de la configuration du client MEF

Finalité: cette procédure permet à un client MEF de récupérer les configurations des clients MEF fournies par la partie prenante administratrice à la fonction MEF.

Conditions préalables: le client MEF a préalablement exécuté la procédure d'enregistrement de client MEF, pour générer le relevé d'enregistrement de client MEF correspondant.

Le relevé d'enregistrement de client MEF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF, comme décrit à l'étape 1 du § 8.3.5.2.3.

Le client MEF doit envoyer une demande de récupération de configuration de client MEF contenant les informations présentées au Tableau 8.3.5.2.4-1.

Tableau 8.3.5.2.4-1 – Contenu du message de demande de récupération de configuration de client MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
MEFClientRegID	Identificateur du relevé d'enregistrement de client MEF en cours d'actualisation	1

La fonction MEF doit traiter la demande dès sa réception. Si une erreur se produit, notamment s'il n'existe aucune configuration de client MEF actuellement associée au relevé d'enregistrement de client MEF identifié, la fonction MEF doit renvoyer une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit tenter de récupérer la configuration de client MEF actuellement associée au relevé d'enregistrement de client MEF identifié.

La fonction MEF doit composer une réponse de récupération de configuration de client MEF contenant les paramètres suivants.

Tableau 8.3.5.2.4-2 – Contenu du message de réponse de récupération de configuration de client MEF

Paramètre	Description	Multiplicité
MEFClientCfg	Configuration de client MEF actuellement associée au relevé d'enregistrement de client MEF identifié	1

La fonction MEF doit envoyer la réponse au client MEF.

Le client MEF doit appliquer la configuration de client MEF récupérée.

8.3.5.2.5 Procédure d'actualisation d'enregistrement de client MEF

Finalité: cette procédure permet à un client MEF d'actualiser l'enregistrement de client MEF par une prolongation du délai *expirationTime* du relevé d'enregistrement de client MEF et/ou une actualisation des libellés.

Conditions préalables: le client MEF a préalablement exécuté la procédure d'enregistrement de client MEF, pour générer le relevé d'enregistrement de client MEF correspondant.

Le relevé d'enregistrement de client MEF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF, comme décrit à l'étape 1 du § 8.3.5.2.3.

Le client MEF doit envoyer une demande d'actualisation d'enregistrement de client MEF contenant les informations présentées au Tableau 8.3.5.2.3-1.

Tableau 8.3.5.2.5-1 – Contenu du message de demande d'actualisation d'enregistrement de client MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
MEFClientRegID	Identificateur du relevé d'enregistrement de client MEF en cours d'actualisation	1
expirationTime	Délai au terme duquel le relevé d'enregistrement du client MEF n'est plus valable	0 ou 1
labels	Libellés facilitant la découverte du relevé d'enregistrement du client MEF	0 ou 1
NOTE – L'un au moins des paramètres expirationTime et labels doit être présent.		

La fonction MEF doit traiter la demande dès sa réception. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit actualiser le relevé d'enregistrement de client MEF avec les valeurs proposées, à condition d'y être autorisée par la partie prenante administratrice. La fonction MEF peut attribuer des valeurs différentes aux paramètres reçus du client MEF, en fonction des instructions de la partie prenante administratrice.

La fonction MEF doit composer une réponse d'actualisation d'enregistrement de client MEF contenant les informations présentées au Tableau 8.3.5.2.3-2.

Tableau 8.3.5.2.5-2 – Contenu du message de réponse d'actualisation d'enregistrement de client MEF

Paramètre	Description	Multiplicité
expirationTime	Nouvelle valeur du délai au terme duquel le relevé d'enregistrement de client MEF n'est plus valable	0 ou 1
labels	Libellés actualisés, destinés à faciliter la découverte du relevé d'enregistrement du client MEF	0 ou 1
NOTE – La réponse comprend les paramètres expirationTime et/ou labels uniquement si ceux-ci figuraient dans le message de demande correspondant.		

La fonction MEF doit envoyer la réponse au client MEF.

Le client MEF et la fonction MEF doivent stocker les paramètres.

8.3.5.2.6 Procédure de désenregistrement de client MEF

Finalité: cette procédure permet à un client MEF de mettre fin à son enregistrement auprès de la fonction MEF.

Conditions préalables: le client MEF a préalablement exécuté la procédure d'enregistrement de client MEF, pour générer le relevé d'enregistrement de client MEF correspondant.

Le relevé d'enregistrement de client MEF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF, comme décrit à l'étape 1 du § 8.3.5.2.3.

Le client MEF doit envoyer une demande de désenregistrement de client MEF contenant les informations présentées au Tableau 8.3.5.2.3-1.

Tableau 8.3.5.2.6-1 – Contenu du message de demande de désenregistrement de client MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
MEFClientRegID	Identificateur du relevé d'enregistrement de client MEF auquel il est mis fin	1

La fonction MEF doit traiter la demande dès sa réception. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit supprimer les informations associées au relevé d'enregistrement de client MEF identifié.

La fonction MEF doit composer une réponse d'actualisation d'enregistrement de client MEF indiquant que la suppression s'est déroulée avec succès. La fonction MEF doit envoyer la réponse au client MEF.

8.3.5.2.7 Procédure d'enregistrement de clé MEF

Finalité: cette procédure permet à un client MEF source d'établir une clé symétrique avec la fonction MEF, clé qui pourra ensuite être récupérée pour être utilisée par le ou les clients MEF cibles.

Cette procédure implique le client MEF source et la fonction MEF.

Conditions préalables: le client MEF source a reçu (ou déterminé d'une quelconque manière) les informations contenues dans la configuration d'enregistrement de clé MEF. Voir le § 8.3.7.3.

Le client MEF source a exécuté la procédure d'enregistrement de client MEF (§ 8.3.5.2.3) avec la fonction MEF pour la partie prenante administratrice identifiée dans la configuration d'enregistrement de clé MEF.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF source doit établir une session TLS ou DTLS avec la fonction MEF en exécutant la procédure de prise de contact MEF décrite au § 8.3.5.2.2. L'un des effets de cette prise de contact est que la fonction MEF établit une identité authentifiée pour le client MEF source.

Le client MEF source choisit la valeur de la clé de connexion sécurisée M2M (Kc) qui sera distribuée par la fonction MEF. Cette valeur doit être choisie parmi les suivantes:

Le client MEF source génère la valeur de la clé symétrique produite à partir des secrets de la session (D)TLS au moyen d'une exportation de clé TLS ([IETF RFC 5705]), comme décrit au § 10.3.1 "Détails concernant l'exportation de clés TLS".

La valeur de la clé symétrique produite est auto-générée par le client MEF source, indépendamment des secrets de la session (D)TLS.

Le client MEF source doit établir une liste des clients MEF cibles auxquels la fonction MEF est autorisée à communiquer la valeur de la clé symétrique produite.

Dans le cas d'un cadre d'établissement d'association de sécurité (SAEF) ou d'un cadre ESPrim fondés sur une fonction MEF: la liste doit contenir exactement un AE-ID absolu ou un CSE-ID absolu.

Dans le cas d'un cadre ESData fondé sur une fonction MEF: la liste doit contenir un nombre non nul d'AE-ID absolu ou de CSE-ID absolus.

NOTE 1 – La façon dont le client MEF sélectionne la liste des clients MEF cibles dépend de l'application.

Le client MEF source doit envoyer une demande d'enregistrement de clé MEF contenant les informations présentées au Tableau 8.3.5.2.7-1.

Tableau 8.3.5.2.7-1 – Contenu du message de demande d'enregistrement de clé MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
expirationTime	Délai au terme duquel l'enregistrement de clé n'est plus valable	1
labels	Libellés facilitant la découverte de l'enregistrement de clé	0 ou 1
adminFQDN	Identificateur de la partie prenante administratrice	1
SUID	Identificateur d'utilisation de sécurité limitant le cadre de sécurité dans lequel la clé symétrique peut être utilisée	1
targetIDs	(Paramètre facultatif) Liste des identificateurs du groupe initial de clients MEF cibles autorisés à récupérer la clé symétrique	0 ou 1
Key Value	(Paramètre facultatif) Quand il est présent, ce paramètre contient une valeur de clé symétrique de sortie qui est auto-générée par le client MEF source. En l'absence de ce paramètre, le client MEF source et la fonction MEF génèrent la valeur de la clé symétrique au moyen d'un exportateur TLS.	0 ou 1

La fonction MEF doit traiter la demande. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit autoriser l'établissement d'une valeur de clé, sur la base de l'identité authentifiée pour le client MEF source.

NOTE 2 – La présente spécification ne donne aucun détail quant à l'autorisation de cette demande.

Si la demande inclut une valeur dans le paramètre de valeur de clé, la fonction MEF doit stocker cette valeur. Dans le cas contraire, la fonction MEF doit générer la valeur de la clé à partir de la session (D)TLS au moyen de l'outil TLS Key Export ([IETF RFC 5705]), comme décrit au § 10.3.1 "Détails concernant l'exportation de clés TLS".

La fonction MEF doit initialiser la liste des clients MEF cibles autorisés (les clients MEF ayant le droit de récupérer ce justificatif d'identité) à partir de la liste contenue dans la demande.

Dans le cas d'un cadre ESData fondé sur une fonction MEF: La liste peut être modifiée ou complétée par les parties prenantes administratrices pendant ou après la procédure d'enregistrement de clé MEF.

NOTE 3 – La présente spécification ne donne aucun détail sur la façon dont les parties prenantes administratrices peuvent mettre à jour la liste des clients MEF cibles autorisés sur la fonction MEF. La fonction MEF peut fournir sa propre logique et sa propre interface pour permettre à ces parties prenantes d'intervenir sur la liste.

La fonction MEF doit sélectionner une valeur de RelativeKeyID qui n'a jamais été utilisée.

La fonction MEF peut attribuer des valeurs différentes aux paramètres reçus du client MEF, en fonction des instructions de la partie prenante administratrice.

La fonction MEF doit envoyer au client MEF source une réponse contenant les informations présentées au Tableau 8.3.5.2.7-2.

Tableau 8.3.5.2.7-2 – Contenu du message de réponse d'enregistrement de clé MEF

Paramètre	Description	Multiplicité
RelativeKeyID	Partie relative de l'identificateur de clé associé à l'enregistrement de clé	1
expirationTime	Délai au terme duquel l'enregistrement de clé n'est plus valable	1
Source MEF Client ID	Identificateur du client MEF source	1
labels	Libellés facilitant la découverte de l'enregistrement de clé	0 ou 1
adminFQDN	Identificateur de la partie prenante administratrice	1
SUID	Identificateur d'utilisation de sécurité limitant le cadre de sécurité dans lequel la clé symétrique peut être utilisée	
targetIDs	Liste des identificateurs du groupe initial de clients MEF cibles autorisés à récupérer la clé symétrique. Cette liste peut avoir été modifiée par rapport à la liste initialement fournie par le client MEF, ou créée par la fonction MEF en l'absence de liste fournie par le client MEF.	1

Le client MEF source et la fonction MEF doivent stocker la valeur de la clé symétrique produite et l'identificateur de clé correspondant.

L'identificateur de clé est généré à partir de l'identificateur RelativeKeyID et du nom de domaine complet de la fonction d'authentification M2M par le client MEF et la fonction MEF, comme décrit au § 10.3.4 "Génération de l'identificateur KeID".

8.3.5.2.8 Procédure de récupération de clé MEF

Finalité: cette procédure permet à un client MEF cible de récupérer, auprès d'une fonction MEF, la valeur de la clé correspondant à un identificateur RelativeKeyID reçu par le client MEF cible.

Conditions préalables: le client MEF cible a exécuté la procédure de configuration de justificatif d'identité de client MEF (§ 8.3.5.2.1) auprès de la fonction MEF, notamment la configuration de l'URI de récupération de la clé MEF.

Le client MEF source a exécuté la procédure d'enregistrement de clé MEF (§ 8.3.5.2.2) auprès de la fonction MEF, produisant ainsi une valeur de clé enregistrée et un RelativeKeyID pour une partie prenante administratrice et un identificateur d'utilisation de sécurité (SUID) spécifiques.

Le client MEF cible reçoit un identificateur de clé du client MEF d'origine, dans le contexte d'un cadre de sécurité correspondant au SUID que le client MEF source a fourni à la fonction MEF pendant la procédure d'enregistrement de clé MEF (§ 8.3.5.2.7). L'identificateur de clé doit être composé du FQDN de la fonction MEF et de l'identificateur RelativeKeyID attribué à la clé enregistrée.

Le client MEF cible peut s'attendre à être autorisé à obtenir la valeur de la clé symétrique produite correspondante.

NOTE – Le client MEF ne devrait pas répéter cette procédure s'il est déjà en possession de la valeur de clé correspondante.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF cible doit établir une session TLS ou DTLS avec la fonction MEF en exécutant la procédure de prise de contact MEF décrite au § 8.3.5.2.2. L'un des effets de cette prise de contact est que la fonction MEF établit une identité authentifiée pour le client MEF cible.

Le client MEF cible doit envoyer une demande de récupération de clé MEF contenant les informations présentées au Tableau 8.3.5.2.8-1.

Tableau 8.3.5.2.8-1 – Contenu du message de demande de récupération de clé MEF

Paramètre	Description	Multiplicité
RelativeKeyID	Partie relative de l'identificateur de clé reçu du client MEF source dans un cadre de sécurité	1

La fonction MEF doit traiter la demande. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande a été traitée avec succès, la fonction MEF doit identifier l'enregistrement de clé avec le paramètre *RelativeKeyID*.

La fonction MEF doit déterminer si le client MEF cible est autorisé à récupérer la clé enregistrée et ses métadonnées en vérifiant si l'identificateur authentifié de ce client figure sur la liste des clients MEF cibles autorisés. Si le client MEF cible n'est pas autorisé, la fonction MEF doit lui renvoyer un message d'erreur. Dans le cas contraire, la fonction MEF passe à l'étape suivante.

La fonction MEF doit envoyer au client MEF cible une réponse contenant les informations présentées au Tableau 8.3.5.2.8-2.

Tableau 8.3.5.2.8-2 – Contenu du message de réponse de récupération de clé MEF

Paramètre	Description	Multiplicité
expirationTime	Délai au terme duquel l'enregistrement de clé n'est plus valable	1
Source MEF Client ID	Identificateur du client MEF source	1
labels	Libellés facilitant la découverte de l'enregistrement de clé	0 ou 1

Tableau 8.3.5.2.8-2 – Contenu du message de réponse de récupération de clé MEF

Paramètre	Description	Multiplicité
adminFQDN	Identificateur de la partie prenante administratrice	1
SUID	Identificateur d'utilisation de sécurité limitant le cadre de sécurité dans lequel la clé symétrique peut être utilisée	
Key Value	Valeur enregistrée de la clé symétrique de sortie	1

Le client MEF cible doit associer les paramètres à l'identificateur de clé.

8.3.5.2.9 Procédure d'actualisation d'enregistrement de clé MEF

Finalité: cette procédure permet à un client MEF source d'actualiser les métadonnées associées à une clé enregistrée.

Cette procédure implique le client MEF et la fonction MEF.

Conditions préalables: le client MEF a préalablement exécuté la procédure d'enregistrement de client MEF, pour générer le relevé d'enregistrement de clé correspondant.

L'enregistrement de la clé n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF, comme décrit à l'étape 1 du § 8.3.5.2.7.

Le client MEF source doit établir une liste des clients MEF cibles auxquels la fonction MEF est autorisée à communiquer la clé de connexion Kc:

Dans le cas d'un cadre d'établissement d'association de sécurité (SAEF) ou d'un cadre ESPrim fondés sur une fonction MEF: la liste doit contenir exactement un AE-ID absolu ou un CSE-ID absolu.

Dans le cas d'un cadre ESData fondé sur une fonction MEF: la liste doit contenir un nombre non nul d'AE-ID absolu ou de CSE-ID absolus.

NOTE – La présente spécification ne donne aucun détail sur la manière dont le client MEF source compose la liste des clients MEF cibles.

Le client MEF source doit envoyer une demande d'actualisation d'enregistrement de clé MEF contenant les informations présentées au Tableau 8.3.5.2.9-1.

Tableau 8.3.5.2.9-1 – Contenu du message de demande d'actualisation d'enregistrement de clé MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
RelativeKeyID	Partie relative de l'identificateur de clé associé à l'enregistrement de clé	1
expirationTime	Délai au terme duquel l'enregistrement de clé n'est plus valable	0 ou 1
labels	Libellés facilitant la découverte de la clé enregistrée	0 ou 1
targetIDs	(Paramètre facultatif) Liste des identificateurs du groupe de clients MEF cibles autorisés à récupérer la clé symétrique	0 ou 1
NOTE – L'un au moins des paramètres expirationTime, labels et targetIDs doit être présent.		

La fonction MEF doit traiter la demande. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit actualiser les métadonnées avec les valeurs proposées, à condition d'y être autorisée par la partie prenante administratrice. La fonction MEF peut attribuer des valeurs différentes aux paramètres reçus du client MEF, en fonction des instructions de la partie prenante administratrice.

La fonction MEF doit envoyer au client MEF source une réponse contenant les informations présentées au Tableau 8.3.5.2.9-2.

Tableau 8.3.5.2.9-2 – Contenu du message de réponse d'actualisation d'enregistrement de clé MEF

Paramètre	Description	Multiplicité
expirationTime	Valeur actualisée du délai au terme duquel l'enregistrement de clé n'est plus valable, s'il a changé depuis la dernière fois qu'il a été communiqué au client MEF	0 ou 1
labels	Liste actualisée des libellés facilitant la découverte de l'enregistrement de clé, le cas échéant	0 ou 1
targetIDs	Liste actualisée des identificateurs du groupe initial de clients MEF cibles autorisés à récupérer la clé symétrique. Cette liste peut avoir été modifiée par rapport à la liste fournie par le client MEF	0 ou 1
NOTE – La réponse comprend uniquement les paramètres qui figuraient dans le message de demande correspondant.		

8.3.5.2.10 Procédure de désenregistrement de clé MEF

Finalité: cette procédure permet à un client MEF source de demander à la fonction MEF de ne plus distribuer la clé enregistrée.

Cette procédure implique le client MEF et la fonction MEF.

Conditions préalables: le client MEF a préalablement exécuté la procédure d'enregistrement de client MEF, pour générer le relevé d'enregistrement de clé correspondant.

L'enregistrement de la clé n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MEF doit établir une connexion TLS (ou DTLS) avec la fonction MEF, comme décrit à l'étape 1 du § 8.3.5.2.7.

Le client MEF doit envoyer une demande de désenregistrement de clé MEF contenant les informations présentées au Tableau 8.3.5.2.10-1.

Tableau 8.3.5.2.10-1 – Contenu du message de demande de désenregistrement de clé MEF

Paramètre	Description	Multiplicité
MEF-FQDN	FQDN de la fonction MEF, fourni par la configuration des instructions MEF	1
RelativeKeyID	Partie relative de l'identificateur de clé associé à l'enregistrement de clé	1

La fonction MEF doit traiter la demande dès sa réception. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur. Lorsque la demande est traitée avec succès, la fonction MEF doit supprimer les informations associées à l'enregistrement de clé identifié.

La fonction MEF doit composer une réponse de désenregistrement de clé MEF indiquant que la suppression s'est déroulée avec succès. La fonction MEF doit envoyer la réponse au client MEF.

8.3.5.3 Correspondances avec les protocoles de la spécification [UIT-T Y.4500.32]

L'interface Mmef définie dans la Recommandation [UIT-T Y.4500.32] doit être utilisée pour les procédures de configuration de clé symétrique. La correspondance entre les procédures MEF décrites au § 8.3.5.2 et l'interface Mmef est décrite dans la Recommandation [UIT-T Y.4500.32].

8.3.6 Détails concernant la configuration de certificats

8.3.6.1 Introduction

La procédure de configuration de certificats implique les acteurs suivants:

Client MEF: un principal de sécurité qui demande à recevoir un certificat configuré par une fonction MEF. Le client MEF utilise ensuite ce certificat pour pouvoir s'authentifier auprès de la fonction MEF. Le principal de sécurité peut utiliser le certificat configuré par la fonction MEF pour s'authentifier par la suite auprès d'un autre principal de sécurité oneM2M.

CA MEF: émetteur de certificats configurés par la fonction MEF.

Fonction MEF: entité qui traite les demandes émanant d'un client MEF, et qui agit en tant qu'autorité d'enregistrement (RA) pour transférer les demandes de signature de certificat (CSR) à la CA MEF. La fonction MEF peut demander à la CA MEF d'ajouter des attributs à ceux déjà présents dans la CSR, ou de modifier ou de supprimer les attributs déjà présents.

La présente procédure de configuration de certificats porte uniquement sur les interactions entre le client MEF et la fonction MEF.

NOTE 1 – La présente spécification ne donne aucun détail sur les interactions entre la fonction MEF et la CA MEF.

La procédure de configuration de certificats produit les effets suivants:

Le client MEF obtient un certificat configuré par la fonction MEF.

Le client MEF obtient le ou les certificats configurés par la CA MEF. Ces certificats peuvent être utilisés par le client MEF pour valider par la suite les certificats authentifiant la fonction MEF. Ces certificats peuvent être utilisés par le principal de sécurité pour valider par la suite les certificats authentifiant d'autres principaux de sécurité et d'autres fonctions MAF.

NOTE 2 – Il est également possible de configurer des certificats de CA ancres de confiance supplémentaires pour valider d'autres principaux de sécurité et d'autres fonctions MAF, par la configuration d'objets MO fondés sur la ressource [*trustAnchorCred*].

La procédure de configuration de certificats comprend deux procédures:

Procédure de configuration de certificat initial: utilisée lorsque le client MEF ne possède aucun certificat préalablement configuré par la fonction MEF.

Procédure de reconfiguration de certificat: utilisée lorsque le client MEF souhaite renouveler un certificat préalablement configuré par la fonction MEF et déjà en sa possession, ou obtenir une nouvelle clé pour ce certificat.

La présente spécification décrit l'utilisation des protocoles suivants pour la procédure de configuration de certificats:

Protocole d'inscription par transport sécurisé (EST), décrit dans [IETF RFC 7030]. L'utilisation de ce protocole est détaillée au § 8.3.6.2.

Fonctions de configuration de certificat faisant appel au protocole SCEP [IETF RFC 8824]. L'utilisation de ce protocole est décrite au § 8.3.6.3.

8.3.6.2 Procédure de configuration de certificats utilisant le protocole EST

8.3.6.2.1 Introduction

Le protocole d'inscription par transport sécurisée (EST) est décrit dans le document [IETF RFC 7030]. Lorsque le protocole EST est employé par les procédures de configuration de certificats, les correspondances conceptuelles suivantes doivent être appliquées:

Le client MEF joue le rôle du client EST.

La fonction MEF joue le rôle du serveur EST.

La CA MEF joue le rôle de l'autorité de certification EST.

Le certificat configuré par la fonction MEF est équivalent au certificat de client EST.

Si une fonction MEF ou un client MEF indique prendre en charge une procédure de configuration de certificat reposant sur le protocole EST:

La fonction MEF ou le client MEF doit prendre en charge les opérations EST obligatoires et les opérations "CSR Attributes" facultatives. Voir la Figure 5 de [IETF RFC 7030].

La fonction MEF ou le client MEF doit prendre en charge l'authentification du serveur TLS et du client TLS au moyen de certificats, comme précisé pour le protocole EST aux § 3.3.1 et 3.3.2 de [IETF RFC 7030].

NOTE 1 – Ce protocole est employé quand le cadre de sécurité utilisé est un cadre de configuration à distance de la sécurité fondé sur les certificats. Le RSPF fondé sur les certificats ordonne au client MEF et à la fonction MEF d'utiliser uniquement des certificats de CA qui ont été identifiés explicitement pour servir à valider les certificats de fonction MEF et les certificats de client MEF. Ceux-ci correspondent aux ancres de confiance (TA, *trust anchors*) explicites, comme décrit au § 1.1 de la spécification [IETF RFC 7030]. En conséquence, le client MEF/client EST utilise une base de données de TA explicites de client EST et la fonction MEF/client EST utilise une autre base de données de TA explicites de client EST, ces bases de données étant définies à la Figure 4 de [IETF RFC 7030].

Si la fonction MEF ou le client MEF prend en charge un cadre RSPF fondé sur une clé symétrique préconfigurée, cette fonction ou client MEF doit prendre en charge l'authentification mutuelle TLS sans certificat EST décrite au § 3.3.3 de [IETF RFC 7030].

La fonction MEF ou le client MEF peuvent prendre en charge la liaison de l'identité et des informations de preuve de possession. Voir le § 3.5 de [IETF RFC 7030].

NOTE 2 – Dans l'attente de disposer de bibliothèques cryptographiques largement utilisées qui prennent en charge cette fonctionnalité, il est peu probable que cette fonctionnalité soit prise en charge par la fonction MEF ou le client MEF.

La fonction MEF ou le client MEF ne doivent pas utiliser la fonctionnalité d'authentification du client par HTTP du protocole EST décrite au § 3.2.3 de [IETF RFC 7030].

NOTE 3 – L'authentification par HTTP du protocole EST peut être employée dans les cas de figure où le client MEF est autorisé par une procédure d'authentification d'utilisateur comme décrit au § 2.2.3 de [IETF RFC 7030]. De tels cas de figure ne sont pas encore pris en compte dans la présente spécification. Ils pourront être pris en charge à l'avenir en ajoutant la prise en charge de l'authentification des clients fondée sur HTTP.

Le client MEF doit prendre en charge la génération de paires clé privée-clé publique. La fonction MEF et le client MEF doivent utiliser la fonctionnalité de génération de clé côté serveur du protocole EST décrite aux § 2.4 et 4.4 de [IETF RFC 7030].

8.3.6.2.2 Procédure de configuration de certificat initial utilisant le protocole EST

Finalité: permettre à un client MEF de demander son premier certificat à la fonction MEF. Voir également les scénarios opérationnels d'inscription initiale au § 2.2 de [IETF RFC 7030], en prenant note des méthodes d'authentification prises en charge énumérées au § 8.3.6.2.1.

Conditions préalables: conditions préalables communes à toutes les procédures de configuration de certificat:

Le client MEF et la fonction MEF prennent en charge le protocole EST.

Le client MEF dispose du paramètre *estBaseURI*, dont le FQDN doit correspondre à celui de la fonction MEF.

Un événement déclencheur amène le client MEF à exécuter le protocole EST.

NOTE 1 – Le paramètre *estBaseURI* de la condition préalable A.ii peut être fourni dans le message de déclenchement du protocole EST de la condition préalable A.iii.

Le client MEF et la fonction MEF ont exécuté avec succès une prise de contact MEF et la fonction MEF associe un identificateur au client MEF. L'un des cadres RSPF ci-après peut être utilisé pour configurer le certificat initial:

RSPF fondé sur une clé symétrique préconfigurée (§ 8.3.2.1), correspondant à l'authentification TLS sans certificat du protocole EST décrite au § 3.3.3 de la spécification [IETF RFC 7030].

RSPF fondé sur des certificats (§ 8.3.2.2), correspondant à l'authentification mutuelle TLS fondée sur des certificats du protocole EST.

Le certificat utilisé pour authentifier le serveur MEF/EST correspond au certificat de serveur EST (défini à la Figure 3 de [IETF RFC 7030]) que la fonction MEF/le client MEF valide par rapport à la base de données d'ancres de confiance explicites de client EST (voir la Note 1 au § 8.3.6.2.1). L'authentification du serveur EST est décrite au § 3.3.1 de la spécification [IETF RFC 7030], qui impose au client EST de procéder aux vérifications d'autorisation de serveur EST décrites au § 3.6 de cette même spécification. Les vérifications demandées avec une base de données d'ancres de confiance explicites de client EST sont détaillées au § 3.6.1 de [IETF RFC 7030].

Le certificat utilisé pour authentifier le client MEF/client EST correspond au certificat de client EST tiers (défini à la Figure 3 de [IETF RFC 7030]) que la fonction MEF/le serveur MEF valide par rapport à la base de données d'ancres de confiance explicites de client EST (voir la Note 1 au § 8.3.6.2.1). L'authentification du client EST est décrite au § 3.3.2 de la spécification [IETF RFC 7030], qui impose au serveur EST de procéder aux vérifications d'autorisation décrites au § 3.7 de cette même spécification.

NOTE 2 – L'authentification client fondée sur HTTP de l'utilisateur ou du client EST n'est pas prise en charge par la procédure de configuration de certificat. Voir la Note 3 au § 8.3.6.2.1.

Description de la procédure: obtention des certificats des CA constituant les ancres de confiance. Voir le § 4.1 de [IETF RFC 7030].

Le client MEF doit demander le jeu de certificats des CA constituant les ancres de confiance, comme décrit au § 4.1.2 de [IETF RFC 7030].

La fonction MEF doit envoyer en réponse les certificats des CA constituant les ancres de confiance décrits au § 4.1.3 de [IETF RFC 7030].

Le client MEF est censé installer les certificats des CA constituant les ancres de confiance.

NOTE 3 – Le client MEF doit valider le chemin de certification et vérifier le statut des certificats, comme indiqué au § 8.1.2.2.

Obtention du jeu de paramètres de la demande de signature de certificat (CSR). Voir le § 4.5 de [IETF RFC 7030].

Le client MEF doit demander le jeu de paramètres CSR à la fonction MEF, comme décrit au § 4.5.1 de [IETF RFC 7030].

La fonction MEF doit renvoyer en réponse le jeu d'attributs CSR requis, comme décrit au § 4.5.2 de [IETF RFC 7030]. Ces attributs doivent être conformes au profil de CSR décrit en 10.1.4. Ils incluent un attribut *challengePassword* et un attribut d'identité pour le type d'identificateur que la fonction MEF associe au client MEF (voir les conditions préalables).

Obtention d'un certificat:

Le client MEF doit générer une paire composée d'une clé publique et d'une clé privée de longueur adéquate, ou sélectionner une paire clé publique-clé privée existante de longueur adéquate.

Le client MEF doit générer une demande de signature de certificat contenant les attributs CSR requis, au moyen de la paire de clés.

Le client MEF doit demander un certificat de client EST en suivant la procédure "Inscription simple de clients" décrite au § 4.2.1 de [IETF RFC 7030].

La fonction MEF doit valider les attributs, notamment *challengePassword*, en les comparant à ceux de l'étape 2. La fonction MEF doit valider l'attribut d'identité par rapport à l'identité authentifiée associée au client MEF (voir la condition préalable B).

NOTE 3 – La fonction MEF, agissant en tant qu'autorité d'enregistrement (RA), transfère la CSR à une autorité de certificat (CA). La CA émet le certificat de client EST (défini à la Figure 3 de [IETF RFC 7030]) et l'envoie en retour à la fonction MEF.

La fonction MEF doit envoyer le certificat de client EST (défini à la Figure 3 de [IETF RFC 7030]) dans sa réponse au client MEF, comme décrit au § 4.2.3 de [IETF RFC 7030].

Le client MEF est censé installer le certificat de client EST et l'associer à la clé privée correspondante. Ce certificat de client EST devra être utilisé pour une authentification ultérieure avec la fonction MEF. Le certificat de client EST peut aussi être utilisé dans d'autres protocoles de sécurité.

8.3.6.2.3 Procédure de reconfiguration de certificat utilisant le protocole EST

Finalité: permettre à un client MEF de renouveler un certificat inscrit en cours de validité ou les clés associées. Voir également le scénario opérationnel de réémission de certificat client au § 2.3 de [IETF RFC 7030].

Conditions préalables: conditions préalables communes à toutes les procédures de configuration de certificat: voir la condition préalable A au § 8.3.6.2.2.

Le client MEF a déjà effectué la procédure de configuration de certificat initial ou la procédure de reconfiguration de certificat avec la fonction MEF; le client MEF a également installé son certificat de client EST et la base de données d'ancres de confiance explicites suite à la plus récente de ces procédures.

Le client MEF et la fonction MEF ont exécuté une prise de contact MEF pour le cadre RSPF fondé sur des certificats (§ 8.3.2.2), le client MEF ayant utilisé son certificat de client EST et la base de données d'ancres de confiance explicites de client EST, comme détaillé dans la condition préalable B. Les détails sont les mêmes que pour la condition préalable B.ii au § 8.3.6.2.2, à la différence que le client MEF/client EST s'authentifie au moyen d'un certificat de client EST (défini à la Figure 4 de [IETF RFC 7030]) et non d'un certificat de client EST tiers, comme pour la condition préalable B.ii.2.

Description de la procédure:

Obtention des certificats des CA constituant les ancres de confiance. Identique à l'étape 1, § 8.3.6.2.2.

Obtention du jeu de paramètres de la demande de signature de certificat (CSR). Identique à l'étape 2, § 8.3.6.2.2.

Obtention d'un certificat. Similaire à l'étape 3, § 8.3.6.2.2, en remplaçant les Sous-étapes 3.c et 3.d par les sous-étapes suivantes:

c) Le client MEF doit demander le renouvellement ou la nouvelle clé de son certificat de client EST en suivant la procédure "Réinscription simple de clients" décrite au § 4.2.2 de [IETF RFC 7030].

d) La fonction MEF doit valider l'attribut *challengePassword* et le(s) EKU en les comparant à ceux de l'Étape 2. La fonction MEF valide l'identité fournie par rapport à l'identité associée au client MEF (voir la condition préalable C).

8.3.6.3 Procédure de configuration de certificats utilisant le protocole SCEP

8.3.6.3.1 Introduction

Le protocole d'inscription simple par certificat (SCEP) est abordé dans la norme [RFC 8824] de l'IETF. Il existe un grand nombre d'implémentations conformes à la norme [RFC 8824] de l'IETF. Pour documenter les implémentations existantes, la présente Recommandation fait référence à la norme [RFC 8824] du l'IETF.

Lorsque le protocole SCEP est employé par les procédures de configuration de certificats, les correspondances conceptuelles suivantes doivent être appliquées.

La fonction d'inscription M2M (fonction MEF) joue le rôle du client SCEP.

La fonction MEF joue le rôle du serveur SCEP (également appelé répondeur SCEP).

La CA MEF joue le rôle de l'autorité de certification SCEP.

Le certificat configuré par la fonction MEF est équivalent au certificat de client SCEP.

Si une fonction MEF ou un client MEF indique prendre en charge une procédure de configuration de certificat reposant sur le protocole SCEP:

La fonction MEF ou le client MEF peuvent prendre en charge la liaison de l'identité et des informations de preuve de possession.

NOTE 1 – Dans l'attente de disposer de bibliothèques cryptographiques largement utilisées qui prennent en charge cette fonctionnalité, il est peu probable que cette fonctionnalité soit prise en charge par la fonction MEF ou le client MEF.

La fonction MEF ou le client MEF ne doivent pas utiliser l'authentification des clients par HTTP.

NOTE 2 – L'authentification des clients par HTTP du protocole SCEP peut être employée dans les cas de figure où le client MEF est autorisé par une procédure d'authentification d'utilisateur. De tels cas de figure ne sont pas encore pris en compte dans la présente spécification. Ils pourront être pris en charge à l'avenir en ajoutant la prise en charge de l'authentification des clients fondée sur HTTP.

Le client MEF doit prendre en charge la génération de paires clé privée-clé publique.

8.3.6.3.2 Détail des procédures de configuration de certificat utilisant le protocole SCEP

Finalité: permettre à un client MEF d'effectuer la demande de certificat initiale et les demandes de certificats ultérieures auprès de la fonction MEF en employant le protocole SCEP décrit dans le document [IETF RFC 8824].

Conditions préalables: conditions préalables communes à toutes les procédures de configuration de certificat.

Le client MEF et la fonction MEF prennent en charge le protocole SCEP.

Le client MEF connaît l'URI de base, dont le FQDN doit correspondre à celui de la fonction MEF.

Un événement déclencheur amène le client MEF à exécuter le protocole SCEP.

NOTE 1 – L'URI de base de la condition préalable A.ii peut être fourni dans la commande de client MEF qui déclenche le protocole SCEP (voir la condition préalable A.iii).

Si le client ne dispose pas d'un certificat existant approprié, il doit générer localement un certificat auto-signé. L'extension *keyUsage* dans le certificat doit indiquer qu'il est valide pour les actions *digitalSignature* et *keyEncipherment*. Le certificat auto-signé devrait utiliser le même nom de sujet que dans la demande au format PKCS #10. Voir le § 2.4 de [IETF RFC 8824].

Description de la procédure:

Obtention des certificats des CA constituant les ancres de confiance. Voir le § 4.2 de [IETF RFC 8824].

Le client MEF doit demander les certificats des CA ancres de confiance.

La fonction MEF doit renvoyer en réponse les certificats des CA ancres de confiance.

Le client MEF est censé installer les certificats des CA ancres de confiance.

NOTE 2 – Le client MEF doit valider le chemin de certification et vérifier le statut des certificats, comme décrit au § 8.1.2.2.

Obtention des capacités de CA. Voir le § 3.5 de [IETF RFC 8824].

Le client MEF doit demander le jeu de capacités de CA à la fonction MEF, comme décrit au § 3.5.1 de [IETF RFC 8824].

La fonction MEF doit renvoyer en réponse le jeu d'attributs CSR requis, comme décrit au § 3.5.2 de [IETF RFC 8824].

Obtention d'un certificat:

Le client MEF doit générer une paire de clés auto-signée composée d'une clé publique et d'une clé privée de longueur adéquate, ou sélectionner une paire clé publique-clé privée existante de longueur adéquate.

Le client MEF doit générer une demande de signature de certificat contenant les attributs CSR requis, au moyen de la paire de clés.

Le client MEF doit demander un certificat de client SCEP en suivant la procédure "Inscription de certificat" décrite au § 4.3 de [IETF RFC 8824].

La fonction MEF doit valider les attributs, parmi lesquels, facultativement, un attribut *challengePassword*. La fonction MEF doit valider l'attribut d'identité par rapport à l'identité authentifiée associée au client MEF.

NOTE 3 – La fonction MEF, agissant en tant qu'autorité d'enregistrement (RA), transfère la CSR à une autorité de certificat (CA). La CA émet le certificat de client SCEP et l'envoie en retour à la fonction MEF.

La fonction MEF doit envoyer le certificat de client SCEP dans sa réponse au client MEF, comme décrit au § 4.3.1 de [IETF RFC 8824].

Le client MEF est censé installer le certificat de client SCEP et l'associer à la clé privée correspondante. Le certificat de client SCEP doit être utilisé pour une authentification ultérieure avec la fonction MEF. Le certificat de client SCEP peut aussi être utilisé dans d'autres protocoles de sécurité.

8.3.7 Détails concernant la configuration de client MEF

8.3.7.1 Configuration des justificatifs d'identité du client MEF

Le client MEF et la fonction MEF doivent être configurés avec les justificatifs d'identité nécessaires à leur authentification mutuelle.

Les justificatifs d'identité d'authentification mutuelle doivent être préconfigurés, ou configurés à distance par une autre fonction MEF via un cadre de configuration à distance de la sécurité, ou en effectuant une configuration de dispositif conformément à la Recommandation [UIT-T Y.4500.22]. Il est possible de configurer des justificatifs d'identité de type clé symétrique ou certificat. Les deux types de justificatifs peuvent être configurés, pour authentifier différents clients MEF. Le choix peut se fonder sur les capacités de chaque client MEF.

Les détails dépendent du type de justificatif d'identité (clé symétrique ou certificats) et, dans le cas des clés symétriques, du type de configuration (préconfiguration ou configuration à distance).

Détails spécifiques aux **clés symétriques préconfigurées (PPSK)**: la clé d'entité inscrite (Kpm) préconfigurée et l'identificateur de clé correspondant (KpmID) doivent être configurés sur le client MEF qui fait office d'entité inscrite, et sur la fonction MEF.

Détails spécifiques aux **clés symétriques configurées à distance (RPSK)**: le client MEF et la fonction MEF doivent être configurés avec des justificatifs d'identité permettant l'exécution d'un cadre de configuration à distance de la sécurité (RSPF). Le client MEF doit être autorisé à utiliser les services de la fonction MEF. Consulter le § 8.3.2 pour plus d'informations.

NOTE 1 – Dans ce cas de figure, la clé d'entité inscrite (Kpm) préconfigurée et l'identificateur de clé correspondant (KpmID) sont établis pendant la procédure d'enregistrement de client MEF.

Détails spécifiques aux **certificats (qu'ils soient préconfigurés ou configurés à distance)**: le client MEF doit être configuré avec un certificat de client MEF, éventuellement associé à une chaîne de certificats. Le certificat de client MEF doit être un certificat de dispositif, un certificat d'AE-ID ou un certificat de CSE-ID.

NOTE 2 – Les certificats des CA ancres de confiance sont configurés pendant la configuration d'enregistrement du client MEF, éventuellement séparément de la configuration des justificatifs d'identité du client MEF.

La spécification TS-0022 relative à la configuration des dispositifs oneM2M [UIT-T Y.4500.22] fournit plusieurs spécialisations <*mgmtObj*> qu'il faut utiliser pour la configuration des justificatifs d'identité du client MEF lorsque celui-ci prend en charge la gestion des dispositifs (manuelle ou à distance). La présente Recommandation ne traite pas la façon dont est représentée la configuration des justificatifs d'identité du client MEF lorsque celui-ci ne prend pas en charge la gestion des dispositifs.

8.3.7.2 Détails concernant la configuration d'enregistrement du client MEF

Finalité: la configuration d'enregistrement du client MEF décrit les informations configurées sur un client MEF pour lui permettre d'exécuter les procédures MEF autorisées par la partie prenante administratrice. La partie prenante administratrice prend les dispositions nécessaires pour que la configuration d'enregistrement du client MEF soit fournie au client MEF.

Conditions préalables: le client MEF et la fonction MEF ont été configurés avec les justificatifs d'identité qui peuvent servir à l'authentification mutuelle; voir la configuration des justificatifs d'identité du client MEF au § 8.3.7.1.

S'il est prévu que le client MEF et la fonction MEF s'authentifient mutuellement au moyen de certificats:

La partie prenante administratrice (ou une autre partie prenante agissant pour le compte de la partie prenante administratrice) possède une copie des informations de certificat du client MEF, comme défini au § 8.1.2.4. La fonction MEF reçoit une copie des informations de certificat du client MEF. La présente Recommandation ne traite pas la façon dont ces informations sont fournies par la partie prenante administratrice (ou par toute autre partie prenante agissant pour le compte de la partie prenante administratrice).

La partie prenante administratrice (ou une autre partie prenante agissant pour le compte de la partie prenante administratrice) possède une copie des certificats des CA ancres de confiance de la fonction MEF. Le client MEF reçoit une copie des certificats des CA ancres de confiance de la fonction MEF.

La partie prenante administratrice prend les dispositions nécessaires pour que la fonction MEF autorise le client MEF à exécuter la procédure d'enregistrement de client MEF, ce qui peut impliquer une pré-autorisation ou une autorisation en temps réel.

Détails: la configuration d'enregistrement de client MEF (*mefClientRegCfg*) contient les informations présentées au Tableau 8.3.7.2-1, et elle est du type sec:ClientRegCfg (voir le § 12.4.2).

Tableau 8.3.7.2-1 – Informations contenues dans la configuration d'enregistrement de client MEF

Nom de l'élément	Multiplicité	Notes
expirationTime	0 ou 1	Délai au terme duquel la configuration n'est plus valide
labels	0 ou 1	Liste de libellés facilitant la découverte du relevé d'enregistrement du client MEF
fqdn	1	FQDN de la fonction MEF (également appelé identificateur de fonction MEF)
adminFQDN	1	FQDN de la partie prenante administratrice
httpPort	0 ou 1	Numéro du port utilisé avec HTTP [b-IETF RFC 7730]
coapPort	0 ou 1	Numéro du port utilisé avec CoAP [b-IETF RFC 7252]
websocketPort	0 ou 1	Numéro du port utilisé avec WebSocket [b-IETF RFC 6455]

8.3.7.3 Détails concernant la configuration d'enregistrement de clé MEF

Finalité: la configuration d'enregistrement de clé MEF décrit les informations configurées sur un client MEF pour lui permettre d'exécuter les procédures MEF autorisées par la partie prenante administratrice. La partie prenante administratrice prend les dispositions nécessaires pour que la configuration d'enregistrement du client MEF soit fournie au client MEF.

Conditions préalables: le client MEF a exécuté la procédure d'enregistrement de client MEF auprès de la fonction MEF pour la partie prenante administratrice concernée.

Le client MEF dispose de justificatifs d'identité valides pour l'authentification mutuelle avec la fonction MEF.

Détails: la configuration d'enregistrement de clé MEF (*mefClientRegCfg*) contient les informations présentées au Tableau 8.3.7.3-1, et elle est du type sec:keyRegCfg (voir le § 12.4.3).

Tableau 8.3.7.3-1 – Informations contenues dans la configuration d'enregistrement de clé MEF

Nom de l'élément	Multiplicité	Notes
expirationTime	0 ou 1	Délai au terme duquel la clé n'est plus valide
labels	0 ou 1	Liste de libellés facilitant la découverte du relevé d'enregistrement de clé
adminFQDN	1	FQDN de la partie prenante administratrice
SUID	1	SUID limitant l'utilisant de la valeur de clé établie pendant la procédure d'enregistrement de clé MEF
targetIDs	0 ou 1	Liste des identificateurs des clients MEF cibles autorisés

8.3.8 Profil de configuration des dispositifs au sein d'un échange d'inscription

La Recommandation [UIT-T Y.4500.22] décrit une série de types de ressources et les procédures associées pour la configuration des AE et des CSE sur les dispositifs déployés sur le terrain.

Comme expliqué au § 8.3.4.5, il est possible d'exécuter une configuration de dispositif dans le cadre d'un échange d'inscription avec une fonction MEF, ou d'une session DM avec d'autres serveurs DM (session distincte de tout échange d'inscription). Lorsqu'une configuration de dispositif intervient dans un échange d'inscription, deux contraintes s'appliquent aux spécialisations de <*mgmtObj*> de cette configuration de dispositif:

[*myCertFileCred*]: cette spécialisation de <*mgmtObj*> n'est pas configurée par une fonction MEF. Au lieu de cela, une fonction MEF doit utiliser les procédures de configuration de certificat du § 8.3.6 pour configurer un certificat que le client MEF utilisera pour s'authentifier.

[*authenticationProfile*]: l'attribut *symmKeyValue* de cette spécialisation de <*mgmtObj*> destiné à la configuration des clés symétriques n'est pas utilisé par les fonctions MEF (voir la Note ci-dessous). Au lieu de cela, une fonction MEF doit utiliser les procédures de configuration de clé symétrique du § 8.3.5 pour configurer les clés symétriques sur le client MEF. Elle procède en deux étapes:

La fonction MEF s'appuie sur une configuration de dispositif pour configurer un objet MO correspondant à la ressource [*authenticationProfile*], en appliquant les contraintes suivantes:

La ressource [*authenticationProfile*] doit faire le lien avec l'élément [*MEFClientRegCfg*] associé à l'enregistrement de la fonction MEF pour la partie prenante administratrice qui a autorisé la ressource [*authenticationProfile*].

La ressource [*authenticationProfile*] doit comprendre les attributs *expirationTime* et *MAFKeyRegDuration*; elle peut contenir l'attribut *MAFKeyRegLabels.*

La ressource [*authenticationProfile*] ne comprend pas les valeurs correspondant aux attributs *symmKeyID* et *symmKeyValue*.

La fonction MEF émet ensuite une commande client MO_NODE MEF correspondant au nœud d'objet MO [*authenticationProfile*], comme décrit au § 8.3.4.3. Cette commande entraîne l'exécution de la procédure d'enregistrement de clé MEF, qui établit une clé symétrique et un identificateur de clé symétrique au niveau du client MEF et de la fonction MEF.

NOTE – L'attribut *symmKeyValue* est présent dans la ressource [*authenticationProfile*] dans les scénarios de configuration de dispositif où le serveur DM n'est pas une fonction MEF. Les procédures de configuration de clé symétrique abordées dans la présente Recommandation offrent une plus grande sécurité, raison pour laquelle elles sont obligatoires lorsque le serveur DM est une fonction MEF.

8.3.9 Traitement des commandes de client MEF

8.3.9.1 Introduction

Finalité: les commandes de client MEF permettent à une fonction MEF de piloter la séquence de procédures d'échange d'inscription exécutées par le client MEF.

La fonction MEF envoie ou renvoie une commande de client MEF au client MEF en réponse à une procédure de récupération (§ 8.3.9.2) ou d'actualisation (§ 8.3.9.3) de commande de client MEF. L'état qui en résulte, après analyse et exécution de la commande, est renvoyée au client MEF dans une procédure d'actualisation de commande de client MEF (§ 8.3.9.3).

Une réponse de récupération ou d'actualisation de commande de client MEF comprend les attributs *cmdID, cmdDescription* et l'attribut *cmdStatusCode* initial de la commande de client MEF envoyée ou renvoyée au client MEF. Une demande d'actualisation de commande de client MEF, envoyée par un client MEF à une fonction MEF, contient les attributs *cmdID* et *cmdStatusCode* qui informent la fonction MEF du résultat de l'analyse et de l'exécution de la commande.

Ces attributs répondent aux objectifs suivants:

cmdID: permet de distinguer des commandes de client MEF séquentielles émises dans le contexte d'un enregistrement de client MEF (qui, dans certains cas, peut être l'un de multiples enregistrements de client MEF sur la fonction MEF). La finalité de l'attribut *cmdID* est double: s'assurer qu'une commande n'est pas accidentellement exécutée deux fois; et corréler l'état d'une commande de client MEF à la commande émise correspondante.

cmdDescription: fournit une description de la commande à exécuter.

cmdStatusCode: permet à la fonction MEF d'indiquer si une commande est une commande réémise ou non, et permet au client MEF d'indiquer le résultat de la tentative d'analyse et d'exécution de la commande.

Les procédures associées aux commandes de client MEF sont exécutées dans le contexte de l'enregistrement (non expiré) d'un client MEF auprès de la fonction MEF.

8.3.9.2 Procédure de récupération d'une commande de client MEF

Déclenchement de la procédure: voir le § 8.3.4.6, qui décrit les mécanismes pouvant déclencher l'exécution de procédures de commande de client MEF.

Conditions préalables: la fonction MEF et le client MEF ont exécuté une prise de contact MEF (voir le § 8.3.5.2.2).

L'enregistrement du client MEF n'a pas expiré et le client MEF est en possession de l'attribut *MEFClientRegID* de l'enregistrement.

Le client MEF ne doit pas envoyer de demande de récupération de commande de client MEF entre le moment où il reçoit une commande de client MEF émanant de la fonction MEF et le moment où il renvoie l'actualisation correspondante pour informer la fonction MEF de l'état de l'exécution de la commande.

Le client MEF ne doit pas envoyer de demande de récupération de commande de client MEF alors qu'il attend déjà une réponse de commande de client MEF, sauf lorsque la fonction MEF met trop de temps à répondre. Ce temps d'attente de réponse de la fonction MEF est laissé à l'appréciation du client MEF et spécifique à l'implémentation.

Procédure: le client MEF doit envoyer une demande de récupération de commande de client MEF contenant les informations présentées au Tableau 8.3.9.2-1.

Tableau 8.3.9.2-1 – Contenu du message de demande de récupération de commande de client MEF

Élément	Description	Multiplicité
MEFClientRegID	Identificateur du relevé d'enregistrement de client MEF pour lequel la commande de client MEF est demandée. Voir la condition préalable B. Il s'agit de l'identificateur de la ressource <mefClientReg> parente de la commande de client MEF.	1

La fonction MEF doit traiter la demande dès sa réception. En cas d'erreur pendant le traitement, la fonction MEF renvoie une réponse d'erreur.

Lorsque la demande est traitée avec succès, la fonction MEF doit tenter de récupérer la commande de client MEF actuellement associée au relevé d'enregistrement de client MEF identifié.

Si aucune autre commande de client MEF ne doit être envoyée au client MEF, la fonction MEF compose l'élément *cmdDescription* comme décrit au § 8.3.9.6.

Si la commande de client MEF doit déclencher une procédure de configuration de certificat, la fonction MEF compose l'élément *cmdDescription* comme décrit au § 8.3.9.7.

Si la commande de client MEF doit déclencher une configuration de dispositif, la fonction MEF compose l'élément *cmdDescription* comme décrit au § 8.3.9.8.

Dans le cas d'une commande MO_NODE, la fonction MEF compose l'élément *cmdDescription* comme décrit au § 8.3.9.9.

La fonction MEF doit composer une réponse de récupération de commande de client MEF contenant les informations présentées au Tableau 8.3.9.2-2.

Tableau 8.3.9.2-2 – Contenu du message de réponse de récupération de commande de client MEF

Élément	Description	Multiplicité
cmdID	Identificateur de la commande de client MEF envoyée par la fonction MEF (voir la définition des types de données au § 8.6.1 de la Recommandation [UIT-T Y.4500.32])	1
cmdDescription	Description de la commande de client MEF envoyée ou renvoyée	1
cmdStatusCode	Code d'état défini sur MEF_CLIENT_CMD_ISSUED ou sur MEF_CLIENT_CMD_REISSUED, selon le cas (voir les § 8.3.9.5.2 et 8.3.9.5.3)	1

La fonction MEF doit envoyer la réponse au client MEF.

Le client MEF doit tenter d'analyser et d'exécuter les informations contenues dans la réponse.

Le client MEF analyse les informations contenues pour tenter de les interpréter en éléments *cmdID*, *cmdDescription* et *cmdStatusCode*; il interprète de même le contenu de *cmdDescription* dans ses deux éléments *cmdClassID* et *cmdArgs*. Si l'analyse est concluante, le client MEF passe à l'étape 5b. En cas d'échec de l'analyse, le client MEF peut choisir de quitter la procédure, ou de recommencer à l'étape 1.

Le client MEF compare l'élément *cmdID* du message de réponse à l'élément *cmdID* envoyé lors de la procédure d'actualisation de commande de client MEF la plus récente. Si les valeurs de *cmdID* sont différentes, le client MEF passe à l'étape 5.c. Si les valeurs de *cmdID* sont identiques, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_REPEATED_CMD_ID.

Le client MEF interprète l'élément *cmdClassID*, pour déterminer la classe de la commande de client MEF correspondante. Si le client MEF prend en charge la classe déterminée, il peut passer à l'étape 5.d. Dans le cas contraire, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_CLASS_NOT_SUPPORTED.

Le client MEF initie les procédures spécifiques à la classe de commande de client MEF:

Les procédures spécifiques à la classe de commande de client MEF NO_MORE_COMMANDS sont décrites au § 8.3.9.6.

Les procédures spécifiques à la classe de commande de client MEF CERT_PROV sont décrites au § 8.3.9.7.

Les procédures spécifiques à la classe de commande de client MEF DEV_CFG sont décrites au § 8.3.9.8.

Les procédures spécifiques à la classe de commande de client MEF MO_NODE sont décrites au § 8.3.9.9.

8.3.9.3 Procédure d'actualisation de commande de client MEF

Déclenchement de la procédure: le client MEF ne doit déclencher la procédure d'actualisation de commande de client MEF que lorsqu'il y est amené dans le cadre d'une procédure de récupération de commande de client MEF (§ 8.3.9.2), d'une procédure d'actualisation de commande de client MEF (définie dans le présent paragraphe), ou d'une procédure spécifique à une classe de commande de client MEF (§ 8.3.9.6, 8.3.9.7, 8.3.9.8, 8.3.9.9). Le message déclencheur contient des valeurs pour les éléments *cmdID* et *cmdStatusCode.*

Conditions préalables: la fonction MEF et le client MEF ont exécuté une prise de contact MEF (voir le § 8.3.5.2.2).

L'enregistrement du client MEF n'a pas expiré et le client MEF a récupéré l'attribut *MEFClientRegID* de l'enregistrement.

Procédure: le client MEF doit envoyer une demande d'actualisation de commande de client MEF contenant les informations présentées au Tableau 8.3.9.3-1.

Tableau 8.3.9.3-1 – Contenu du message de demande d'actualisation de commande de client MEF

Élément	Description	Multiplicité
MEFClientRegID	Identificateur du relevé d'enregistrement de client MEF pour lequel la commande de client MEF est demandée. Voir la condition préalable B.	1
cmdID	Contient une valeur lorsque la procédure est exécutée suite à un déclenchement.	1
cmdStatusCode	Contient une valeur lorsque la procédure est exécutée suite à un déclenchement.	1

La fonction MEF doit traiter la demande dès sa réception.

La fonction MEF tente d'analyser les informations du message de réponse pour les interpréter en éléments *MEFClientRegID, cmdID* et *cmdStatusCode*. Si l'analyse est concluante, la fonction MEF passe à l'étape 2b. En cas d'échec de l'analyse, la fonction MEF doit envoyer une réponse d'actualisation de commande de client MEF contenant un code d'état de réponse BAD_REQUEST selon le Tableau 5.1.2-3 de la Recommandation [UIT-T Y.4500.32], et la procédure s'arrête là.

La fonction MEF compare la valeur de *cmdID* de la demande à la valeur de *cmdID* de la dernière commande de client MEF émise. Si les valeurs ne correspondent pas, les éléments *cmdID* et *cmdStatusCode* sont rejetées. Si les valeurs correspondent, la fonction MEF peut enregistrer les éléments *cmdID* et *cmdStatusCode.*

La fonction MEF détermine la commande de client MEF suivante à envoyer au client MEF, comme décrit à l'étape 3 du § 8.3.9.2.

La fonction MEF doit composer une réponse d'actualisation de commande de client MEF, qui contiendra les mêmes éléments qu'une réponse de récupération de commande de client MEF (voir le Tableau 8.3.9.2-2 à l'étape 4 du § 8.3.9.2). La fonction MEF doit envoyer la réponse au client MEF.

Le client MEF doit tenter d'analyser et d'exécuter le message de réponse, comme décrit à l'étape 5 du § 8.3.9.2.

8.3.9.4 L'élément cmdDescription

L'élément *cmdDescription* est du type sec:cmdDescription défini au § 12.4.4 et il comprend les éléments suivants:

cmdClassID: identifie la classe des commandes de client MEF.

(conditionné par *cmdClassID*) *cmdArgs:* contient des arguments spécifiques à *cmdClass*.

(facultatif) *targetID*: lorsque le client MEF est un nœud agissant pour le compte d'une CSE et/ou de plusieurs AE, l'élément *targetID* précise à quelle entité s'applique la commande.

Le traitement des commandes de client MEF prend en charge les classes de commande suivantes:

NO_MORE_COMMANDS: indique que la fonction MEF n'a plus de commandes à envoyer au client MEF. La commande de client MEF NO_MORE_COMMANDS est décrite au § 8.3.9.6.

CERT_PROV: déclenche les procédures de configuration de certificat. Les commandes de client MEF CERT_PROV sont décrites au § 8.3.9.7.

DEV_CFG: déclenche une configuration de dispositif. Les commandes de client MEF DEV_CFG sont décrites au § 8.3.9.8.

MO_NODE: déclenche l'exécution de procédures. Les commandes de client MEF MO_NODE sont décrites au § 8.3.9.9.

8.3.9.5 L'élément cmdStatusCode

8.3.9.5.1 Introduction

L'élément *cmdStatusCode* est utilisé par la fonction MEF et le client MEF pour indiquer le statut d'une commande qui a été envoyée. La valeur de l'élément *cmdStatusCode* est conforme au type d'énumération sec:cmdStatusCode, décrit au § 12.3.2.4. Le Tableau 8.3.9.5.1-1 présente un aperçu global de *cmdStatusCode*, la description normative de cet élément étant détaillée dans les sous-paragraphes de 8.3.9.5 ci-après.

Tableau 8.3.9.5.1-1 – Aperçu général de l'élément *cmdStatusCode*

cmdStatusCode	Assigné par	cmdClass de la commande de client MEF envoyée	Paragraphe
MEF_CLIENT_CMD_ISSUED	La MEF	Toute valeur	8.3.9.5.2
MEF_CLIENT_CMD_REISSUED	La MEF	Toute valeur	8.3.9.5.3
MEF_CLIENT_CMD_OK	Client MEF	Toute valeur. Voir Note.	8.3.9.5.4
MEF_CLIENT_CMD_REPEATED_CMD_ID	Le client MEF	Toute valeur	8.3.9.5.5
MEF_CLIENT_CMD_CLASS_NOT_SUPPORTED	Le client MEF	Toute valeur. Voir Note.	8.3.9.5.6
MEF_CLIENT_CMD_BAD_ ARGUMENTS	Le client MEF	Toute valeur	8.3.9.5.7
MEF_CLIENT_CMD_UNACCEPTABLE_ARGUMENTS	Le client MEF	CERT_PROV, DEV_CFG, MO_NODE	8.3.9.5.8
MEF_CLIENT_CMD_CERT_PROV_SERVER_ERROR	Le client MEF	CERT_PROV	8.3.9.5.9

Tableau 8.3.9.5.1-1 – Aperçu général de l'élément *cmdStatusCode*

cmdStatusCode	Assigné par	cmdClass de la commande de client MEF envoyée	Paragraphe
MEF_CLIENT_CMD_CERT_PROV_CLIENT_ERROR	Le client MEF	CERT_PROV	8.3.9.5.10
MEF_CLIENT_CMD_DEV_CFG_SERVER_ERROR	Le client MEF	DEV_CFG	8.3.9.5.11
MEF_CLIENT_CMD_DEV_CFG_CLIENT_ERROR	Le client MEF	DEV_CFG	8.3.9.5.12
MEF_CLIENT_CMD_MO_NODE_NOT_FOUND	Le client MEF	MO_NODE	8.3.9.5.13
MEF_CLIENT_CMD_MO_NODE_TYPE_CONFLICT	Le client MEF	MO_NODE	8.3.9.5.14
MEF_CLIENT_CMD_MO_NODE_BAD_ARGS	Le client MEF	MO_NODE	8.3.9.5.15
MEF_CLIENT_CMD_MO_NODE_UNACCEPTABLE_ARGS	Le client MEF	MO_NODE	8.3.9.5.16
MEF_CLIENT_CMD_MO_NODE_INCONSITENT_CONFIG	Le client MEF	MO_NODE	8.3.9.5.17
MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED	Le client MEF	MO_NODE	8.3.9.5.18
NOTE – En temps normal, un client MEF ne devrait pas fournir cet élément *cmdStatusCode* lorsque la commande émise contient un élément *cmdClass* indiquant NO_MORE_COMMANDS.			

8.3.9.5.2 *cmdStatusCode* MEF_CLIENT_CMD_ISSUED

La fonction MEF envoie la commande, qui est normalement reçue à cette occasion pour la première fois par le client MEF.

8.3.9.5.3 *cmdStatusCode* MEF_CLIENT_CMD_REISSUED

La fonction MEF a déjà envoyé cette commande, mais n'a pas reçu la réponse d'actualisation de commande de client MEF correspondante de la part du client MEF (notamment le statut de la commande exécutée). En conséquence, la fonction MEF émet de nouveau la commande.

Si le client MEF a déjà exécuté la commande, il en renvoie le statut; dans le cas contraire, le client MEF exécute la commande.

8.3.9.5.4 *cmdStatusCode* MEF_CLIENT_CMD_OK

Le client MEF a exécuté la commande avec succès.

8.3.9.5.5 *cmdStatusCode* MEF_CLIENT_CMD_ REPEATED_CMD_ID

L'élément *cmdID* dans le message de réponse à la commande de client MEF correspond au *cmdID* envoyé dans la procédure d'actualisation de commande de client MEF la plus récente. Cette situation indique qu'une erreur de traitement s'est produite au niveau de la fonction MEF.

8.3.9.5.6 *cmdStatusCode* MEF_CLIENT_CMD_CLASS_NOT_SUPPORTED

Le client MEF ne prend pas en charge la classe de commande *cmdClass* demandée.

8.3.9.5.7 *cmdStatusCode* MEF_CLIENT_CMD_BAD_ARGUMENTS

Le client MEF prend en charge la classe de commande *cmdClass*, mais il n'a pas été en mesure d'interpréter l'élément *cmdArgs*.

8.3.9.5.8 *cmdStatusCode* MEF_CLIENT_CMD_UNACCEPTABLE_ARGUMENTS

Le client MEF prend en charge la classe de commande *cmdClass* et a interprété avec succès l'élément *cmdArgs*, mais l'un au moins des éléments de *cmdArgs* traités par le client MEF présente une valeur inacceptable.

8.3.9.5.9 *cmdStatusCode* MEF_CLIENT_CMD_CERT_PROV_SERVER_ERROR

Le client MEF prend en charge la classe de commande *cmdClass* CERT_PROV et a interprété avec succès l'élément *cmdArgs*, mais il n'a pas pu exécuter la commande en raison d'une erreur de communication avec le serveur de configuration des certificats, ou d'une erreur interne de ce même serveur.

8.3.9.5.10 *cmdStatusCode* MEF_CLIENT_CMD_CERT_PROV_CLIENT_ERROR

Le client MEF prend en charge la classe de commande *cmdClass* CERT_PROV et a interprété avec succès l'élément *cmdArgs*, mais une erreur interne l'a empêché d'exécuter la commande.

8.3.9.5.11 *cmdStatusCode* MEF_CLIENT_CMD_DEV_CFG_SERVER_ERROR

Le client MEF prend en charge la classe de commande *cmdClass* DEV_CFG et a interprété avec succès l'élément *cmdArgs*, mais il n'a pas pu exécuter la commande en raison d'une erreur de communication avec le serveur de configuration des dispositifs, ou d'une erreur interne de ce même serveur.

8.3.9.5.12 *cmdStatusCode* MEF_CLIENT_CMD_DEV_CFG_CLIENT_ERROR

Le client MEF prend en charge la classe de commande *cmdClass* DEV_CFG et a interprété avec succès l'élément *cmdArgs*, mais une erreur interne l'a empêché d'exécuter la commande.

8.3.9.5.13 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_NOT_FOUND

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE et a interprété avec succès l'élément *cmdArgs*, mais il n'a pas trouvé le nœud MO au niveau de l'élément *objectPath* dans les arguments de la commande de client MEF.

8.3.9.5.14 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_TYPE_CONFLICT

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE, il a interprété avec succès l'élément *cmdArgs* et trouvé le nœud MO au niveau de l'élément *objectPath*, mais le type du nœud MO ne correspond pas à l'élément *objectType* dans les arguments de la commande de client MEF.

8.3.9.5.15 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_BAD_ARGS

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE, il a interprété avec succès l'élément *cmdArgs* et trouvé le nœud MO au niveau de l'élément *objectPath*, le type du nœud MO correspond bien à l'élément *objectTypeID*, mais le client MEF n'a pas réussi à interpréter l'élément *objectTypeSpecifcArgs*.

8.3.9.5.16 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_UNACCEPTABLE_ARGS

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE, il a interprété avec succès l'élément *cmdArgs* et trouvé le nœud MO au niveau de l'élément *objectPath*, le type du nœud MO correspond bien à l'élément *objectType* et le client MEF a réussi à interpréter l'élément *objectTypeSpecifcArgs*, mais il ne peut pas accepter les autres arguments spécifiques au type d'objet MO de la commande.

8.3.9.5.17 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_INCONSISTENT_CONFIG

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE, il a interprété avec succès l'élément *cmdArgs* et trouvé le nœud MO au niveau de l'élément *objectPath*, le type du nœud MO correspond bien à l'élément *objectType* et le client MEF a réussi à interpréter l'élément *objectTypeSpecifcArgs*, les autres arguments spécifiques au type d'objet MO de la commande sont acceptables, mais la configuration des nœuds MO n'est pas cohérente et empêche le traitement au niveau de ces nœuds.

8.3.9.5.18 *cmdStatusCode* MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED

Le client MEF prend en charge la classe de commande *cmdClass* MO_NODE, il a interprété avec succès l'élément *cmdArgs* et trouvé le nœud MO au niveau de l'élément *objectPath*, le type du nœud MO correspond bien à l'élément objectType et le client MEF a réussi à interpréter l'élément *objectTypeSpecifcArgs*, les autres arguments spécifiques au type d'objet MO de la commande sont acceptables, mais une autre erreur s'est produite pendant l'exécution du traitement au niveau des nœuds MO.

8.3.9.6 Processus spécifiques à la classe NO_MORE_COMMANDS de commande de client MEF

Finalité: lorsque l'élément *cmdClassID* a pour valeur NO_MORE_COMMANDS, la fonction MEF indique qu'elle n'a plus de commandes à envoyer au client MEF.

Éléments de *cmdArgs*: si *cmdDescription* contient un élément *cmdClassID* indiquant NO_MORE_COMMANDS, l'élément *cmdArgs* doit contenir l'élément *noMoreCmdArgs* conforme au type sec:noMoreCmdArgs, lequel inclut l'attribut suivant:

retryDuration: délai au terme duquel le client MEF est censé envoyer une demande de récupération de commande de client MEF. L'élément *retryDuration* est annulé si le client MEF exécute avec succès une nouvelle procédure de traitement de commande de client MEF dans le cadre de l'enregistrement de client MEF, avant expiration du délai *retryDuration*. Voir le § 8.3.4.6, qui décrit d'autres mécanismes pouvant déclencher l'exécution de procédures de commande de client MEF.

Formation de *cmdDescription*: la fonction MEF doit former l'élément *cmdArgs* en y incluant les éléments décrits dans "Éléments de cmdArgs" ci-dessus:

retryDuration: délai au terme duquel la fonction MEF souhaite que le client MEF envoie un nouveau message de récupération de commande de client MEF, l'élément *retryDuration* étant annulé si le client MEF exécute avec succès une nouvelle procédure de commande de client MEF.

La fonction MEF doit former *cmdDescription* en y incluant un élément *cmdClassID* qui indique NO_MORE_COMMANDS et l'élément *cmdArgs* formé à l'étape 1.

Analyse et exécution de *cmdArgs*: la fonction MEF doit tenter d'analyser *cmdArgs* pour en extraire les éléments décrits dans "Éléments de cmdArgs" ci-dessus. Si l'analyse est concluante, le client MEF passe à l'étape 4. En cas d'échec de l'analyse, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_BAD_ARGUMENTS.

Le client MEF n'exécute pas de procédure de récupération ou d'actualisation de commande de client MEF, sauf s'il reçoit un message qui l'y invite.

Le client MEF doit déclencher un minuteur réglé sur la valeur de *retryDuration*.

Ce minuteur est annulé si un quelconque mécanisme décrit au § 8.3.4.6 amène le client MEF à exécuter une procédure de récupération ou d'actualisation de commande de client MEF avant expiration du délai.

Au terme du délai, le client MEF doit exécuter une procédure de récupération de commande de client MEF (§ 8.3.9.2) au moment qu'il jugera bon.

8.3.9.7 Processus spécifiques à la classe CERT_PROV de commande de client MEF

Finalité: lorsque l'élément *cmdClassID* a pour valeur CERT_PROV, la fonction MEF indique que le client MEF doit exécuter une procédure de configuration de certificats avec la fonction MEF.

Éléments de *cmdArgs*: si *cmdDescription* contient un élément *cmdClassID* indiquant CERT_PROV, l'élément *cmdArgs* doit contenir l'élément *certProvArgs* correspondant au type de données sec:certProvArgs, lequel inclut les éléments suivants:

– *certProvProtocol*: précise le protocole de configuration de certificat (EST ou SCEP) que doit utiliser le client MEF;

– *URI*: indique l'URI de base que doit utiliser le protocole de configuration de certificat sélectionné;

– *certSubjectType*: précise si l'objet du certificat configuré est un nœud, une CSE ou une AE;

– *certSubjectID*: identificateur du nœud, de la CSE ou de l'AE objet du certificat.

Formation de *cmdDescription*: la fonction MEF doit former l'élément *cmdArgs* en y incluant les éléments décrits dans "Éléments de cmdArgs" ci-dessus:

– *certProvProtocol*: la fonction MEF doit indiquer dans cet élément le protocole (EST ou SCEP) que devra utiliser le client MEF pour configurer le certificat;

– *URI*: la fonction MEF doit indiquer dans cet élément l'URI de base que devra utiliser le protocole de configuration de certificat sélectionné. Le FQDN de l'URI de base doit correspondre au FQDN de la fonction MEF à l'origine de la commande de client MEF;

– *certSubjectType*: la fonction MEF doit indiquer dans cet élément l'objet du certificat configuré (nœud, CSE ou AE);

– *certSubjectID*: la fonction MEF doit indiquer dans cet élément l'identificateur (de nœud, de CSE ou d'AE) de l'objet du certificat.

La fonction MEF doit former *cmdDescription* en y incluant un élément *cmdClassID* qui indique CERT_PROV et l'élément *cmdArgs* formé à l'étape 1.

Analyse et exécution de *cmdArgs*: voir l'étape 3 du § 8.3.9.6.

Le client MEF doit vérifier que les éléments de *cmdArgs* sont acceptables:

certProvProtocol: la vérification de cet élément réussit uniquement si le protocole indiqué (EST ou SCEP) est pris en charge par le client MEF. Lorsque la vérification est concluante, le client MEF sélectionne le protocole de configuration de certificat indiqué par l'élément.

URI: la vérification de cet élément réussit uniquement si le FQDN de l'URI de base correspond au FQDN de la fonction MEF à l'origine de la commande de client MEF. Si la vérification est concluante, le client MEF définit l'URI de base sur la valeur de cet élément.

certSubjectType: Si le client MEF réside dans un nœud qui agit pour le compte d'une CSE et/ou d'AE multiples, la vérification de cet élément réussit uniquement si la valeur indique un nœud, une CSE ou une AE.

Si le client MEF réside dans une CSE, la vérification de cet élément réussit uniquement si la valeur indique une CSE.

Si le client MEF réside dans une CSE, la vérification de cet élément réussit uniquement si la valeur indique une CSE.

Lorsque la vérification est concluante, le client MEF enregistre l'élément *certSubjectType* en tant que type de sujet de certificat.

certSubjectID: la vérification de cet élément dépend de la valeur de *certSubjectType*:

Si *certSubjectType* indique Node, la vérification de l'élément réussit uniquement si la valeur est un identificateur de nœud.

Si *certSubjectType* indique une CSE, la vérification de cet élément réussit uniquement si la valeur est un identificateur d'AE.

Si *certSubjectType* indique une AE, la vérification de cet élément réussit uniquement si la valeur est un identificateur de CSE.

Lorsque la vérification est concluante, le client MEF enregistre l'élément *certSubjectID* en tant qu'identité de l'objet du certificat.

En cas d'échec de la vérification d'un seul argument, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_UNACCEPTABLE_ARGUMENTS.

Le client MEF doit tenter d'exécuter la procédure de configuration de certificat sélectionnée (via le protocole EST décrit au § 8.3.6.2 ou SCEP décrit au § 8.3.6.3), l'URI de base, le type de sujet de certificat et l'identité de sujet de certificat étant tels que déterminés à l'étape 4. L'identité de l'objet du certificat doit figurer dans l'extension SubjectAltName de la demande de signature de certificat.

Suite à sa tentative d'exécution de la procédure de configuration de certificat choisie, le client MEF doit exécuter une procédure d'actualisation de commande de client MEF, en donnant à l'élément *cmdID* la valeur du *cmdID* de la commande reçue et en assignant comme suit une valeur à *cmdStatusCode*:

Si la procédure de configuration de certificat s'est déroulée avec succès, le client MEF doit donner la valeur MEF_CLIENT_CMD_OK à l'élément *cmdStatusCode*.

Si la procédure de configuration de certificat a échoué en raison d'une erreur de communication avec le serveur de configuration de certificat ou d'une erreur interne de ce même serveur, le client MEF doit donner la valeur MEF_CLIENT_CMD_CERT_PROV_SERVER_ERROR à l'élément *cmdStatusCode*.

Si la procédure de configuration de certificat a échoué en raison d'une erreur sur le client de configuration de certificat (dans le client MEF), le client MEF doit donner la valeur MEF_CLIENT_CMD_CERT_PROV_CLIENT_ERROR à l'élément *cmdStatusCode*.

8.3.9.8 Processus spécifiques à la classe DEV_CFG de commande de client MEF

Finalité: Lorsque *cmdClassID* a pour valeur DEV_CFG, la fonction MEF indique que le client MEF Client doit exécuter une configuration de dispositif ([UIT-T Y.4500.22]), le client MEF faisant alors office de client DM et la fonction MEF de serveur DM.

Éléments de *cmdArgs*: Si *cmdDescription* contient un élément *cmdClassID* indiquant DEV_CFG, l'élément *cmdArgs* doit contenir l'élément *devCfgArgs*, lequel inclut les éléments suivants:

devMgmtID: indique le protocole DM (par exemple OMA DMv1.3, OMA DMv2.0, OMA LwM2M, BBF TR-069) que doit utiliser le client MEF pour la configuration de dispositif.

URI: URI du serveur DM.

Formation de *cmdDescription*: La fonction MEF doit former l'élément *cmdArgs* en y incluant les éléments décrits dans "Éléments de cmdArgs" ci-dessus:

devMgmtID: la fonction MEF doit indiquer dans cet élément le protocole (par exemple OMA DMv1.3, OMA DMv2.0, OMA LwM2M, BBF TR-069) que doit utiliser le client MEF pour la configuration de dispositif.

URI: la fonction MEF doit indiquer dans cet élément l'URI du serveur DM. Le FQDN de l'URI de base doit correspondre au FQDN de la fonction MEF à l'origine de la commande de client MEF.

La fonction MEF doit former *cmdDescription* en y incluant un *cmdClassID* qui indique DEV_CFG et l'élément *cmdArgs* formé à l'étape 1.

Analyse et exécution de *cmdArgs*: voir l'étape 3 du § 8.3.9.6.

Le client MEF doit vérifier que les éléments de *cmdArgs* sont acceptables:

devMgmtID: la vérification de cet élément réussit uniquement si le protocole indiqué (par exemple OMA DMv1.3, OMA DMv2.0, OMA LwM2M, BBF TR-069) est pris en charge par le client MEF. Lorsque la vérification est concluante, le client MEF sélectionne le protocole DM indiqué par l'élément.

URI: la vérification de cet élément réussit uniquement si le FQDN de l'URI de base correspond au FQDN de la fonction MEF à l'origine de la commande de client MEF. Lorsque la vérification est concluante, le client MEF donne à l'URI de serveur DM la valeur de cet élément.

En cas d'échec de la vérification d'un seul argument, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour cet élément *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_UNACCEPTABLE_ARGUMENTS.

Le client MEF doit tenter d'exécuter une configuration de dispositif selon la Recommandation [UIT-T Y.4500.22], en utilisant le protocole DM et l'URI de serveur DM déterminés à l'étape 4.

Suite à sa tentative d'exécution de la procédure de configuration de dispositif, le client MEF doit exécuter une procédure d'actualisation de commande de client MEF, en donnant à l'élément *cmdID* la valeur du *cmdID* de la commande reçue et en assignant comme suit une valeur à *cmdStatusCode*:

Si la procédure de configuration de dispositif s'est déroulée avec succès, le client MEF doit attribuer la valeur MEF_CLIENT_CMD_OK à l'élément *cmdStatusCode*.

Si la procédure de configuration de dispositif a échoué en raison d'une erreur de communication avec le serveur DM ou d'une erreur interne de ce même serveur, le client MEF doit attribuer la valeur MEF_CLIENT_CMD_DEV_CFG_SERVER_ERROR à l'élément *cmdStatusCode*.

Si la procédure de configuration de certificat a échoué en raison d'une erreur sur le client DM (dans le client MEF), le client MEF doit attribuer la valeur MEF_CLIENT_CMD_DEV_CFG_CLIENT_ERROR à l'élément *cmdStatusCode*.

8.3.9.9 Processus spécifiques à la classe MO_NODE de commande de client MEF

8.3.9.9.1 Processus MO_NODE génériques

Finalité: Lorsque *cmdClassID* a pour valeur MO_NODE, la fonction MEF indique que le client MEF doit traiter un MO_NODE déjà configuré sur le client DM du client MEF (par exemple via une configuration de dispositif, comme détaillé dans la Recommandation [UIT-T Y.4500.22]).

Éléments de *cmdArgs*: Si *cmdDescription* contient un élément *cmdClassID* indiquant MO_NODE, l'élément *cmdArgs* doit contenir l'élément *MONodeCmdArgs*, lequel inclut les éléments suivants:

objectPath: chemin du nœud MO à traiter.

objectTypeID: indique le type de spécialisation de la ressource <*mgmtObj*> qui fournit le modèle de données pour le nœud MO à traiter.

(facultatif) objectTypeSpecificArgs: arguments supplémentaires, suivant le type de spécialisation de la ressource <*mgmtObj*> (voir *objectTypeID*).

si *objectTypeID* correspond à la spécialisation [*authenticationProfile*] dans la ressource <*mgmtObj*>, l'élément *objectTypeSpecificArgs* est présent et tel que défini au § 8.3.9.9.2.

L'élément est absent dans tous les autres cas.

Formation de *cmdDescription*: La fonction MEF doit former l'élément *cmdArgs* en y incluant les éléments décrits dans "Éléments de cmdArgs" ci-dessus:

objectPath: la fonction MEF doit indiquer dans cet élément le chemin du nœud MO à traiter.

objectTypeID: la fonction MEF doit indiquer dans cet élément l'identificateur du type de nœud MO à traiter.

(facultatif) objectTypeSpecificArgs: Si *objectTypeID* correspond à la spécialisation [*authenticationProfile*] dans la ressource <*mgmtObj*>, l'élément *objectTypeSpecificArgs* est présent et tel que défini au § 8.3.9.9.2.

L'élément est absent dans tous les autres cas.

La fonction MEF doit former *cmdDescription* en y incluant un *cmdClassID* qui indique MO_NODE et l'élément *cmdArgs* formé à l'étape 1.

Analyse et exécution de *cmdArgs*: voir l'étape 3 du § 8.3.9.6.

Le client MEF doit vérifier que les éléments de *cmdArgs* sont acceptables:

objectPath: la vérification de cet élément réussit uniquement si un chemin de nœud MO est indiqué dans *objectPath*. Lorsque la vérification est concluante, le client MEF passe à l'étape 4b. En cas d'échec de la vérification de cet argument, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_NOT_FOUND.

objectTypeID: la vérification de cet élément réussit uniquement si *objectTypeID* correspond au type de nœud MO dont le chemin est indiqué dans *objectPath* (voir l'étape 4a). Lorsque la vérification est concluante, le client MEF passe à l'étape 5. En cas d'échec de la vérification de cet argument, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_TYPE_CONFLICT.

Le client MEF applique le traitement spécifique à l'*objectTypeID*:

Si *objectTypeID* correspond à la spécialisation [*authenticationProfile*] dans la ressource <*mgmtObj*>, le client MEF doit exécuter la procédure "Traitement d'un nœud MO [*authenticationProfile*]" décrite au § 8.3.9.9.2.

Si *objectTypeID* correspond à la spécialisation [*trustAnchorCred*] dans la ressource <*mgmtObj*>, le client MEF doit exécuter la procédure "Traitement d'un nœud MO [*trustAnchorCred*]" décrite au § 8.3.9.9.7.

Si *objectTypeID* correspond à la spécialisation [*MAFClientCfgReg*] dans la ressource <*mgmtObj*>, le client MEF exécute la procédure "Traitement d'un nœud MO [MAFClientCfgReg]" décrite au § 8.3.9.9.8.

8.3.9.9.2 Processus spécifiques à [*authenticationProfile*]

Finalité: Le traitement d'un nœud MO [*authenticationProfile*] assure que le client MEF a pu établir les justificatifs d'identité nécessaires pour pouvoir utiliser ce nœud MO [*authenticationProfile*] à des fins d'authentification mutuelle.

Éléments de *objectTypeSpecificArgs*: Lorsque *objectTypeID* correspond à la spécialisation [*authenticationProfile*] de la ressource <*mgmtObj*>, les éléments *objectTypeSpecificArgs* doivent être présents contenir l'élément *authProfileMONodeArgs*, lequel contient les éléments suivants:

SUID: doit correspondre au SUID du nœud MO concerné.

Formation de *objectTypeSpecificArgs*: La fonction MEF doit former l'élément *objectTypeSpecificArgs* contenant *authProfileMONodeArgs* en y incluant les éléments décrits dans "Éléments de *objectTypeSpecificArgs*" ci-dessus:

SUID: la fonction MEF doit donner à cet élément la valeur de l'élément *SUID* attendue dans le nœud MO situé à l'adresse *objectPath* sur le client MEF.

Traitement d'un nœud MO [*authenticationProfile*]: La fonction MEF doit tenter d'analyser *objectTypeSpecificArgs* pour en extraire les éléments décrits dans "Éléments de *objectTypeSpecificArgs*". Si l'analyse est concluante, le client MEF passe à l'étape 3. En cas d'échec de l'analyse, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MN_NODE_BAD_ARGS.

Le client MEF doit vérifier que les éléments de *objectTypeSpecificArgs* sont acceptables:

SUID: la vérification réussit si le SUID dans *objectTypeSpecificArgs* correspond au *SUID* du nœud MO_NODE concerné.

Lorsque la vérification de *objectTypeSpecificArgs* est concluante, le client MEF passe à l'étape 5. En cas d'échec de la vérification de *objectTypeSpecificArgs,* le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_UNACCEPTABLE_ARGS.

Le client MEF doit vérifier que le *SUID* correspond à la configuration du nœud MO [*authenticationProfile*] et de ses nœuds MO parents et enfants.

Si le *SUID* fait partie de l'ensemble {11, 21, 31, 41}, la vérification doit échouer si le nœud MO parent du nœud MO [*authenticationProfile*] ne correspond pas à la spécialisation [*MAFClientRegCfg*] de la ressource <*mgmtObj*>.

Si le *SUID* fait partie de l'ensemble {12, 22, 32, 42}, la vérification doit échouer si le nœud MO parent du nœud MO [*authenticationProfile*] ne correspond pas à la spécialisation [*registration*] de la ressource <*mgmtObj*>.

Si le *SUID* fait partie de l'ensemble {13, 23, 33, 43}, la vérification doit échouer si le nœud MO parent du nœud MO [*authenticationProfile*] ne correspond pas à la spécialisation [*dataCollection*] de la ressource <*mgmtObj*>.

Si le *SUID* fait partie de l'ensemble {11, 12, 13}, la vérification doit échouer si l'attribut *symmKeyID* est absent du nœud MO [*authenticationProfile*].

Si le *SUID* fait partie de l'ensemble {21, 22, 23}, la vérification doit échouer si:

l'attribut *keyRegDuration* est absent du nœud MO [*authenticationProfile*], ou

le nœud MO enfant du nœud MO [*authenticationProfile*] ne correspond pas à la spécialisation [*MEFClientRegCfg*] de la ressource <*mgmtObj*>.

Si le *SUID* fait partie de l'ensemble {31, 32, 33}, la vérification doit échouer si:

l'attribut *keyRegDuration* est absent du nœud MO [*authenticationProfile*], ou

le nœud MO enfant du nœud MO [*authenticationProfile*] ne correspond pas à la spécialisation [*MAFClientRegCfg*] de la ressource <*mgmtObj*>.

Si le *SUID* fait partie de l'ensemble {11, 12, 21, 22, 31, 32}, la vérification doit échouer si:

l'attribut *TLSCiphersuites* est absent du nœud MO [*authenticationProfile*], ou

l'attribut *TLSCiphersuites* est présent mais il ne contient pas les systèmes cryptographiques DTLS ou TLS requis par les cadres de sécurité PSK-TLS au § 10.2.2.

Si le *SUID* fait partie de l'ensemble {41, 41,42}, la vérification doit échouer si:

l'attribut *myCertFingerprint* est absent du nœud MO [*authenticationProfile*], ou

l'attribut *TLSCiphersuites* est absent du nœud MO [*authenticationProfile*], ou l'attribut *TLSCiphersuites* est présent mais il ne contient pas les systèmes cryptographiques DTLS ou TLS requise par les cadre de sécurité fondés sur des certificats au § 10.2.3, ou

le nœud MO [*authenticationProfile*] possède un ou plusieurs nœuds MO enfants correspondant à la spécialisation [*trustAnchorCred*] de la ressource <*mgmtObj*>.

Lorsque la vérification est concluante, le client MEF passe à l'étape 5. En cas d'échec de la vérification d'un quelconque élément de cet argument, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_INCONSISTENT_CONFIG.

Le client MEF applique le traitement spécifique au *SUID*:

Si le *SUID* fait partie de l'ensemble {11, 12, 13}, correspondant à un SUID de clé symétrique préconfigurée, le client MEF exécute l'opération "Traitement de nœud MO [*authenticationProfile*] avec clé symétrique préconfigurée" décrite au § 8.3.9.9.3.

Si le *SUID* fait partie de l'ensemble {21, 22, 23}, correspondant à un SUID de clé symétrique établie par une fonction MEF, le client MEF exécute l'opération "Traitement de nœud MO [*authenticationProfile*] avec clé symétrique établie par fonction MEF" décrite au § 8.3.9.9.4.

Si le *SUID* fait partie de l'ensemble {31, 32, 33}, correspondant à un SUID de clé symétrique établie par une fonction MAF, le client MEF exécute l'opération "Traitement de nœud MO [*authenticationProfile*] avec clé symétrique établie par fonction MAF" décrite au § 8.3.9.9.5.

Si le *SUID* fait partie de l'ensemble {41, 42, 43}, correspondant à un SUID de certificat, le client MEF exécute l'opération "Traitement de nœud MO [*authenticationProfile*] avec certificat" décrite au § 8.3.9.9.6.

8.3.9.9.3 Traitement de nœud MO [*authenticationProfile*] avec clé symétrique préconfigurée

Finalité: Le traitement d'un nœud MO [*authenticationProfile*] avec un SUID de clé symétrique préconfigurée (faisant partie de l'ensemble {11, 12, 13}) assure que le client MEF a accès à une copie locale de la clé symétrique préconfigurée, pour pouvoir l'utiliser par la suite avec le nœud MO [*authenticationProfile*].

Conditions préalables: L'opération réussit uniquement s'il existe une copie locale de la valeur de la clé symétrique préconfigurée à laquelle peut accéder la fonction MEF.

Procédure: Le client MEF doit déterminer si l'attribut *symmKeyValue* est présent dans le nœud MO [*authenticationProfile*].

Si l'attribut est absent, le client MEF passe à l'étape 2.

Si l'attribut est présent, le client MEF a accès à la valeur de la clé symétrique préconfigurée. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

Le client MEF récupère la valeur de l'attribut *symmKeyID* du nœud MO [*authenticationProfile*]. Le client MEF détermine s'il dispose d'une copie locale de la valeur de la clé symétrique préconfigurée dont l'identificateur correspond à l'attribut *symmKeyID*.

En cas d'absence de cette copie locale, le client MEF n'a pas accès à la valeur de la clé symétrique préconfigurée. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

Si une copie locale de la valeur de la clé symétrique préconfigurée est présente, le client MEF peut accéder à cette valeur. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

8.3.9.9.4 Traitement de nœud MO [*authenticationProfile*] avec clé symétrique établie par fonction MEF

Finalité: Le traitement d'un nœud MO [*authenticationProfile*] avec un SUID de clé symétrique établie par fonction MEF (faisant partie de l'ensemble {21, 22, 23}) assure que le client MEF établit une clé symétrique avec la fonction MEF pour pouvoir l'utiliser par la suite avec le nœud MO [*authenticationProfile*].

Conditions préalables: Cette opération se déroule avec succès uniquement si le nœud MO [*authenticationProfile*] possède un nœud MO enfant [*MEFClientRegCfg*] et un nœud MO parent qui peut être de type [*MAFClientRegCfg*], [*registration*] ou [*dataCollection*].

Cette procédure suppose que le client MEF dispose d'un enregistrement de client MEF en cours de validité auprès de la fonction MEF et de la partie prenante administratrice identifiées dans le nœud MO enfant [*MEFClientRegCfg*].

Procédure: Le client MEF doit tenter d'exécuter la procédure d'enregistrement de clé MEF comme décrit au § 8.3.5.2.7, le client MEF faisant office de client MEF source et les clarifications suivantes étant apportées:

À l'étape 4 du § 8.3.5.2.7, le client MEF doit formuler une demande d'enregistrement de clé MEF (voir le Tableau 8.3.5.2.7-1) de la façon suivante:

MEFFQDN: doit recevoir la valeur de l'attribut *fqdn* du nœud MO enfant [*MEFClientRegCfg*];

expirationTime: doit être calculé en ajoutant, à l'heure actuelle, la valeur de *keyRegDuration* dans le nœud MO [*authenticationProfile*];

labels: doit recevoir la valeur de l'attribut *keyRegLabels* du nœud MO [*authenticationProfile*];

adminFQDN: doit recevoir la valeur de l'attribut *adminFQDN* du nœud MO enfant [*MEFClientRegCfg*];

SUID: doit recevoir la valeur de l'attribut *SUID* du nœud MO enfant [*authenticationProfile*];

targetIDs: sa valeur dépend du nœud MO parent du nœud MO [*authenticationProfile*]:

Dans le cas d'un nœud MO parent [*registration*], *targetIDs* reçoit comme valeur le CSE-ID de la CSE d'enregistrement.

Dans le cas d'un nœud MO parent [*dataCollection*], *targetIDs* reçoit comme valeur le CSE-ID déterminé à partir de l'attribut *containerPath* du nœud MO [*dataCollection*].

Dans le cas d'un nœud MO parent [*MAFClientRegCfg*], *targetIDs* reçoit comme valeur l'attribut *fqdn* du nœud MO [*MAFClientRegCfg*].

keyValue: ne doit pas être présent.

Le client MEF doit choisir un protocole (HTTP, CoAP, WebSocket) sur la base des protocoles qu'il prend en charge et des protocoles que prend en charge la fonction MEF, indiqués par la présence d'attributs *httpPort*, *coapPort* et *websocketPort* dans le nœud MO enfant [*MEFClientRegCfg*]. Le client MEF doit utiliser le numéro de port indiqué dans l'attribut *httpPort*, *coapPort* ou *websocketPort* adéquat du nœud MO enfant [*MEFClientRegCfg*].

À l'étape 10 du § 8.3.5.2.7, si le client MEF reçoit une réponse positive d'enregistrement de clé MEF (voir le Tableau 8.3.5.2.7-2), il doit vérifier les informations suivantes dans le message de réponse:

expirationTime: la vérification échoue si cette valeur est antérieure à l'heure de la présente vérification.

Source MEF Client ID: la vérification échoue si cette valeur est différente de l'identificateur du client MEF.

adminFQDN: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.3.5.2.7.

SUID: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.3.5.2.7.

targetIDs: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.3.5.2.7.

Si la vérification est concluante, le client MEF doit stocker les informations contenues dans la réponse d'enregistrement de clé MEF et associer ces informations au nœud MO [*authenticationProfile*]. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

En cas d'échec de la vérification, ou de l'ensemble de la procédure pour une autre raison, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

8.3.9.9.5 Traitement de nœud MO [*authenticationProfile*] avec clé symétrique établie par fonction MAF

Finalité: Le traitement d'un nœud MO [*authenticationProfile*] avec un SUID de clé symétrique établie par fonction MEF (faisant partie de l'ensemble {31, 33, 33}) assure que le client MEF, agissant comme client MAF source, établit une clé symétrique avec la fonction MAF pour pouvoir l'utiliser par la suite avec le nœud MO [*authenticationProfile*].

Conditions préalables: cette opération se déroule avec succès uniquement si le nœud MO [*authenticationProfile*] possède un nœud MO enfant [*MAFClientRegCfg*] et un nœud MO parent qui peut être de type [*registration*] ou [*dataCollection*].

Cette procédure suppose que le client MEF, agissant en tant que client MAF, dispose d'un enregistrement de client MAF valide auprès de la fonction MEF et de la partie prenante administratrice identifiée dans le nœud MO enfant [*MEFClientRegCfg*].

Procédure: le client MEF tente d'exécuter la procédure d'enregistrement de clé MAF comme décrit au § 8.8.2.7, le client MAF faisant office de client MAF source et les clarifications suivantes étant apportées:

À l'étape 4 du § 8.8.2.7, le client MEF doit formuler une demande d'enregistrement de clé MAF (voir le Tableau 8.8.2.7-1) de la façon suivante:

– *MAF-FQDN*: doit recevoir la valeur de l'attribut *fqdn* du nœud MO enfant [*MAFClientRegCfg*];

– *expirationTime*: doit être calculé en ajoutant, à l'heure actuelle, la valeur de *keyRegDuration* dans le nœud MO [*authenticationProfile*];

– *labels*: doit recevoir la valeur de l'attribut *keyRegLabels* du nœud MO [*authenticationProfile*];

– *adminFQDN*: doit recevoir la valeur de l'attribut *adminFQDN* du nœud MO enfant [*MAFClientRegCfg*];

– *SUID*: doit recevoir la valeur de l'attribut *SUID* du nœud MO enfant [*authenticationProfile*];

targetIDs: sa valeur dépend du nœud MO parent du nœud MO [*authenticationProfile*]:

Dans le cas d'un nœud MO parent [*registration*], *targetIDs* reçoit comme valeur le CSE-ID de la CSE d'enregistrement.

Dans le cas d'un nœud MO parent [*dataCollection*], *targetIDs* reçoit comme valeur le CSE-ID déterminé à partir de l'attribut *containerPath* du nœud MO [*dataCollection*].

keyValue: ne doit pas être présent.

Le client MEF doit choisir un protocole (HTTP, CoAP, WebSocket) sur la base des protocoles que prend en charge le client MAF et des protocoles que prend en charge la fonction MAF, indiqués par la présence d'attributs *httpPort*, *coapPort* et *websocketPort* dans le nœud MO enfant [*MAFClientRegCfg*]. Le client MEF doit utiliser le numéro de port indiqué dans l'attribut *httpPort*, *coapPort* ou *websocketPort* adéquat du nœud MO enfant [*MAFClientRegCfg*].

À l'étape 10 du § 8.8.2.7, si le client MEF reçoit une réponse positive d'enregistrement de clé MAF (voir le Tableau 8.8.2.7-2), il doit vérifier les informations suivantes dans le message de réponse:

– *expirationTime:* la vérification échoue si cette valeur est antérieure à l'heure de la présente vérification.

– *Source MEF Client ID:* la vérification échoue si cette valeur est différente de l'identificateur du client MEF.

– *adminFQDN*: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.8.2.7.

– *SUID*: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.8.2.7.

– *targetIDs*: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MEF à l'étape 4 du § 8.8.2.7.

Si la vérification est concluante, le client MEF doit stocker les informations contenues dans la réponse d'enregistrement de clé MEF et associer ces informations au nœud MO [*authenticationProfile*]. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

En cas d'échec de la vérification, ou de l'ensemble de la procédure pour une autre raison, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

8.3.9.9.6 Traitement de nœud MO [*authenticationProfile*] avec certificat

Finalité: le traitement d'un nœud MO [*authenticationProfile*] avec un SUID de certificat (faisant partie de l'ensemble {41, 42, 43}) assure que le client MEF a accès à une copie locale du certificat et de la clé privée correspondante, pour pouvoir les utiliser par la suite avec le nœud MO [*authenticationProfile*].

Conditions préalables: cette opération réussit uniquement si le nœud a été configuré avec le certificat correspondant à l'attribut *myCertFingerprint* du nœud MO [*authenticationProfile*].

Procédure: le client MEF récupère la valeur de l'attribut *myCertFingerprint* du nœud MO [*authenticationProfile*]. Le client MEF détermine s'il dispose d'une copie locale de la clé privée correspondant au certificat, identique à la clé privée correspondante de l'attribut *myCertFingerprint*.

En l'absence d'une copie locale de ces éléments, le client MEF associe le certificat et la clé privée correspondante au nœud MO [*authenticationProfile*] pour pouvoir les utiliser ultérieurement. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

S'il existe une copie locale du certificat et de la clé privée correspondante, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

8.3.9.9.7 Processus spécifiques à [*trustAnchorCred*]

Finalité: le traitement d'un nœud MO [*trustAnchorCred*] assure que le client MEF dispose d'une copie locale du certificat de la CA constituant l'ancre de confiance identifiée par le nœud MO [*trustAnchorCred*].

Éléments de *objectTypeSpecificArgs*: l'élément *objectTypeSpecificArgs* est absent dans le cas de la spécialisation [*trustAnchorCred*].

Traitement d'un nœud MO [*trustAnchoCred*]: le client MEF récupère la valeur de l'attribut *certFingerprint* du nœud MO [*trustAnchorCred*]. Le client MEF détermine s'il dispose d'une copie locale d'un certificat correspondant à l'attribut *certFingerprint*.

Si une copie locale de ce certificat est effectivement présente, le client MEF associe le certificat au nœud MO [*trustAnchorCred*] pour pouvoir l'utiliser ultérieurement. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

En l'absence d'une copie locale du certificat, le client MEF passe à l'étape 2.
Le client MEF récupère la valeur de l'attribut *URI* du nœud MO [*trustAnchorCred*]. Le client MEF tente d'obtenir le certificat de l'ancre de confiance en exécutant une méthode HTTPS GET adressée à l'*URI*.

En cas d'échec de la méthode HTTPS GET, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

Si la méthode HTTPS GET entraîne une réponse positive, le client MEF extrait la charge utile de la réponse. Le client MEF analyse la charge utile pour déterminer s'il s'agit d'un certificat; si c'est le cas, le client MEF vérifie que le certificat reçu correspond à l'attribut *certFingerprint* du nœud MO [*trustAnchorCred*].

Si l'analyse est concluante et que le certificat reçu correspond à l'attribut *certFingerprint* du nœud MO [*trustAnchorCred*], le client MEF associe ce certificat au nœud MO [*trustAnchorCred*] pour pouvoir l'utiliser ultérieurement. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

Dans le cas contraire, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID,* l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

8.3.9.9.8 Processus spécifiques à [*MAFClientRegCfg*]

Finalité: le traitement d'un nœud MO [*MAFClientRegCfg*] assure que le client MEF, agissant comme un client MAF, s'est enregistré avec succès auprès de la fonction MAF au moyen des attributs du nœud MO [*MAFClientRegCfg*].

Éléments de *objectTypeSpecificArgs*: l'élément *objectTypeSpecificArgs* est absent dans le cas de la spécialisation [*MAFClientRegCfg*].

Traitement d'un nœud MO [*MAFClientRegCfg*]: le client MEF doit tenter d'exécuter la procédure d'enregistrement de client MAF comme décrit au § 8.8.2.3, le client MAF faisant office de client MAF source et les clarifications suivantes étant apportées:

À l'étape 2 du § 8.8.2.3, le client MEF doit formuler une demande d'enregistrement de client MAF (voir le Tableau 8.8.2.3-1) à partir des attributs du nœud MO [*MAFClientRegCfg*], de la façon suivante:

– *MAF-FQDN*: doit recevoir la valeur de l'attribut *fqdn*;

– *expirationTime*: doit recevoir la valeur de l'attribut *expirationTime*;

– *labels*: doit recevoir la valeur de l'attribut *labels*;

– *adminFQDN*: doit recevoir la valeur de l'attribut *adminFQDN*.

Le client MEF doit choisir un protocole (HTTP, CoAP, WebSocket) sur la base des protocoles que prend en charge le client MAF et des protocoles que prend en charge la fonction MAF, indiqués par la présence d'attributs *httpPort*, *coapPort* et *websocketPort* dans le nœud MO enfant [*MAFClientRegCfg*]. Le client MEF doit utiliser le numéro de port indiqué dans l'attribut *httpPort*, *coapPort* ou *websocketPort* adéquat du nœud MO enfant [*MAFClientRegCfg*].

À l'étape 3 du § 8.8.2.3, si le client MEF reçoit une réponse positive d'enregistrement de clé MAF (voir le Tableau 8.8.2.3-2), il doit vérifier les informations suivantes dans le message de réponse:

– *expirationTime:* la vérification échoue si cette valeur est antérieure à l'heure de la présente vérification.

– *MAF Client ID:* la vérification échoue si cette valeur est différente de l'identificateur du client MEF.

– *adminFQDN*: la vérification échoue si cette valeur est différente de la valeur correspondante envoyée dans la demande d'enregistrement de clé MAF à l'étape 2 du § 8.8.2.3.

Si la vérification est concluante, le client MEF doit stocker les informations contenues dans la réponse d'enregistrement de clé MEF et associer ces informations au nœud MO [*MAFClientRegCfg*]. Le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_OK.

En cas d'échec de la vérification, ou de l'ensemble de la procédure pour une autre raison, le client MEF quitte la procédure, ce qui déclenche la procédure d'actualisation de commande de client MEF pour ce *cmdID*, l'élément *cmdStatusCode* prenant alors la valeur MEF_CLIENT_CMD_MO_NODE_PROCESSING_FAILED.

8.4 Cadre de sécurité de bout en bout des primitives (ESPrim)

8.4.1 Finalité du cadre de sécurité de bout en bout des primitives (ESPrim)

Le cadre de sécurité de bout en bout des primitives (ESPrim) vise à assurer la sécurité les primitives oneM2M de sorte que les entités de services communs (qui transmettent lesdites primitives) n'aient pas besoin de garantir leur confidentialité et leur intégrité. Le cadre ESPrim présente des fonctions d'authentification mutuelle, de confidentialité et de protection de l'intégrité en plus de vérifier l'actualité des objets ESPrim (grâce à leur durée d'existence).

Les cas d'utilisation et les exigences correspondants sont exposés dans le document oneM2M TR-0012 [b-oneM2M TR0012].

Dans la présente Recommandation, on suppose que les points d'extrémité du cadre ESPrim sont l'expéditeur et le destinataire de la primitive.

La présente Recommandation définit les aspects du cadre ESPrim liés à la gestion des justificatifs d'identité et à la protection des données. L'acheminement des objets ESPrim est décrit dans la Recommandation [ITU-T Y.4500.1].

8.4.2 Architecture du cadre de sécurité de bout en bout des primitives (ESPrim)

Le présent paragraphe détaille les aspects du cadre ESPrim liés à la gestion des justificatifs d'identité et à la protection des données. Le paragraphe 11.3.2 de la Recommandation [ITU-T Y.4500.1] décrit l'acheminement des objets ESPrim.

La primitive à sécuriser est dénommée *primitive interne* et la primitive servant à acheminer une primitive interne sécurisée s'appelle *primitive externe*. La primitive interne est protégée à l'aide d'une fonction de chiffrement et de protection de l'intégrité prenant en entrée une clé symétrique *sessionESPrimKey*. La clé *sessionESPrimKey* est calculée à partir d'une clé *pairwiseESPrimKey* établie entre l'expéditeur et le destinataire ainsi que d'un objet *receiverESPrimRandObject* et *originatorESPrimRandObject*.

La séquence d'événements du cadre ESPrim comprend les trois phases suivantes:

– L'établissement d'une clé pairwiseESPrimKey.

– L'établissement d'une clé sessionESPrimKey à l'expéditeur.

– La sécurisation d'un échange de primitives.

La Figure 8.4.2-1 illustre le flux de messages du cadre ESPrim lors de l'établissement d'une clé *pairwiseESPrimKey* et *sessionESPrimKey* par l'expéditeur. Quant à la Figure 8.4.2-2, elle montre le flux de messages du cadre ESPrim lors de la sécurisation d'un échange de primitives.

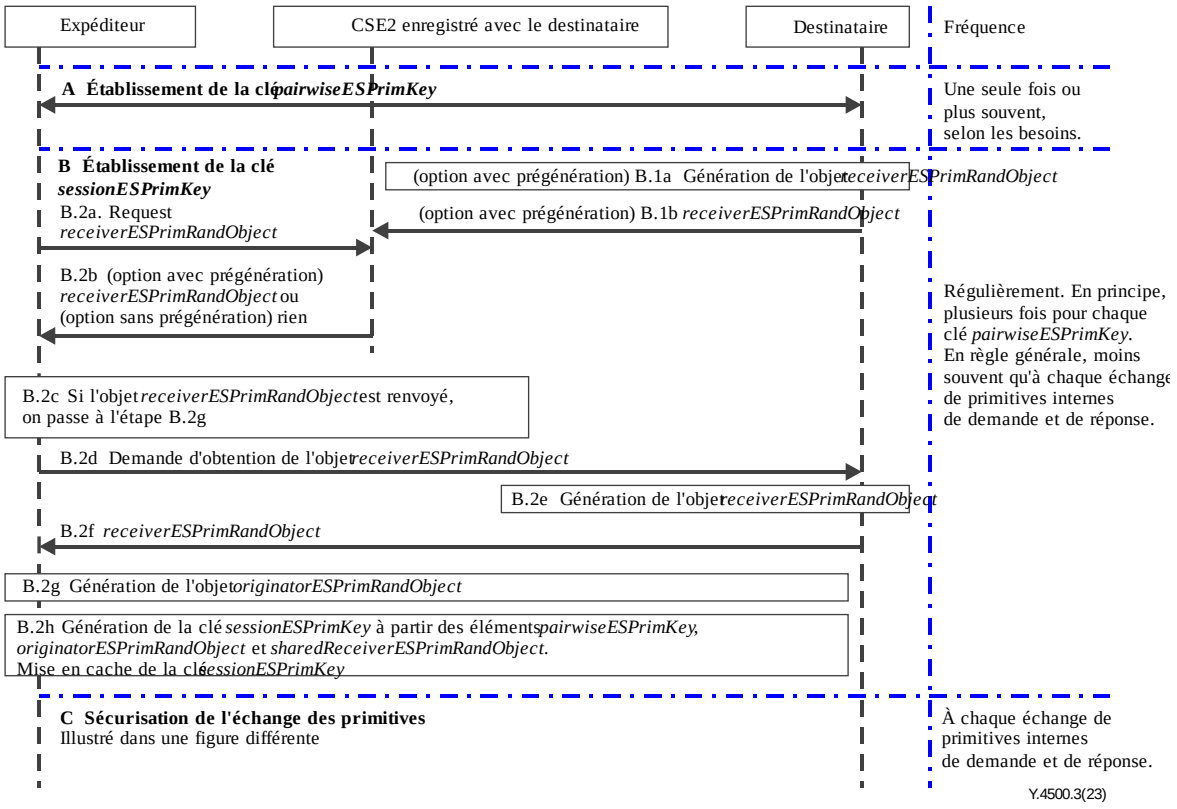


Figure 8.4.2-1 – Flux de messages du cadre de sécurité de bout en bout des primitives (ESPrim) lors de l'établissement d'une clé pairwiseESPrimKey et sessionESPrimKey par l'expéditeur

A Établissement de la clé *pairwiseESPrimKey*: il est possible d'établir la clé *pairwiseESPrimKey* grâce à l'un des cadres suivants:

Cadre de configuration de clé *pairwiseESPrimKey*: la clé *pairwiseESPrimKey* doit être allouée à l'expéditeur et au destinataire. Ce justificatif d'identité doit être configuré de l'une des manières suivantes:

– par préconfiguration;

– à l'aide d'un cadre de configuration à distance de la sécurité (RSPF) défini au § 8.3; ou

– par l'établissement d'une clé à l'aide d'un certificat de sécurité de bout en bout (ESCertKE) entre l'expéditeur et le destinataire conformément au § 8.7 "Établissement d'une clé à l'aide d'un certificat de sécurité de bout en bout (ESCertKE)".

Au cours de ce processus, l'expéditeur et le destinataire établissent également l'identificateur *pairwiseESPrimKeyID* et, éventuellement, l'élément *pairwiseESPrimKeyLifetime*. Si aucun élément *pairwiseESPrimKeyLifetime* n'est configuré, la clé *pairwiseESPrimKey* n'expire jamais. L'expéditeur et le destinataire mettent en cache l'objet (*pairwiseESPrimKeyID, pairwiseESPrimKey, pairwiseESPrimKeyLifetime* [facultatif]) afin de l'utiliser pour le traitement des primitives suivantes.

Cadre MAF ESPrim: l'expéditeur et la fonction d'authentification M2M (MAF) s'authentifient mutuellement à l'aide de clés symétriques (préconfigurées ou configurées à distance) et calculent la clé de connexion sécurisée M2M (Kc) et l'identificateur de clé correspondant (KcID). L'expéditeur génère ensuite la clé *pairwiseESPrimKey* en fonction de la clé Kc ainsi qu'une chaîne de caractères réservée. La valeur de l'identificateur KcID est utilisée durant la phase C en tant qu'identificateur *pairwiseESPrimKeyID* dans l'objet JWE/XML-ENC. L'expéditeur met en cache la paire (*pairwiseESPrimKeyID, pairwiseESPrimKey*) afin de l'utiliser pour le traitement des primitives suivantes. Le destinataire récupère la clé Kc et la durée de vie des justificatifs d'identité à l'aide de la fonction MAF après réception d'une primitive interne de demande sécurisée au moyen de la clé *pairwiseESPrimKey* correspondante (voir étape C.6.a.).

Si la clé *pairwiseESPrimKey* est établie à l'aide de la fonction MAF, elle est généralement dotée d'une durée de vie plus courte qu'avec l'option précédemment exposée.

Le destinataire indique qu'il peut prendre en charge le cadre ESPrim: si le destinataire prend en charge le cadre ESPrim, il doit garantir les éléments suivants pour sa ressource <*remoteCSE*> sur toutes les entités de services communs enregistrées auprès de lui:

– La ressource <*remoteCSE*> du destinataire doit inclure l'attribut *e2eSecInfo*.

– L'attribut *e2eSecInfo* de cette ressource doit indiquer la prise en charge du cadre ESPrim.

B Établissement d'une clé sessionESPrimKey à l'expéditeur: le destinataire doit (a) générer au préalable un objet *receiverESPrimObject*, qui est distribué afin d'être utilisé par plusieurs expéditeurs en vue d'établir une clé *sessionESPrimKey*, ou bien (b) générer un objet *receiverESPrimRandObject* unique sur demande (auquel cas, aucune action n'est requise avant la réception d'une telle demande). Si le destinataire choisit de générer un objet *receiverESPrimRandObject* unique sur demande, il doit, dans le deuxième cas, garantir que le paramètre *sharedReceiverESPrimRandObject* n'est présent dans l'attribut *e2eSecInfo* de sa ressource <*remoteCSE*> sur aucune entité de services communs enregistrée auprès de lui. L'absence du paramètre *sharedReceiverESPrimRandObject* indique que le destinataire fournira un objet *receiverESPrimRandObject* unique sur demande.

B.1 **Prégénération de l'objet *sharedReceiverESPrimRandObject* par le destinataire (le cas échéant)**: si le destinataire a opté pour la prégénération et la distribution de l'objet *receiverESPrimRandObject*, il effectue les étapes suivantes chaque fois qu'il s'apprête à configurer un nouvel objet partagé *receiverESPrimRandObject*.

B.1a.1 Le destinataire doit générer un objet *receiverESPrimRandObject* comprenant les paramètres suivants:

Le destinataire doit générer une nouvelle valeur aléatoire *ESPrimRandValue* de 128 bits.

Le destinataire doit affecter une valeur à l'élément *ESPrimRandExpiry*, qui indique quand l'objet *receiverESPrimRandObject* cessera d'être valide.

Le destinataire doit attribuer un identificateur *ESPrimRandID* à l'objet *receiverESPrimRandObject* en respectant les critères suivants: (a) l'identificateur *ESPrimRandID* doit indiquer que l'objet *receiverESPrimRandObject* est partagé; (b) l'identificateur *ESPrimRandID* doit être unique parmi les objets partagés *receiverESPrimRandObject* émis par le destinataire et valides au moment de l'émission. Ces critères garantissent que l'objet *receiverESPrimRandObject* est unique jusqu'à son expiration.

Le destinataire doit inclure une liste de valeurs *sessionESPrimKeyGenerationAlgorithmID* qui répertorie les algorithmes dont il peut se servir en vue de générer une clé *sessionESPrimKey* à l'aide de cet objet *receiverESPrimRandObject*.

Le destinataire doit inclure une liste de valeurs *AEADAlgorithmID* qui répertorie les algorithmes qu'il peut utiliser sur cet objet receiverESPrimRandObject.

B.1b Le destinataire doit mettre à jour le paramètre *sharedReceiverESPrimRandObject* de l'attribut *e2eSecInfo* de sa ressource <*remoteCSE*> sur toutes les entités de services communs enregistrées auprès de lui.

NOTE 1 – Plusieurs expéditeurs finiront par utiliser des valeurs identiques pour l'objet *sharedReceiverESPrimRandObject* actif.

B.2 **Obtention de l'objet *receiverESPrimRandObject* par l'expéditeur**: l'expéditeur doit effectuer les étapes suivantes quand elle établit une clé *sessionESPrimKey* avec le destinataire:

B.2a L'expéditeur doit récupérer l'attribut *e2eSecInfo* de la ressource <*remoteCSE*> du destinataire sur une entité de services communs, notée ici sous le nom de "CSE2", qui est enregistrée auprès du destinataire.

B.2b Si l'attribut *e2eSecInfo* est présent dans la ressource <*remoteCSE*> du destinataire sur l'entité CSE2, cette dernière renvoie l'attribut *e2eSecInfo.* Autrement, CSE2 renvoie un message d'erreur à l'expéditeur.

B.2c L'expéditeur détermine si le destinataire prend en charge le cadre ESPrim, l'attribut *e2eSecInfo* devant être présent et confirmer la prise en charge de cette dernière.

B.2.c.1 Si le destinataire ne prend pas en charge le cadre ESPrim, l'expéditeur annule la procédure.

B.2.c.2 Si le destinataire prend en charge le cadre ESPrim et que l'attribut *e2eSecInfo* comprend un paramètre *sharedReceiverESPrimRandObject*, l'expéditeur examine l'élément *ESPrimRandExpiry* de ce dernier afin de déterminer si l'objet *sharedReceiverESPrimRandObject* a expiré. Si l'objet *sharedReceiverESPrimRandObject* n'a pas expiré, l'expéditeur attribue la valeur de celui-ci à l'objet *receiverESPrimRandObject*, puis passe à l'étape B.2g. Si l'objet *sharedReceiverESPrimRandObject* a expiré, l'expéditeur passe à l'étape B.2d.

B.2c.3 Si le destinataire prend en charge le cadre ESPrim et que l'attribut *e2eSecInfo* ne comprend aucun paramètre *sharedReceiverESPrimRandObject*, l'expéditeur passe à l'étape B.2d.

B.2d L'expéditeur doit envoyer un message au destinataire demandant un objet *receiverESPrimRandObject*.

B.2e À la réception de la demande, le destinataire doit générer un objet *receiverESPrimRandObject* comprenant les paramètres suivants:

– Le destinataire doit générer une nouvelle valeur aléatoire *ESPrimRandValue* de 128 bits.

– Le destinataire doit affecter une valeur à l'élément *ESPrimRandExpiry*, qui indique quand l'objet *receiverESPrimRandObject* cessera d'être valide.

– Le destinataire doit attribuer un identificateur *ESPrimRandID* à l'objet *receiverESPrimRandObject* en respectant les critères suivants: (a) l'identificateur *ESPrimRandID* doit indiquer que l'objet *receiverESPrimRandObject* n'est pas partagé; (b) l'identificateur *ESPrimRandID* doit être unique parmi les objets non partagés *receiverESPrimRandObject* émis par le destinataire et valides au moment de l'émission. Ces critères garantissent que l'objet *receiverESPrimRandObject* est unique jusqu'à son expiration.

– Le destinataire doit inclure une liste de valeurs *sessionESPrimKeyGenerationAlgorithmID* qui répertorie les algorithmes dont il peut se servir en vue de générer une clé *sessionESPrimKey* à l'aide de cet objet *receiverESPrimRandObject*.

– Le destinataire doit inclure une liste de valeurs *AEADAlgorithmID* qui répertorie les algorithmes qu'il peut utiliser sur cet objet receiverESPrimRandObject.

B.2f Le destinataire envoie à l'expéditeur un message contenant l'objet *receiverESPrimRandObject.*

B.2g L'expéditeur génère un objet *originatorESPrimRandObject* comprenant les paramètres suivants:

– L'expéditeur génère une nouvelle valeur aléatoire *ESPrimRandValue* de 128 bits.

– L'expéditeur affecte une valeur à l'élément *ESPrimRandExpiry*, qui indique quand l'objet *originatorESPrimRandObject* cessera d'être valide. La date d'expiration *ESPrimRandExpiry* ne doit pas être ultérieure à celle de l'élément *ESPrimRandExpiry* contenu dans l'objet *receiverESPrimRandObject* récupéré à l'étape B.2c ou B.2f.

– L'expéditeur doit attribuer un identificateur *ESPrimRandID* à l'objet *originatorESPrimRandObject* en respectant les critères suivants: (a) l'identificateur *ESPrimRandID* doit indiquer que l'objet *originatorESPrimRandObject* n'est pas partagé; (b) l'identificateur *ESPrimRandID* doit être unique parmi les objets non partagés *originatorESPrimRandObject* émis par l'expéditeur et valides au moment de l'émission. Ces critères garantissent que l'objet *originatorESPrimRandObject* est unique jusqu'à son expiration.

– L'expéditeur doit attribuer un seul identificateur *sessionESPrimKeyGenerationAlgorithmID* à l'algorithme que l'expéditeur utilise pour générer une clé *sessionESPrimKey* à l'aide de l'objet *originatorESPrimRandObject*. L'algorithme en question doit être l'un de ceux qui figurent parmi les valeurs *sessionESPrimKeyGenerationAlgorithmID* de l'objet *receiverESPrimRandObject* obtenu durant l'étape B.2c ou B.2f.

– L'expéditeur doit inclure une liste de valeurs *AEADAlgorithmID* qui répertorie les algorithmes qu'il peut utiliser sur cet objet *originatorESPrimRandObject*. Les algorithmes en question doivent être ceux qui portent l'identificateur *AEADAlgorithmID* de l'objet *receiverESPrimRandObject* obtenu durant l'étape B.2c ou B.2f.

B.2h L'expéditeur doit générer la clé *sessionESPrimKey* en fonction de la clé *pairwiseESPrimKey*, de l'objet *receiverESPrimRandObject* reçu à l'étape B.2c et du tuple *originatorESPrimRandTuple* généré à l'étape B.2.g.

NOTE 2 – La clé *sessionESPrimKey* servant à sécuriser la primitive interne de demande est toujours utilisée pour protéger la primitive interne de réponse correspondante. Ainsi, la clé *sessionESPrimKey* doit être mise en cache, au minimum, le temps que la primitive interne de réponse ait été vraisemblablement reçue. En règle générale, l'expéditeur met la clé *sessionESPrimKey* en cache plus longtemps, car celle-ci peut permettre de sécuriser plusieurs échanges de primitives.

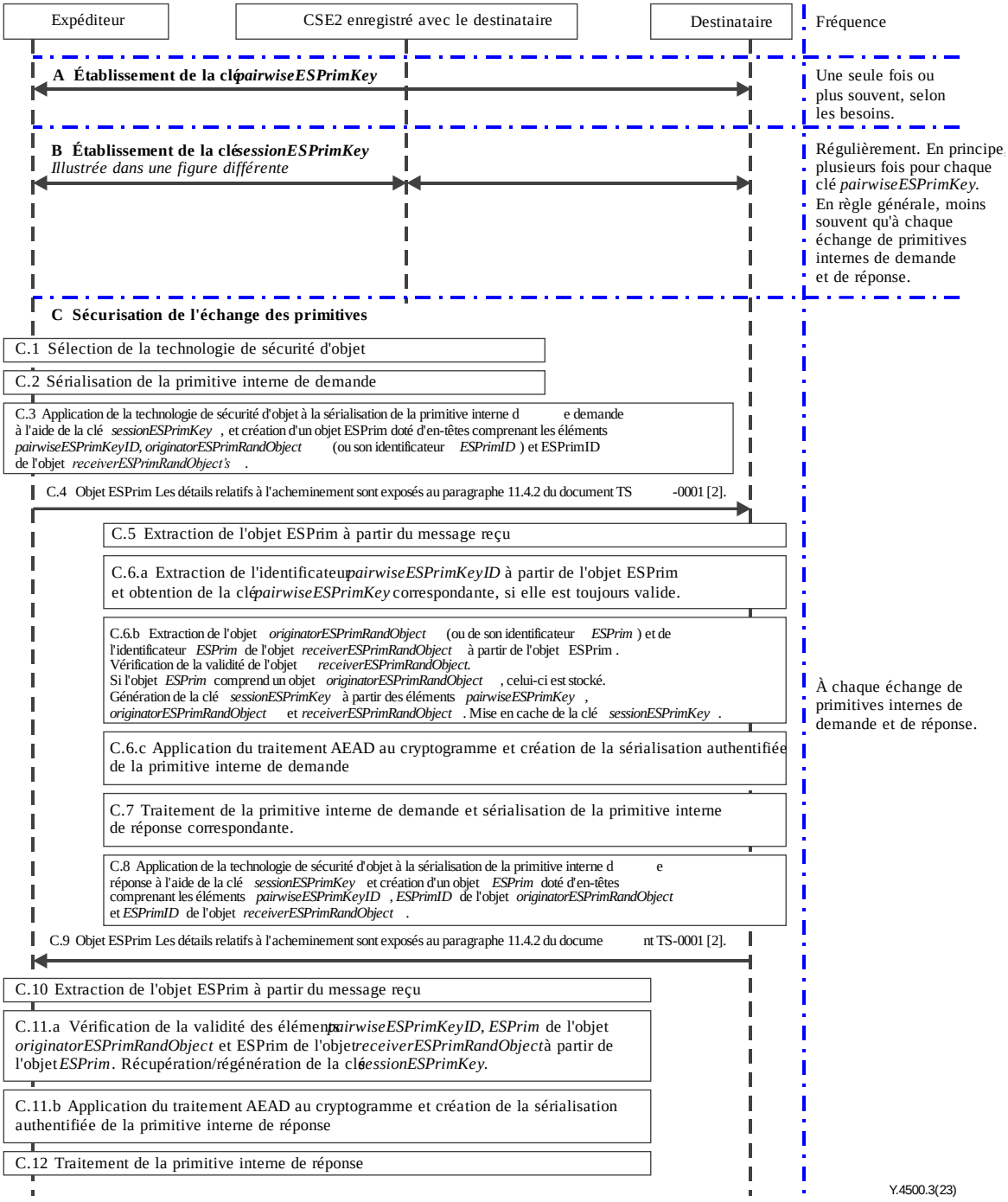


Figure 8.4.2-2 – Flux de messages du cadre de sécurité de bout en bout des primitives (ESPrim) lors de la sécurisation d'un échange de primitives

C Sécurisation d'un échange de primitives

NOTE 3 – L'expéditeur sélectionne le type de sérialisation (par exemple au format JSON ou XML) de la primitive interne de demande à sécuriser.

C.1 L'expéditeur sélectionne la technologie de sécurité d'objet qu'il peut prendre en charge ainsi que le type de sérialisation de la primitive interne de demande.

C.2 L'expéditeur sérialise la primitive interne de demande.

C.3 L'expéditeur crée un objet ESPrim à l'aide de la technologie de sécurité d'objet comme suit:

Les en-têtes d'un objet ESPrim doivent inclure les informations suivantes:

– Une clé *pairwiseESPrimKey*.

– Un objet *originatorESPrimRandObject* utilisé pour générer la clé *sessionESPrimKey* ou l'identificateur *ESPrimRandID* correspondant. Si cet objet ESPrim est le premier à être reçu par le destinataire et sécurisé par l'expéditeur grâce à un objet *originatorESPrimRandObject* spécifique, ce dernier doit être entièrement inclus. Sinon, il faut inclure l'un des objets *originatorESPrimRandObject* ou l'un des identificateurs *ESPrimRandID*.

– L'identificateur *ESPrimRandID* de l'objet *receiverESPrimRandObject* utilisé pour générer la clé *sessionESPrimKey*.

– L'identificateur *AEADAlgorithmID* de l'objet ESPrim. Celui-ci doit correspondre à l'un des algorithmes AEAD indiqués dans l'objet *originatorESPrimRandObject.*

Le texte en clair (à chiffrer) doit être la sérialisation de la primitive interne de demande.

La clé *sessionESPrimKey* est utilisée directement en tant que clé symétrique, qui permet le chiffrement authentifié du texte en clair et la création du cryptogramme dans l'objet ESPrim. On supposera que le cryptogramme contient le code MIC destiné à vérifier l'intégrité de la primitive interne de demande.

C.4 L'expéditeur crée une primitive externe de demande pour l'acheminement de l'objet ESPrim dans les conditions exposées dans la Recommandation [ITU-T Y.4500.1]. L'expéditeur envoie la primitive externe de demande au destinataire.

C.5 Le destinataire traite la primitive externe de demande reçue afin d'extraire l'objet ESPrim conformément au document [ITU-T Y.4500.1].

C.6 Le destinataire traite l'objet ESPrim.

C.6a Le destinataire extrait l'identificateur *pairwiseESPrimKeyID* des en-têtes de l'objet ESPrim et récupère la clé *pairwiseESPrimKey* correspondante:

Si l'identificateur *pairwiseESPrimKeyID* correspond à une clé *pairwiseESPrimKey* configurée, le destinataire utilise ladite clé (configurée au préalable).

Si l'identificateur *pairwiseESPrimKeyID* correspond à une clé *pairwiseESPrimKey* MAF: si le destinataire reçoit un message contenant l'identificateur *pairwiseESPrimKeyID* pour la première fois, la procédure qui s'applique est la suivante.

C.6a.1 Le destinataire identifie la fonction MAF de l'identificateur *pairwiseESPrimKeyID* (c'est-à-dire un identificateur KcID).

C.6a.2 Le destinataire établit une connexion TLS sécurisée à la fonction MAF, puis demande une clé de connexion sécurisée M2M (Kc) et la durée de vie de la clé Kc correspondant à l'identificateur *pairwiseESPrimKeyID* (qui est identique à l'identificateur KcID).

C.6a.3 La fonction MAF fournit une clé Kc et indique sa durée de vie au destinataire.

C.6a.4 Le destinataire génère ensuite la clé *pairwiseESPrimKey* en fonction de la clé Kc ainsi qu'une chaîne de caractères réservée.

C.6a.5 Le destinataire attribue la durée de vie de la clé Kc à l'élément *pairwiseESPrimKeyLifetime*.

C.6a.6 Le destinataire met en cache l'ensemble (*pairwiseESPrimKeyID, pairwiseESPrimKey, pairwiseESPrimKeyLifetime)* afin de l'utiliser pour le traitement des primitives suivantes.

Si le destinataire a au préalable mis en cache l'ensemble (*pairwiseESPrimKeyID*, pairwiseESPrimKey, *pairwiseESPrimKeyLifetime*) et que l'élément *pairwiseESPrimKeyLifetime* n'a pas expiré, il peut utiliser la clé *pairwiseESPrimKey* mise en cache.

C.6b Le destinataire doit effectuer la procédure suivante afin de générer la clé *sessionESPrimKey*:

C.6b.1 Le destinataire extrait l'identificateur *ESPrimRandID* de l'objet *receiverESPrimRandObject* depuis les en-têtes de l'objet ESPrim et essaie de récupérer la valeur correspondante mise en cache de l'objet *receiverESPrimRandObject*. Si aucune valeur mise en cache n'est trouvée ou que la valeur mise en cache a expiré, le destinataire répond à la primitive externe de demande avec une erreur.

C.6b.2 Le destinataire extrait l'encodage de l'objet *originatorESPrimRandObject ou* l'identificateur *ESPrimRandID* de celui-ci depuis les en-têtes de l'objet ESPrim et utilise la méthode de décodage qui convient. Si un objet *originatorESPrimRandObject* est fourni, il doit être mis en cache. Si un identificateur *ESPrimRandID* est fourni, le destinataire récupère la valeur mise en cache correspondante de l'objet *originatorESPrimRandObject*. Si aucune valeur mise en cache n'est trouvée ou que la valeur mise en cache est réputée expirée, le destinataire répond à la primitive externe de demande avec un message d'erreur.

Le destinataire traite l'objet *originatorESPrimRandObject*:

C.6b.2.i Le destinataire vérifie la date *ESPrimExpiry* de l'objet *originatorESPrimRandObject* afin de s'assurer qu'elle (a) n'est pas passée; et qu'elle (b) n'est pas ultérieure à la date *ESPrimExpiry* de l'objet *receiverESPrimRandObject*.

C.6b.2.ii Le destinataire extrait l'identificateur *sessionESPrimKeyGenerationAlgorithmID* et vérifie que l'algorithme identifié correspond à l'un des identificateurs *sessionESPrimKeyGenerationAlgorithmID* contenus dans l'objet *receiverESPrimRandObject*.

C.6b.2.iii Le destinataire génère la clé *sessionESPrimKey* en se basant sur la clé *pairwiseESPrimKey*, ainsi que sur les objets *receiverESPrimRandObject* et *originatorESPrimRandObject*, ou bien récupère la valeur de la clé *sessionESPrimKey* si elle a été générée et mise en cache au préalable.

NOTE 4 – La *sessionESPrimKey* utilisée pour sécuriser une primitive de demande interne est toujours utilisée pour protéger la primitive de réponse interne correspondante, de sorte que la *sessionESPrimKey* doit être mise en cache au moins jusqu'à ce que la primitive de réponse interne correspondante soit envoyée. Le récepteur met généralement en cache la *sessionESPrimKey* pendant une période plus longue, car l'expéditeur peut utiliser la *sessionESPrimKey* pour sécuriser plusieurs échanges de primitives.

C.6c Étapes de déchiffrement authentifié suivies par le destinataire:

C.6c.1 Le destinataire extrait les identificateurs *AEADAlgorithmID* de l'objet *originatorESPrimRandObject* et vérifie que les algorithmes identifiés forment un sous-ensemble de l'identificateur *AEADAlgorithmID* de l'objet *receiverESPrimRandObject*.

Le destinataire traite l'identificateur *AEADAlgorithmID* des en-têtes de l'objet ESPrim et vérifie que l'algorithme identifié correspond à l'un des identificateurs *AEADAlgorithmID* contenus dans l'objet *originatorESPrimRandObject*.

C.6c.2 Le destinataire applique l'identificateur de l'algorithme AEAD contenu dans l'en-tête de l'objet ESPrim au paramètre du cryptogramme dudit objet afin d'obtenir le texte en clair vérifié à l'aide de la clé *sessionESPrimKey*. On supposera que le cryptogramme contient le code MIC destiné à vérifier l'intégrité de la primitive interne de demande. La sérialisation authentifiée de la primitive interne de demande est le texte en clair vérifié produit par l'algorithme AEAD.

C.7 Le destinataire traite la primitive interne de demande et procède ainsi à la sérialisation de la primitive interne de réponse correspondante.

NOTE 5 – Les Étapes C.8 à C.12 reproduisent presque à l'identique les Étapes C.2 à C.7, à ceci près que l'expéditeur et le destinataire intervertissent leurs rôles durant l'échange et que les primitives de demande sont remplacées par des primitives de réponse. Quelques différences mineures les distinguent. Par exemple, certaines procédures de traitement des demandes réalisées pendant les Étapes C.2 à C.7 ne sont pas requises

durant le traitement des réponses, car l'expéditeur génère déjà une clé *sessionESPrimKey*. Cela n'est nécessaire que pour identifier la clé *sessionESPrimKey* adéquate, comme à l'étape C.11.a.

C.8 L'expéditeur utilise la même clé *sessionESPrimKey* que dans l'objet ESPrim reçu à l'étape C.5. Par conséquent, les éléments *pairwiseESPrimKeyID, originatorESPrimRandObject* et *receiverESPrimRandObject* sont identiques à ceux qui sont reçus à l'étape C.5.

Le destinataire crée un objet ESPrim à l'aide de la technologie de sécurité d'objet comme suit:

Les en-têtes d'un objet ESPrim doivent inclure les informations suivantes:

– Une clé *pairwiseESPrimKey*.

– l'identificateur *ESPrimRandID* de l'objet *originatorESPrimRandObject*.

– l'identificateur *ESPrimRandID* de l'objet *receiverESPrimRandObject*.

– L'identificateur *AEADAlgorithmID* de l'objet ESPrim. Celui-ci doit correspondre à l'un des algorithmes AEAD indiqués dans l'objet *originatorESPrimRandObject.*

Le texte en clair (à chiffrer) doit être la sérialisation de la primitive interne de réponse.

La clé *sessionESPrimKey* est utilisée directement en tant que clé symétrique, qui permet le chiffrement authentifié du texte en clair et la création du cryptogramme dans l'objet ESPrim. On supposera que le cryptogramme contient le code MIC destiné à vérifier l'intégrité de la primitive interne de demande.

C.9 Le destinataire crée une primitive externe de demande pour l'acheminement de l'objet ESPrim dans les conditions exposées dans la Recommandation [ITU-T Y.4500.1]. L'expéditeur envoie la primitive externe de réponse au destinataire.

C.10 L'expéditeur traite la primitive externe de réponse reçue afin d'extraire l'objet ESPrim conformément à la Recommandation [ITU-T Y.4500.1].

C.11 L'expéditeur traite l'objet ESPrim.

C.11a L'expéditeur extrait, depuis les en-têtes de l'objet ESPrim, la valeur des éléments *pairwiseESPrimKeyID, ESPrimRandID* de l'objet *originatorESPrimRandObject* et *ESPrimRandID* de l'objet *receiverESPrimRandObject*. Ces valeurs doivent correspondre à celle des éléments *pairwiseESPrimKeyID, ESPrimRandID* de l'objet *originatorESPrimRandObject* et *ESPrimRandID* de l'objet *receiverESPrimRandObject* d'une session que l'expéditeur considère comme valide.

Si l'une de ces valeurs a expiré, la primitive externe de réponse est ignorée.

NOTE 6 – C'est pourquoi le délai d'expiration de ces valeurs doit être suffisamment long pour permettre la réception de la primitive interne de réponse correspondante.

Autrement, l'expéditeur doit mettre en cache la valeur de la clé *sessionESPrimKey* correspondant à ces valeurs ou générer une nouvelle clé *sessionESPrimKey.*

C.11b L'expéditeur applique l'algorithme AEAD identifié dans l'en-tête de l'objet ESPrim au paramètre du cryptogramme dudit objet afin d'obtenir le texte en clair vérifié à l'aide de la clé *sessionESPrimKey.* On supposera que le cryptogramme contient le code MIC destiné à vérifier l'intégrité de la primitive interne de demande. La sérialisation authentifiée de la primitive interne de demande est le texte en clair vérifié produit par l'algorithme AEAD.

C.12 L'expéditeur traite la primitive interne de réponse.

8.4.3 Détails du protocole de sécurité de bout en bout des primitives (ESPrim)

8.4.3.1 Définition des paramètres du protocole de sécurité de bout en bout des primitives (ESPrim)

8.4.3.1.1 Définition du paramètre originatorESPrimRandObject

La structure du paramètre originatorESPrimRandObject est indiquée dans le Tableau 8.4.3.1.1-1. Ce paramètre sert à établir une clé *sessionESPrimKey* dans le cadre du protocole de sécurité de bout en bout des primitives (ESPrim) décrit au § 8.4.2. Le type de données du paramètre originatorESPrimRandObject est détaillé dans la Recommandation [ITU-T Y.4500.4].

Tableau 8.4.3.1.1-1 – Structure du paramètre originatorESPrimRandObject

Chemin de l'élément	Multiplicité	Description
esprimRandID	1	Identificateur du paramètre originatorESPrimRandObject attribué par l'entité de services communs ou l'entité d'application qui génère ce dernier.
esprimRandValue	1	Valeur de 128 bits générée aléatoirement.
esprimRandExpiry	1	Date et heure d'expiration du paramètre originatorESPrimRandObject.
esprimKeyGenAlgID	1	Identificateur énuméré de l'algorithme sélectionné pour la génération de la clé sessionESPrimKey par l'entité de services communs ou l'entité d'application qui génère le paramètre originatorESPrimRandObject.
esprimProtocolAndAlgIDs	1	Liste d'identificateurs énumérés des algorithmes AEAD pris en charge par l'entité de services communs ou l'entité d'application qui génère ce dernier.

8.4.3.1.2 Définition du paramètre receiverESPrimRandObject

La structure du paramètre receiverESPrimRandObject est indiquée dans le Tableau 8.4.3.1.2-1. Ce paramètre sert à établir une clé sessionESPrimKey dans le cadre du protocole de sécurité de bout en bout des primitives (ESPrim) décrit au § 8.4.2. Le type de données du paramètre receiverESPrimRandObject est détaillé dans la Recommandation [ITU-T Y.4500.4].

Tableau 8.4.3.1.2-1 – Structure du paramètre *receiverESPrimRandObject*

Chemin de l'élément	Multiplicité	Description
esprimRandID	1	Identificateur du paramètre receiverESPrimRandObject attribué par l'entité de services communs ou l'entité d'application qui génère ce dernier.
esprimRandValue	1	Valeur de 128 bits générée aléatoirement.
esprimRandExpiry	1	Date et heure d'expiration du paramètre receiverESPrimRandObject.
esprimKeyGenAlgIDs	1	Liste d'identificateurs énumérés des algorithmes de génération de la clé sessionESPrimKey pris en charge par l'entité de services communs ou l'entité d'application qui génère le paramètre receiverESPrimRandObject.
esprimProtocolAndAlgIDs	1	Liste d'identificateurs énumérés des algorithmes AEAD pris en charge par l'entité de services communs ou l'entité d'application qui génère le paramètre receiverESPrimRandObject.

8.4.3.1.3 Définition de l'attribut de ressource *e2eSecInfo*

L'attribut *e2eSecInfo* fait partie des types de ressources <*CSEBase*>, <*remoteCSE*> et <*AE*>. La structure de l'attribut de ressource *e2eSecInfo* est indiquée dans le Tableau 8.4.3.1.3-1. Ce paramètre sert à établir une clé sessionESPrimKey dans le cadre du protocole de sécurité de bout en bout des primitives (ESPrim) décrit au § 8.4.2. Les types de données sont détaillés dans la Recommandation [ITU-T Y.4500.4].

Tableau 8.4.3.1.3-1 – Structure de l'attribut *e2eSecInfo*

Chemin de l'élément	Multiplicité	Description
supportedE2ESecurityFeatures	1	Liste des identificateurs d'utilisation de sécurité (SUID) des fonctionnalités de sécurité de bout en bout prises en charge par l'entité de services communs ou l'entité d'application associée à la ressource <*CSEBase*>, <*remoteCSE*> ou <*AE*> qui contient l'attribut de ressource *e2eSecInfo*.
e2ECertificates	0 ou 1	Liste de certificats associés à l'entité de services communs ou à l'entité d'application liée à la ressource <*CSEBase*>, <*remoteCSE*> ou <*AE*> qui contient l'attribut de ressource *e2eSecInfo*.
sharedReceiverESPrimRandObject	0 ou 1	Paramètre receiverESPrimRandObject (voir le § 8.4.3.1.2) généré par l'entité de services communs ou l'entité d'application associée à la ressource <*CSEBase*>, <*remoteCSE*> ou <*AE*> qui contient l'attribut de ressource *e2eSecInfo*.

8.4.3.2 Formatage et traitement de l'objet ESPrim à l'aide de la fonction de sérialisation compacte JWE

Contexte: la norme JSON Web Encryption (JWE) définie dans [IETF RFC 7516] fournit un format simple permettant de chiffrer n'importe quel objet de données. Deux sérialisations JWE sont fournies: une sérialisation compacte protégée par URI et une sérialisation JSON.

Il est possible d'encoder le formatage et l'analyse d'une sérialisation compacte JWE de sécurité de bout en bout des primitives (ESPrim) sans les outils génériques utilisés pour le formatage ou l'analyse de fichier JSON.

Une sérialisation compacte JWE est représentée par la concaténation de cinq paramètres JWE:

– BASE64URL(UTF8(JWE Protected Header)) || '.' ||
– BASE64URL(JWE Encrypted Key) || '.' ||
– BASE64URL(JWE Initialization Vector) || '.' ||
– BASE64URL(JWE Ciphertext) || '.' ||
– BASE64URL(JWE Authentication Tag)

où BASE64URL(OCTETS) indique l'encodage base64url du paramètre OCTETS conformément à la section 2 de la spécification JSON Web Signature [IETF RFC 7515].

NOTE 1 – Si le paramètre OCTETS est une séquence d'octets vide, BASE64URL(OCTETS) sera une chaîne de caractères vide selon [IETF RFC 7515].

NOTE 2 – La sérialisation compacte JWE n'est pas aussi souple que la sérialisation JWE JSON. Toutefois, elle offre suffisamment de souplesse pour les objets ESPrim. De plus, les sérialisations compactes JWE étant faciles à formater et à analyser, il est simple de les créer à partir de représentations XML et JSON de primitives.

Définition des paramètres JWE pour le cadre ESPrim: le tableau 8.4.3.2-1 indique la valeur des cinq paramètres JWE utilisés dans le cadre d'une sérialisation compacte JWE.

Tableau 8.4.3.2-1 – Composants JWE utilisés dans les objets ESPrim

Valeur JWE	Type d'élément	Vide	Contenu du composant
En-tête protégé JWE	JSON	Jamais	Voir le Tableau 8.4.3.2-2 (Paramètres de l'en-tête protégé JWE)
Clé de chiffrement JWE	Valeur binaire	Toujours	Cette valeur est vide pour le mode de gestion des clés utilisé pour le cadre ESPrim
Vecteur d'initialisation JWE	Valeur binaire	Sous certaines conditions	Voir [IETF RFC 7516]
Cryptogramme JWE	Valeur binaire	Jamais	
Étiquette d'authentification	Valeur binaire	Sous certaines conditions	
NOTE – L'algorithme sélectionné pour le chiffrement détermine si ces composants sont vides ou non.			

Le Tableau 8.4.3.2-2 (Paramètres de l'en-tête protégé JWE) décrit les paramètres contenus dans l'en-tête protégé JWE lors de l'utilisation de JWE pour le cadre ESPrim.

Tableau 8.4.3.2-2 – Paramètres de l'en-tête protégé JWE

Chemin de l'élément	Multiplicité pour le cadre ESPrim	Finalité	Spécification de l'élément	Description de la valeur attribuée
"alg"	1	Mode de gestion des clés	[IETF RFC 7516]	La valeur "dir" indique la sélection du chiffrement direct.
"enc"	1	Algorithme de chiffrement		Les options disponibles sont identiques à celles qui sont utilisées durant le chiffrement JWE effectué dans le cadre ESData. Voir le § 8.7.3 (Détails concernant le cadre ESData)
"kid"	1	Identificateur de clé	[IETF RFC 7516]	Identificateur de la clé pairwiseESPrimKey
"cty"	0 ou 1	Type de support du contenu sécurisé	[IETF RFC 7516]	Déterminé par la sérialisation de la primitive (XML ou JSON) sélectionnée par l'expéditeur.
"ori"	1	Correspondance de la saisie de l'expéditeur à la génération de clé de session	Paragraphe 8.4.3.1.1	Identificateur esprimRandID de l'objet originatorESPrimRandObject utilisé pour générer la clé sessionESPrimKey de ce l'objet ESPrim.
"rri"	1	Correspondance de la saisie du destinataire à la génération de clé de session	Paragraphe 8.4.3.1.2	Identificateur esprimRandID de l'objet receiverESPrimRandObject utilisé pour générer la clé sessionESPrimKey de ce l'objet ESPrim.
"oro"	0 ou 1	originatorESPrimRandObject	Paragraphe 8.4.3.1.1 (avec utilisation de la sérialisation JSON)	Représentation JSON d'un objet originatorESPrimRandObject généré par l'expéditeur. Envoyée uniquement par l'expéditeur.
"rro"	0 ou 1	receiverESPrimRandObject	Paragraphe 8.4.3.1.2 (avec utilisation de la sérialisation JSON)	Représentation JSON d'un objet receiverESPrimRandObject généré par le destinataire. Envoyée uniquement par le destinataire.

Les paramètres de l'en-tête protégé JWE et leur recours quand JWE est utilisé pour le cadre ESPrim sont décrits plus en profondeur ci-dessous.

Pour rappel, un objet ESPrim est formé en chiffrant la primitive interne à l'aide de la clé symétrique sessionESPrimKey. La génération de la clé sessionESPrimKey à partir de la clé pairwiseESPrimKey et des objets originatorESPrimRandObject et receiverESPrimRandObject est à distinguer de la gestion des clés permise par JWE. Dans le cadre de JWE, la clé sessionESPrimKey est simplement la valeur d'une clé symétrique secrète partagée entre l'expéditeur et le destinataire. JWE utilise le mode de gestion des clés par chiffrement direct dans ce cas de figure. La sérialisation JWE est nécessaire à la transmission de l'identificateur pairwiseESPrimKey, des identificateurs esprimRandID des objets originatorESPrimRandObject et receiverESPrimRandObject et, éventuellement, d'un objet receiverESPrimRandObject ou originatorESPrimRandObject afin que l'expéditeur et le destinataire puissent générer la clé sessionESPrimKey correcte.

La valeur du paramètre de l'en-tête JWE "alg" (algorithme) [IETF RFC 7516] doit être égale à "dir" pour indiquer le recours au mode de gestion des clés par chiffrement direct.

Le paramètre de l'en-tête JWE "enc" (algorithme de chiffrement) [IETF RFC 7516] peut être sélectionné par l'expéditeur de l'objet ESPrim. L'algorithme de chiffrement doit correspondre aux identificateurs esprimProtocolAndAlgID des objets originatorESPrimRandObject et receiverESPrimRandObject identifiés.

La valeur du paramètre de l'en-tête JWE "kid" (identificateur de clé) [IETF RFC 7516] doit être égale à l'identificateur de la clé pairwiseESPrimKey.

Le paramètre de l'en-tête JWE "cty" doit identifier le type de support de la représentation (JSON ou XML) de la primitive interne.

Les identificateurs esprimRandID des objets originatorESPrimRandObject et receiverESPrimRandObject doivent être attribués aux paramètres "ori" (identificateur aléatoire de l'expéditeur) et "rri" (identificateur aléatoire du destinataire) inclus dans l'en-tête protégé JWE. Ces paramètres sont spécifiques à la norme oneM2M et doivent se conformer à la définition d'un identificateur esprimRandID exposée aux § 8.4.3.1.1 et 8.4.3.1.2.

L'expéditeur peut inclure un objet originatorESPrimRandObject dans le paramètre "oro" (objet aléatoire de l'expéditeur) soit au début d'une session ESPrim ou durant une session ESPrim existante afin d'actualiser les clés de session en bande. Dans le premier cas, le paramètre "ori" doit correspondre à l'identificateur esprimRandID de l'objet originatorESPrimRandObject. Cette condition ne s'applique pas au deuxième cas. Ce paramètre est défini aux paragraphes 8.4.2 et 8.4.3.1.1 et doit être représenté au format JSON.

Le destinataire peut inclure un objet receiverESPrimRandObject dans le paramètre "rro" (objet aléatoire du destinataire) afin d'actualiser les clés de session en bande durant une session existante. Ce paramètre est défini aux § 8.4.2 et 8.4.3.1.2 et doit être représenté au format JSON.

Formation d'un objet ESPrim: la représentation de la primitive interne doit être chiffrée conformément à la procédure de chiffrement des messages décrite dans [IETF RFC 7516]. L'en-tête protégé JWE doit être composé comme indiqué ci-dessus.

La clé de chiffrement de contenu (CEK) doit être la clé sessionESPrimKey générée à l'aide des éléments pairwiseESPrimKey, originatorESPrimRandObject et receiverESPrimRandObject identifiés dans l'en-tête protégé JWE.

Le texte en clair doit être la représentation de la primitive interne.

Le vecteur d'initialisation JWE, le cryptogramme JWE et l'étiquette d'authentification JWE doivent être générés à partir de l'en-tête protégé JWE, du texte en clair et de la clé CEK comme indiqué dans [IETF RFC 7516] et selon l'algorithme de chiffrement identifié.

La sérialisation compacte JWE doit être composée à partir des paramètres JWE.

L'objet ESPrim est la sérialisation compacte JWE.

Traitement d'un objet ESPrim: la sérialisation compacte JWE doit être traitée conformément à la procédure de chiffrement des messages décrite dans [IETF RFC 7516] en veillant à ce que les conditions suivantes soient remplies:

L'en-tête protégé JWE doit être composé comme indiqué ci-dessus.

Si l'objet originatorESPrimRandObject ou receiverESPrimRandObject est inclus, celui-ci doit être enregistré.

La clé de chiffrement de contenu (CEK) doit être la clé sessionESPrimKey générée à l'aide des éléments pairwiseESPrimKey, originatorESPrimRandObject et receiverESPrimRandObject identifiés dans l'en-tête protégé JWE. La génération de la clé sessionESPrimKey est décrite au § 8.4.2.

Le texte en clair doit être généré à partir du vecteur d'initialisation JWE, du cryptogramme JWE, de l'étiquette d'authentification JWE et de la clé CEK comme indiqué dans [IETF RFC 7516] et selon l'algorithme de chiffrement identifié.

8.5 Procédure de sécurité de bout en bout des données (ESData)

8.5.1 Finalité du cadre ESData

La procédure de sécurité de bout en bout des données (ESData) fournit un cadre interopérable de protection des données transportées à l'aide de points de référence oneM2M afin que les entités de services communs par lesquelles les données sont acheminées ne soient pas chargées du traitement de ces dernières. Les données à protéger sont désignées par le terme *charge utile ESData*. De manière générale, la charge utile ESData peut composer entièrement ou en partie la valeur d'un attribut (par exemple, l'attribut *content* d'une ressource <*contentInstance*>) ou le paramètre d'une primitive (par exemple, un jeton d'accès autonome et signé transmis dans une primitive de demande afin d'obtenir une autorisation dynamique).

NOTE – Tout au long du § 8.5, le mot "ESData" qui compose certains termes peut en être retiré afin d'en faciliter la lisibilité. Par exemple, le terme "charge utile ESData" est souvent abrégé, donnant ainsi simplement "charge utile".

Les cas d'utilisation et les exigences correspondants sont exposés dans le document oneM2M TR-0012 [b-oneM2M TR0012].

Dans le cadre ESData, on suppose qu'un seul *point d'extrémité source ESData* effectue le traitement ESData de la charge utile afin d'obtenir une *enveloppe ESData* contenant les données sécurisées et les en-têtes nécessaires. Dans le même temps, un ou plusieurs *points d'extrémité cibles ESData* traitent l'enveloppe pour extraire les données vérifiées. La charge utile est composée de texte en clair (à chiffrer et dont l'intégrité doit être protégée) et de données d'authentification associées (dont seule l'intégrité doit être protégée).

Il n'existe aucune restriction inhérente quant aux entités qui peuvent constituer des points d'extrémité sources ou cibles. Ces derniers peuvent être des entités d'un système oneM2M (c'est-à-dire des entités d'application ou de services communs) ou des entités se trouvant en dehors d'un tel système (par exemple des entités qui font partie d'un système qui fonctionne avec oneM2M).

La présente Recommandation définit les aspects du cadre ESData liés à la gestion des justificatifs d'identité et à la protection des données. En revanche, elle ne traite pas de l'acheminement des enveloppes ESData.

8.5.2 Architecture du cadre ESData

8.5.2.1 Liste des classes de sécurité ESData et des options de protection ESData

Les classes de sécurité ESData suivantes sont fournies:

Chiffrement uniquement: garantit la protection de la confidentialité et de l'intégrité (voir la note). Cette charge utile est protégée à l'aide de clés symétriques établies grâce à un ou plusieurs des éléments suivants:

Des clés symétriques établies à l'aide des points d'extrémité cibles. Dans ce cas, le point d'extrémité source peut être authentifié à moins que la clé symétrique ne soit partagée avec plusieurs points d'extrémité cibles.

Un certificat de point d'extrémité cible: quand un certificat de point d'extrémité cible est utilisé, ledit point d'extrémité ne peut pas authentifier le point d'extrémité source.

NOTE – À proprement parler, cette classe assure le chiffrement et la protection de l'intégrité. Cela étant, ce terme englobe l'usage qui en est fait dans divers protocoles, tels que JSON Web Encryption (JWE) et XML-Encryption, qui peuvent également assurer ce genre de fonctionnalités.

Signature uniquement: assure l'authentification de la source, la protection de l'intégrité et la non-répudiation (quand les signatures asymétriques numériques sont utilisées). Cette classe utilise soit le code MIC fondée sur des clés symétriques ou bien des signatures numériques asymétriques vérifiées à l'aide de certificats de point d'extrémité source.

Signature et chiffrement imbriqués: cela est utilisé quand le chiffrement est nécessaire en plus de l'authentification ou de la non-répudiation à l'aide d'un certificat de point d'extrémité source. Une signature numérique est d'abord réalisée sur la charge utile, puis la charge utile et la signature sont chiffrées.

ESData prend en charge l'utilisation de multiples justificatifs d'identité pour protéger une seule unité de charge utile.

Chaque classe de sécurité ESData prend en charge les trois options de protections ESData illustrées dans le Tableau 8.5.2.1-1.

Tableau 8.5.2.1-1 – Options de protection ESData

Classe de sécurité ESData	Option de protection ESData	Gestion des clés	Vérification de la source	Non-répudiation
Chiffrement uniquement (8.5.2.2)	Chiffrement à l'aide d'une clé ESData symétrique configurée	Clé symétrique fournie	Symétrique	–
	Chiffrement par fonction TEF	TEF	Symétrique	–
	Chiffrement à l'aide d'un certificat de point d'extrémité cible	Certificat	–	–
Signature uniquement (8.5.2.3)	Code MIC fondé sur une clé ESData symétrique configurée	Clé symétrique fournie	Symétrique	–
	Code MIC utilisant la fonction TEF	TEF	Symétrique	–
	Signature numérique à l'aide d'un certificat de point d'extrémité source	Certificat	Certificat	Certificat
Signature et chiffrement imbriqués (8.5.2.4)	Signature numérique à l'aide d'un certificat de point d'extrémité source suivie d'une combinaison d'options de protection par chiffrement seul	Clés symétriques fournies, fonctions TEF ou certificats de chiffrement. Certificat de signature.	Certificat	Certificat

8.5.2.2 Classe de sécurité ESData de chiffrement seul

8.5.2.2.1 Aperçu de la classe de sécurité ESData de chiffrement seul

Les options de protection ESData prises en charge pour la classe de sécurité de chiffrement seul sont répertoriées dans le Tableau 8.5.2.1-1 intitulé "Options de protection ESData".

L'option ESData de chiffrement seul prend en charge le chiffrement grâce à une combinaison d'options de protection et à plusieurs justificatifs d'identité pour chaque option de protection.

Mode de chiffrement ESData: la classe de sécurité ESData prend en charge deux modes de chiffrement principaux:

Mode de chiffrement direct ESData: dans ce mode, une clé symétrique est utilisée directement par l'algorithme de chiffrement pour sécuriser la charge utile. Le mode de chiffrement direct est recommandé uniquement lorsque les critères suivants sont remplis:

La minimisation de la surcharge d'objets de données est hautement prioritaire.

La fonction de chiffrement ne sera pas utilisée avec la même valeur de clé plus de 2^{32} fois, du moins pour la méthode AES-GCM, pour les raisons exposées au § 8.4 de [IETF RFC 7518].

Ce mode sera utilisé uniquement dans le cas où une seule clé symétrique est employée pour protéger la charge utile.

Mode par clé chiffrée ESData: dans ce mode, la clé de chiffrement du contenu (CEK), utilisée par l'algorithme de chiffrement pour sécuriser la charge utile, est chiffrée à l'aide d'un ou plusieurs justificatifs d'identité. La clé CEK ainsi chiffrée est ensuite ajoutée à un en-tête avec les données sécurisées.

Contraintes d'applicabilité du mode de chiffrement: dans les cas où:

– le chiffrement réalisé à l'aide d'une clé ESData symétrique fournie est appliqué à l'aide d'une clé symétrique fournie; ou

– le chiffrement par fonction TEF est appliqué à l'aide d'une seule clé symétrique enregistrée par une fonction TEF;

Le mode de chiffrement direct ou le mode par clé chiffrée peut être employé.

Dans tous les autres cas, le mode par clé chiffrée est utilisé.

Séquence de haut niveau d'événements: la séquence d'événements qui se produit lorsque la classe de sécurité de chiffrement seul est utilisée est décrite ci-dessous.

NOTE – La présente Recommandation ne décrit pas les processus qui permettent au point d'extrémité source et aux points d'extrémité cibles de déterminer quels justificatifs d'identité doivent être utilisés pour sécuriser une charge utile ni quel algorithme doit être exécuté.

A Configuration des justificatifs d'identité: le point d'extrémité source obtient les justificatifs d'identité nécessaires pour sécuriser la charge utile des points d'extrémité cibles souhaités. Cela comprend n'importe quelle combinaison d'options de protection, plusieurs justificatifs d'identité étant autorisés pour chaque option de protection:

Chiffrement à l'aide d'une clé ESData symétrique fournie: le point d'extrémité source et les points d'extrémité cibles sont fournis grâce à une clé ESData symétrique fournie conformément au § 8.5.2.2.2 (Chiffrement à l'aide d'une clé ESData symétrique fournie).

Chiffrement par fonction TEF: le point d'extrémité source génère une clé symétrique secrète aléatoire enregistrée par une fonction TEF et enregistre cette clé grâce à la fonction TEF comme indiqué au § 8.5.2.2.3 (Chiffrement à l'aide d'une fonction génératrice de confiance).

Chiffrement par certificats: le point d'extrémité source obtient le certificat du point d'extrémité cible comme indiqué au § 8.5.2.2.4 (Chiffrement à l'aide d'un certificat de point d'extrémité cible).

B Gestion des clés CEK du point d'extrémité source:

Si le mode de chiffrement direct doit être utilisé, la clé ESData symétrique fournie ou la clé symétrique enregistrée par la fonction TEF doit être directement utilisée comme clé CEK. Le recours au mode de chiffrement direct doit être indiqué dans les paramètres des *en-têtes ESData* de l'enveloppe ESData. La clé ESData symétrique fournie ou la clé symétrique enregistrée par la fonction TEF doit être directement identifiée dans les en-têtes.

Autrement, le point d'extrémité source génère une valeur secrète aléatoire pour la clé CEK et chiffre cette dernière à l'aide des justificatifs d'identité obtenus durant la phase A (Gestion des justificatifs d'identité) selon ce qui est indiqué au § 8.5.2.2.2, 8.5.2.2.3 et 8.5.2.2.4. Chaque clé CEK chiffrée est ajoutée aux en-têtes avec l'identificateur du justificatif d'identité à utiliser pour déchiffrer la clé CEK. La valeur de la clé CEK peut être utilisée pour une ou plusieurs charges utiles.

C Chiffrement du point d'extrémité source:

C.1 L'algorithme de chiffrement doit être identifié dans les en-têtes.

C.2 Le point d'extrémité source doit appliquer le processus de chiffrement de l'algorithme identifié à la charge utile à l'aide de la clé CEK. Le texte en clair est chiffré pour donner le cryptogramme. L'intégrité du texte en clair et des données d'authentification associées (AAD) est protégée par le code MIC généré.

C.3 Le point d'extrémité source crée l'enveloppe à partir des en-têtes, du cryptogramme, des données d'authentification associées et du code MIC. Ce processus peut inclure l'encodage des données à l'aide du protocole base64, par exemple.

La présente Recommandation ne précise pas la manière dont l'enveloppe est obtenue ou fournie aux points d'extrémité cibles. Les étapes suivantes sont exécutées pour chaque point d'extrémité cible:

D Gestion des clés CEK du point d'extrémité cible:

D.1 Le point d'extrémité cible analyse l'enveloppe en appliquant tout procédé d'encodage nécessaire, puis extrait les paramètres des en-têtes.

D.2 Si le mode de chiffrement direct est indiqué dans les en-têtes, le point d'extrémité cible utilise les identificateurs de justificatif d'identité contenus dans ces derniers pour obtenir la clé ESData symétrique fournie identifiée ou la clé symétrique enregistrée à l'aide de la fonction TEF (décrites respectivement aux paragraphes 8.2.2.2 et 8.5.2.2.3). Le point d'extrémité cible utilise la clé symétrique directement en tant que clé CEK.

D.3 Autrement, le point d'extrémité cible doit utiliser les identificateurs des justificatifs d'identité contenus dans les en-têtes afin d'identifier une clé CEK qui peut être déchiffrée par un justificatif d'identité connu ou auquel le point d'extrémité cible a accès. Le point d'extrémité cible obtient ce justificatif d'identité et déchiffre la clé CEK chiffrée selon les indications des paragraphes 8.5.2.2.2 (procédure par clé ESData symétrique fournie), 8.5.2.2.3 (procédure par fonction TEF) et 8.5.2.2.4 (procédure par certificat de point d'extrémité cible). La cible doit utiliser la clé CEK obtenue pour traiter la charge utile sécurisée de l'enveloppe. Le point d'extrémité cible peut mettre la valeur de la clé CEK en cache, car elle peut être utilisée pour protéger les charges utiles suivantes.

E Déchiffrement du point d'extrémité cible:

E.1 Le point d'extrémité cible détermine l'algorithme de chiffrement adapté identifié dans les en-têtes.

E.2 Le point d'extrémité cible soumet le cryptogramme, les données d'authentification associées et le code MIC au processus de déchiffrement de l'algorithme identifié à l'aide de la clé CEK, puis produit le texte en clair et des données d'authentification associées vérifiés.

8.5.2.2.2 Chiffrement à l'aide d'une clé ESData symétrique fournie

Dans le cadre de cette option de protection, une clé ESData symétrique fournie, son identificateur et, de manière facultative, sa durée de vie, doivent être transmis au point d'extrémité source et à chaque point d'extrémité cible. Ce justificatif d'identité doit être configuré de l'une des manières suivantes:

– par préconfiguration;

– à l'aide d'un cadre de configuration à distance de la sécurité (RSPF) défini dans le § 8.3; ou

– par l'établissement d'une clé à l'aide d'un certificat de sécurité de bout en bout entre l'expéditeur et le destinataire conformément au § 8.7 "Établissement d'une clé à l'aide d'un certificat de sécurité de bout en bout (ESCertKE)".

8.5.2.2.3 Chiffrement à l'aide d'une fonction génératrice de confiance (TEF)

Cette procédure est décrite au § 8.6.

8.5.2.2.4 Chiffrement à l'aide de certificats de point d'extrémité cible

8.5.2.2.4.1 Association d'un certificat de clé publique à un point d'extrémité cible

Dans le cadre de cette option de protection, chaque point d'extrémité cible doit recevoir un certificat de clé publique que le point d'extrémité source considère comme étant associé au point d'extrémité cible souhaité. Les options suivantes sont prises en charge:

Les certificats de point d'extrémité cible peuvent utiliser les variantes de certificat de clé publique suivantes mentionnées au § 8.1.2.1 (Variantes de certificats de clé publique):

Dans le cas d'un certificat de clé publique brute, le point d'extrémité source doit être configuré de manière sûre (soit directement ou à distance) de sorte à associer le point d'extrémité cible à la clé publique brute ou à son hachage. Les détails de cette configuration ne sont pas exposés dans la présente spécification.

Dans le cas d'un certificat de dispositif:

Le point d'extrémité source doit être configuré de manière sûre avec l'ancre de confiance contenue dans la chaîne de certificats du certificat de dispositif, généralement durant la configuration initiale.

Le point d'extrémité source doit être configuré de manière sûre de sorte à associer le point d'extrémité cible à l'identificateur unique mondialement de l'instance matérielle. Les détails de cette configuration ne sont pas exposés dans la présente spécification.

Dans le cas d'un certificat AE-ID ou CSE-ID, le point d'extrémité source doit être configuré de manière sûre avec l'ancre de confiance contenue dans la chaîne de certificats du certificat AE-ID ou CSE-IS, généralement durant la configuration initiale. Le point d'extrémité source confirme ensuite que le point d'extrémité cible présentant un identificateur AE-ID ou CSE-ID spécifique est associé au certificat qui contient ledit identificateur.

Les certificats du point d'extrémité cible peuvent utiliser d'autres infrastructures à clés publiques, notamment si ledit point d'extrémité n'est pas un système oneM2M, mais qu'il fonctionne avec le système oneM2M. La présente Recommandation ne donne aucune garantie d'interopérabilité quand de tels certificats sont utilisés.

Les clés publiques visant à vérifier les signatures ne peuvent pas être utilisées pour cette option de protection.

8.5.2.2.4.2 Obtention de certificats de point d'extrémité cible

Le point d'extrémité source ne peut pas sécuriser un message destiné au point d'extrémité cible avant d'obtenir le certificat du point d'extrémité cible. La présente spécification n'impose pas le mécanisme par lequel le certificat du point d'extrémité cible est fourni au point d'extrémité source. Il existe une grande variété de mécanismes adaptés à cette procédure. L'attribut *e2ESecurityParameters* constitue un mécanisme de la norme oneM2M qui permet au point d'extrémité source de récupérer les certificats associés à une entité de services communs ou à une entité d'application.

L'entité d'application d'un point d'extrémité cible peut mettre les certificats à disposition de l'attribut *e2ESecurityParameters* de la ressource <*AE*> qui représente ladite entité d'application. Le processus de récupération n'est pas un mécanisme entièrement sûr pour associer le point d'extrémité cible au certificat. Les indications du § 8.5.2.2.4.1 (Association du certificat de clé publique aux points d'extrémité cibles) doivent être appliquées.

L'entité de services communs d'un point d'extrémité cible peut mettre les certificats à disposition de l'attribut *e2ESecurityParameters* des ressources <*CSE*> et <*remoteCSE*> qui représente ladite entité de services communs. Le processus de récupération n'est pas un mécanisme entièrement sûr pour associer le point d'extrémité cible au certificat. Les indications du § 8.5.2.2.4.1 (Association du certificat de clé publique aux points d'extrémité cibles) doivent être appliquées.

8.5.2.3 Classe de sécurité ESData de signature seule

8.5.2.3.1 Aperçu de la classe de sécurité ESData de signature seule

Les options de protection ESData prises en charge pour la classe de sécurité de signature seule sont répertoriées dans le Tableau 8.5.2.1-1 intitulé "Options de protection ESData".

NOTE 1 – La présente spécification prend en charge uniquement une option de protection ESData de signature seule, mais le paragraphe est structuré pour permettre la prise en charge d'options de protection supplémentaires de ce genre si cela est préférable à l'avenir.

L'option ESData de signature seule prend en charge le chiffrement grâce à une combinaison d'options de protection et à plusieurs justificatifs d'identité pour chaque option de protection.

Séquence de haut niveau d'événements: la séquence d'événements qui se produit lorsque la classe de sécurité de signature seule est utilisée est décrite ci-dessous.

NOTE 2 – La présente Recommandation ne décrit pas les processus qui permettent au point d'extrémité source et aux points d'extrémité cibles de déterminer quels justificatifs d'identité doivent être utilisés pour signer une charge utile ni quels algorithmes doivent être exécutés.

A Configuration des justificatifs d'identité: le point d'extrémité source obtient les justificatifs d'identité nécessaires pour signer la charge utile des points d'extrémité cibles souhaités. Cela comprend n'importe quelle combinaison d'options de protection, plusieurs justificatifs d'identité étant autorisés pour chaque option de protection:

Code MIC utilisant une clé ESData symétrique fournie: le point d'extrémité source et les points d'extrémité cibles sont configurés grâce à une clé ESData symétrique fournie conformément au § 8.5.2.2.2 (Chiffrement à l'aide d'une clé ESData symétrique fournie).

Code MIC utilisant la fonction TEF: le point d'extrémité source génère une clé symétrique secrète aléatoire enregistrée par une fonction TEF et enregistre cette clé grâce à la fonction TEF comme indiqué au § 8.5.2.2.3 (Chiffrement à l'aide d'une fonction génératrice de confiance).

Signature numérique à l'aide de certificats de point d'extrémité source: Le point d'extrémité source sélectionne une clé privée et le certificat de point d'extrémité source correspondant de la façon décrite au § 8.5.2.3.2 (Signature numérique à l'aide de certificats de point d'extrémité source).

B Signature du point d'extrémité source:

B.1 La charge utile est encodée (à l'aide du protocole base64, par exemple).

B.2 Pour chaque justificatif d'identité, le point d'extrémité source génère un tableau d'éléments de données comme suit:

B.2.i Le point d'extrémité source forme un en-tête, qui identifie la signature numérique ou l'algorithme du code MIC, ainsi que le justificatif d'identité qu'un point d'extrémité cible peut utiliser pour vérifier la signature numérique ou le code MIC. Si nécessaire, l'en-tête est également encodé (à l'aide du protocole base64, par exemple).

B.2.ii Le point d'extrémité source génère une signature ou un code MIC en appliquant la signature numérique ou l'algorithme MIC à la charge utile et à l'en-tête au moyen du justificatif d'identité et de l'encodage adaptés (base64, par exemple).

B.2.iii Le point d'extrémité source forme un élément de données à partir des en-têtes, de la charge utile et de la signature ou du code MIC.

B.3 Le point d'extrémité source forme l'enveloppe à partir de la charge utile encodée et du tableau d'éléments de données générés à l'étape B.2.

La présente Recommandation ne précise pas la manière dont l'enveloppe est obtenue ou fournie aux points d'extrémité cibles. Les étapes suivantes sont exécutées pour chaque point d'extrémité cible.

C Vérification du point d'extrémité cible:

C.1 Le point d'extrémité cible analyse l'enveloppe et en extrait la charge utile et le tableau d'éléments de données encodés, chacun contenant un en-tête et une signature ou un code MIC.

C.2 Le point d'extrémité cible examine le tableau d'éléments de données pour identifier ces derniers, qui peuvent être vérifiés à l'aide d'un justificatif d'identité auquel un point d'extrémité cible peut se fier. Les actions suivantes sont effectuées pour chaque élément de données:

C.2.i Le point d'extrémité cible obtient le justificatif d'identité défini comme indiqué aux paragraphes 8.5.2.2.2 (procédure par clé ESData symétrique fournie), 8.5.2.2.3 (procédure par fonction TEF) et 8.5.2.3.2 (procédure par certificat de point d'extrémité source).

C.2.ii Le point d'extrémité cible vérifie le code MIC ou la signature grâce au justificatif d'identité.

C.3 Le point d'extrémité cible décode la charge utile encodée et vérifiée (on obtient ainsi la charge utile d'origine) et enregistre les justificatifs d'identité utilisés pour vérifier la charge utile.

8.5.2.3.2 Signature numérique à l'aide d'un certificat de point d'extrémité source

8.5.2.3.2.1 Association d'un certificat de clé publique à un point d'extrémité source:

Dans le cadre de cette option de protection, chaque point d'extrémité source doit recevoir un certificat de clé publique que le point d'extrémité cible considère comme étant associé au point d'extrémité source souhaité. Les options suivantes sont prises en charge:

Les certificats de point d'extrémité source peuvent utiliser les variantes de certificat de clé publique suivantes mentionnées au § 8.1.2.1 (Variantes de certificats de clé publique):

Dans le cas d'un certificat de clé publique brute, le point d'extrémité cible doit être configuré de manière sûre (soit directement ou à distance) de sorte à associer le point d'extrémité source à la clé publique brute ou à son hachage. Les détails de cette configuration ne sont pas exposés dans la présente spécification.

Dans le cas d'un certificat de dispositif:

Le point d'extrémité cible doit être configuré de manière sûre avec l'ancre de confiance contenue dans la chaîne de certificats du certificat de dispositif, généralement durant la configuration initiale.

Le point d'extrémité cible doit être configuré de manière sûre de sorte à associer le point d'extrémité source à l'identificateur unique mondialement de l'instance matérielle. Les détails de cette configuration ne sont pas exposés dans la présente spécification.

Dans le cas d'un certificat AE-ID ou CSE-ID, le point d'extrémité cible doit être configuré de manière sûre avec l'ancre de confiance contenue dans la chaîne de certificats du certificat AE-ID ou CSE-IS, généralement durant la configuration initiale. Le point d'extrémité cible confirme ensuite que le point d'extrémité source présentant un identificateur AE-ID ou CSE-ID spécifique est associé au certificat qui contient ledit identificateur.

Les certificats du point d'extrémité cible peuvent utiliser d'autres infrastructures à clés publiques, notamment si ledit point d'extrémité n'est pas un système oneM2M, mais qu'il fonctionne avec le système oneM2M. La présente Recommandation ne donne aucune garantie d'interopérabilité quand de tels certificats sont utilisés.

Les clés publiques visant à vérifier les signatures doivent être utilisées pour cette option de protection.

8.5.2.3.2.2 Obtention de certificats de point d'extrémité source

Le point d'extrémité cible ne peut pas sécuriser un message destiné au point d'extrémité source avant d'obtenir le certificat de ce dernier. La présente spécification n'impose pas le mécanisme par lequel le certificat du point d'extrémité source est fourni au point d'extrémité cible. Il existe une grande variété de mécanismes adaptés à cette procédure. L'attribut *e2ESecurityParameters* constitue un mécanisme de la norme oneM2M qui permet au point d'extrémité source de récupérer les certificats associés à une entité de services communs ou à une entité d'application.

L'entité d'application d'un point d'extrémité source peut mettre les certificats à disposition de l'attribut *e2ESecurityParameters* de la ressource <*AE*> qui représente ladite entité d'application. Le processus de récupération n'est pas un mécanisme entièrement sûr pour associer le point d'extrémité source au certificat. Les indications du § 8.5.2.3.2.1 (Association du certificat de clé publique aux points d'extrémité sources) doivent être appliquées.

L'entité de services communs d'un point d'extrémité source peut mettre les certificats à disposition de l'attribut *e2ESecurityParameters* des ressources <*CSE*> et <*remoteCSE*> qui représente ladite entité de services communs. Le processus de récupération n'est pas un mécanisme entièrement sûr pour associer le point d'extrémité source au certificat. Les indications du § 8.5.2.3.2.1 (Association du certificat de clé publique aux points d'extrémité sources) doivent être appliquées.

8.5.2.4 Signature et chiffrement imbriqués

Pour ces options, les étapes principales suivantes sont effectuées (les étapes de configuration des justificatifs d'identité et de gestion des clés CEK ne sont pas montrées):

Le point d'extrémité source génère une enveloppe interne contenant une ou plusieurs signatures numériques pour la charge utile interne qui utilise au moins un certificat conformément aux indications du § 8.5.2.3 relatives à l'option de protection par signature seule (cf. "Signature numérique à l'aide d'un certificat de point d'extrémité source").

Le point d'extrémité source attribue le texte en clair de la charge utile externe à l'enveloppe interne produite à l'étape 1. Le texte en clair est ensuite chiffré grâce à n'importe quelle combinaison des options de protection par chiffrement seul décrites au § 8.5.2.2. Une enveloppe externe est ainsi obtenue.

La présente Recommandation ne précise pas la manière dont l'enveloppe externe est obtenue ou fournie aux points d'extrémité cibles. Les étapes suivantes sont ensuite exécutées pour chaque point d'extrémité cible:

Le point d'extrémité cible déchiffre l'enveloppe externe produite à l'étape 1 à l'aide de l'une des options de protection par chiffrement seul décrites au § 8.5.2.2, permettant ainsi d'obtenir la charge utile externe, qui est également l'enveloppe interne.

Le point d'extrémité cible vérifie au moins une signature numérique de l'enveloppe interne grâce à un ou plusieurs certificats conformément aux indications du § 8.5.2.3 relatives à l'option de protection par signature seule (cf. "Signature numérique à l'aide d'un certificat de point d'extrémité source") afin d'obtenir la charge utile interne vérifiée.

8.5.3 Détails du protocole de sécurité de bout en bout des données (ESData)

8.5.3.1 Introduction

Les classes de sécurité de bout en bout des primitives (ESData) prennent en charge les protocoles répertoriés dans le Tableau 8.5.3.1-1.

Tableau 8.5.3.1-1 – Classe de sécurité ESData et correspondance avec les protocoles de sécurité fondés sur les formats XML et JSON

Classe de sécurité ESData	XML	JOSE: protocole de sécurité fondé sur le format JSON
Chiffrement uniquement	Chiffrement XML-ENC appliqué à la charge utile ESData	Chiffrement JWE appliqué à la charge utile ESData
Signature uniquement	Signature XML-SIG appliquée à la charge utile ESData	Signature JWS appliquée à la charge utile ESData
Signature et chiffrement imbriqués	Signature XML-SIG appliquée à la charge utile ESData et résultat chiffré par XML-ENC	Signature JWS appliquée à la charge utile ESData et résultat chiffré par JWE

Le protocole JOSE permet de créer des sérialisations JSON polyvalentes en plus de sérialisations compactes plus rigides (dont l'URI est sûre). Il existe donc trois options de sérialisation: XML, JWE/JWS par sérialisation JSON et JWE/JWS par sérialisation compacte.

8.5.3.2 Détails du protocole de la classe de sécurité ESData de chiffrement seul

Pour assurer une certaine cohérence, les algorithmes de gestion des clés fournis sont compatibles avec les protocoles de chiffrement XML-ENC [W3C XMLENC] et JSON Web Encryption (JWE) [IETF RFC 7516].

– Chiffrement direct.
– Enveloppement de clés AES utilisant des clés de 128 bits ou 256 bits.
– Protocole RSA-OAEP avec fonction MGF1 et SHA256.
– Protocole de concordance de clés Diffie-Hellman fondé sur les courbes elliptiques (ECDH) en mode éphémère statique utilisant l'enveloppement de clés AES.

Le Tableau 8.5.3.2-1 répertorie les algorithmes de gestion de clés pris en charge par le protocole de chiffrement XML pour la classe de sécurité ESData de chiffrement seul.

Tableau 8.5.3.2-1 – Algorithmes de gestion de clés pris en charge par le protocole de chiffrement XML pour la classe de sécurité ESData de chiffrement seul

Gestion des clés	Algorithme		<xenc:EncryptionMethod Algorithm=".."> pour le chiffrement de la clé	Autres paramètres
Chiffrement direct	non disponible		non disponible	<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> <ds:KeyName>John Smith</ds:KeyName>
Envelop-pement de clés symétriques	Envelop-pement de clés AES avec	clé 128 bits	http://www.w3.org/2001/04/xmlenc#kwaes128	
		clé 192 bits	http://www.w3.org/2001/04/xmlenc#kwaes192	
		clé 256 bits	http://www.w3.org/2001/04/xmlenc#kwaes256	
RSA	Protocole RSA-OAEP avec fonction MFG1 et SHA256		http://www.w3.org/2009/xmlenc11#rsa-oaep	<xenc11:MGF Algorithm="http://www.w3.org/2009/xmlenc11#mgf1sha256>
Concordance de clés ECDH	ECDH-ES avec envelop-pement de clés AES	clé 128 bits	http://www.w3.org/2001/04/xmlenc#kwaes128	<xenc:AgreementMethod Algorithm="http://www.w3.org/2009/xmlenc11#ECDH-ES">
		clé 192 bits	http://www.w3.org/2001/04/xmlenc#kwaes192	
		clé 256 bits	http://www.w3.org/2001/04/xmlenc#kwaes256	

Le Tableau 8.5.3.2-2 répertorie les algorithmes de chiffrement de la charge utile pris en charge par le protocole de chiffrement XML pour la classe de sécurité ESData de chiffrement seul.

Tableau 8.5.3.2-2 – Algorithmes de chiffrement de la charge utile pris en charge par le protocole de chiffrement XML pour la classe de sécurité ESData de chiffrement seul

Algorithme de chiffrement de la charge utile		<EncryptionMethod Algorithm="..">
AES-GCM avec	clé 128 bits	http://www.w3.org/2009/xmlenc11#aes128gcm
	clé 192 bits	http://www.w3.org/2009/xmlenc11#aes192gcm
	clé 256 bits	http://www.w3.org/2009/xmlenc11#aes256gcm

La sortie générée par le chiffrement XML est sérialisée sous la forme d'un objet XML. L'objet de chiffrement XML peut être transmis en clair (non encodé) ou être encodé en base64.

Le Tableau 8.5.3.2-3 répertorie les algorithmes de gestion de clés pris en charge par le protocole de chiffrement JWE pour la classe de sécurité ESData de chiffrement seul.

Tableau 8.5.3.2-3 – Algorithmes de gestion de clés pris en charge par le protocole JSON Web Encryption (JWE) pour la classe de sécurité ESData de chiffrement seul

Gestion des clés	Algorithme		"alg":".."
Chiffrement direct	non disponible		dir
Enveloppement de clés symétriques	Enveloppement de clés AES avec	clé 128 bits	A128KW
		clé 192 bits	A192KW
		clé 256 bits	A256KW
RSA	Protocole RSA-OAEP avec fonction MFG1 et SHA256		"alg": "RSA-OAEP-256"
Concordance de clés ECDH	ECDH-ES avec enveloppement de clés AES	clé 128 bits	ECDH-ES + A128KW
		clé 192 bits	ECDH-ES + A192KW
		clé 256 bits	ECDH-ES + A256KW

Le Tableau 8.5.3.2-4 répertorie les algorithmes de chiffrement de la charge utile pris en charge par le protocole de chiffrement JWE pour la classe de sécurité ESData de chiffrement seul.

Tableau 8.5.3.2-4 – Algorithmes de chiffrement de la charge utile pris en charge par le protocole JSON Web Encryption (JWE) pour la classe de sécurité ESData de chiffrement seul

Algorithme de chiffrement de la charge utile		"enc":".."
AES-GCM avec	clé 128 bits	A128GCM
	clé 192 bits	A192GCM
	clé 256 bits	A256GCM

La sortie générée par le protocole JWE est conforme à la sérialisation JSON JWE ou à une sérialisation compacte JWE à URI sûre. La sérialisation JSON JWE peut être transmise en clair (non encodée) ou être encodée en base64. La Recommandation [ITU-T Y.4500.4] définit le type de données m2m:e2eCompactJWE pour la sérialisation compacte JWE.

8.5.3.3 Détails de la classe de sécurité ESData de signature seule

Pour assurer une certaine cohérence, les types de signatures fournis sont compatibles avec les protocoles de signature XML-Signature [W3C XMLSIG] et JSON Web Signature (JWS) [IETF RFC 7515].

– HMAC utilisant les fonctions SHA-256, SHA-384 ou SHA-512.

– Signature RSA utilisant PKCS1-v1.5 et MGF1 avec les fonctions SHA-256, SHA-384 ou SHA-512.

– Signature ECDSA utilisant la courbe P-256, P-384 ou P-512 avec les fonctions SHA-256, SHA-284 ou SHA-512 respectivement.

Le Tableau 8.5.3.3-1 répertorie les algorithmes pris en charge par le protocole de signature XML-SIG pour la classe de sécurité ESData de signature seule.

Tableau 8.5.3.3-1 – Algorithmes pris en charge par le protocole de signature XML-Signature pour la classe de sécurité ESData de signature seule

Type de signature	Algorithme		<SignatureMethod Algorithm="..">
HMAC	SHA-256		http://www.w3.org/2001/04/xmldsigmore#hmacsha256
	SHA-384		http://www.w3.org/2001/04/xmldsigmore#hmacsha384
	SHA-512		http://www.w3.org/2001/04/xmldsigmore#hmacsha512
RSA	RSA PKCS1-v1.5 et MGF1 avec:	SHA-256	http://www.w3.org/2001/04/xmldsigmore#rsasha256
		SHA-384	http://www.w3.org/2001/04/xmldsigmore#rsasha384
		SHA-512	http://www.w3.org/2001/04/xmldsigmore#rsasha512
ECDSA	P-256 et SHA-256		http://www.w3.org/2001/04/xmldsigmore#ecdsasha256
	P-384 et SHA-384		http://www.w3.org/2001/04/xmldsigmore#ecdsasha384
	P-512 et SHA-512		http://www.w3.org/2001/04/xmldsigmore#ecdsasha512

L'objet de signature XML peut être transmis en clair (non encodé) ou être encodé en base64.

Le Tableau 8.5.3.3-2 répertorie les algorithmes pris en charge par le protocole de signature JWS pour la classe de sécurité ESData de signature seule.

Tableau 8.5.3.3-2 – Algorithmes pris en charge par le protocole de signature JSON Web Signature (JWS) pour la classe de sécurité ESData de signature seule

Type de signature	Algorithme		"alg":".."
HMAC	SHA-256		HS256
	SHA-384		HS384
	SHA-512		HS512
RSA	RSA PKCS1-v1.5 et MGF1 avec:	SHA-256	RS256
		SHA-384	RS384
		SHA-512	RS512
ECDSA	P-256 et SHA-256		ES256
	P-384 et SHA-384		ES384
	P-512 et SHA-512		ES512

La sortie générée par le protocole JWS est conforme à la sérialisation JSON JWS ou à une sérialisation compacte JWS à URI sûre. La sérialisation JSON JWS peut être transmise en clair (non encodée) ou être encodée en base64. La Recommandation [ITU-T Y.4500.4] définit le type de données m2m:e2eCompactJWS pour la sérialisation compacte JWS.

8.5.3.4 Détails de la classe de sécurité ESData de signature et de chiffrement imbriqués

Les étapes principales de la classe de sécurité ESData de signature et de chiffrement imbriqués sont décrites au § 8.5.2.4. L'enveloppe interne doit être générée et traitée selon un ou plusieurs des types de signature RSA ou ECDSA décrits au § 8.5.3.3. De plus, elle doit être générée et traitée à l'aide de n'importe quelle combinaison des algorithmes de gestion de clés décrits au § 8.5.3.2.

8.6 Cadres de sécurité de bout en bout à distance

8.6.1 Aperçu de la configuration et de l'enregistrement à distance des justificatifs d'identité pour la procédure de sécurité de bout en bout

8.6.1.1 Introduction

Le cadre de configuration à distance destiné à la procédure de sécurité de bout en bout doit permettre à une entité d'enregistrer et de configurer des justificatifs d'identité de bout en bout à l'aide d'une fonction génératrice de confiance afin d'assurer une sécurité de bout en bout. Une fonction d'inscription M2M, une fonction d'authentification M2M ou une entité de services communs de nœud intermédiaire (MN-CSE) pouvant enregistrer et configurer des justificatifs d'identité de sécurité de bout en bout peut agir en tant que fonction génératrice de confiance pour la sécurité de bout en bout.

Les justificatifs d'identité de sécurité de bout en bout calculés peuvent être utilisés pour fournir les mécanismes de protection suivants:

– Protection de l'intégrité et de l'authenticité du message (primitive) à l'aide d'un code de vérification de l'intégrité du message (MIC)

– Vérification de la confidentialité du message (primitive).

– Intégrité et authenticité des données (attributs) à l'aide d'une étiquette d'intégrité des données (DIT).

– Vérification de la confidentialité des données (attributs).

Les messages et données (attributs) protégés peuvent être enveloppés à l'aide des cadres ESPrim et ESData respectivement et aux mécanismes décrits aux paragraphes 8.4 (Cadre de sécurité de bout en bout des primitives) et 8.5 (Cadre de sécurité de bout en bout des données). L'authenticité, l'intégrité et la confidentialité des messages sont garanties grâce au cadre ESPrim. Quant à l'intégrité et à la confidentialité des données d'application (attributs), elles sont garanties à l'aide d'objets ESData.

La sécurité de bout en bout peut être assurée au moyen des éléments suivants:

– processus de configuration à distance de la sécurité reposant sur les indications du § 8.3 et décrit au § 8.6.2;

– justificatifs d'identité générés par la source décrits au § 8.6.3.

8.6.1.2 Description générale du processus d'enregistrement et de configuration à distance de la procédure de sécurité de bout en bout

Ce paragraphe décrit les mécanismes pouvant servir à la génération, à l'enregistrement et à la configuration des justificatifs d'identité qui doivent être utilisés pour garantir la sécurité de bout en bout. En fonction des exigences ou du profil de sécurité associés à l'entité (par exemple, à une entité d'application) et indiqués dans la ressource *<e2ESecurityCapabilities>* décrite au § 9.6.1.3.2 de la Recommandation [ITU-T Y.4500.1], des justificatifs d'identité de sécurité de bout en bout doivent être générés. Les mécanismes de configuration à distance tirent parti des mécanismes décrits au § 8.3 relatif aux cadres de configuration à distance de la sécurité et les étendent afin que les justificatifs d'identité de sécurité de bout en bout puissent être enregistrés et configurés pour les entités séparées de plus d'un bond l'une de l'autre. La Figure 8.6.1.2-1 illustre la séquence d'étapes principales à suivre pour enregistrer à distance et configurer des justificatifs d'identité de bout en bout.

La procédure de sécurité de bout en bout comprend les étapes suivantes:

– un point d'extrémité ESP source qui identifie l'ensemble de mécanismes de sécurité adapté et qui génère les justificatifs d'identité appropriés;

– l'enregistrement des justificatifs d'identité à l'aide d'une fonction génératrice de confiance;

– la fonction TEF fournit les justificatifs d'identité de bout en bout à un point d'extrémité ESP cible;

– le traitement d'éléments ESData et ESPrim grâce aux justificatifs d'identité de bout en bout.

Quand un processus de configuration à distance de la sécurité est exécuté, la fonction TEF doit effectuer les étapes 1 et 2 en premier. Si des justificatifs d'identité de sécurité de bout en bout générés par la source sont utilisés, le point d'extrémité ESF source doit effectuer les étapes 1 et 2.

La Figure 8.6.1.2-1 résume le processus d'enregistrement et de fourniture des justificatifs d'identité.

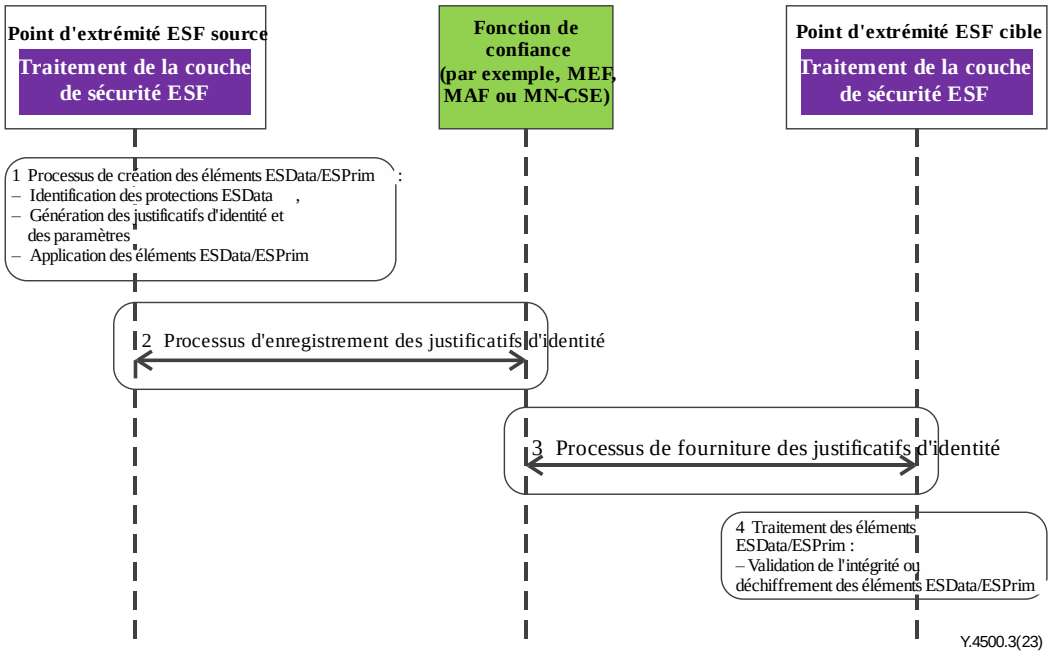


Figure 8.6.1.2-1 – Résumé du processus d'enregistrement et de fourniture des justificatifs d'identité

Le processus d'enregistrement et de fourniture de justificatifs d'identité de bout en bout pour la fourniture d'objets ESData/ESPrim comporte les étapes suivantes:

Le processus de création des objets ESData/ESPrim par le point d'extrémité ESF source, qui comprend les étapes suivantes:

a) L'identification des mécanismes de protection en fonction des exigences de sécurité associées aux données d'application;

b) La génération des justificatifs d'identité de sécurité appropriés et des paramètres associés en fonction des exigences de sécurité;

c) La protection de l'application au moyen des justificatifs d'identité de sécurité et des paramètres associés en vue de générer l'objet ESData/ESPrim.

NOTE 1 – Dans le cas du processus de configuration à distance de la sécurité, les étapes a et b sont effectuées par une fonction génératrice de confiance (TEF). En revanche, dans le cas d'une génération par la source, cette dernière exécute les étapes susmentionnées.

Processus d'enregistrement des justificatifs d'identité:

a) Le point d'extrémité ESF source enregistre les justificatifs d'identité et les paramètres associés à l'aide d'une fonction TEF;

b) Le point d'extrémité ESP source doit configurer l'identité des points d'extrémité ESF cibles qui peuvent être configurés à l'aide des justificatifs d'identité de bout en bout et des paramètres associés.

NOTE 2 – Dans le cas du processus de configuration à distance de la sécurité, le processus d'enregistrement des justificatifs d'identité est effectué par une fonction TEF. En revanche, dans le cas d'une génération par la source, cette dernière exécute les étapes susmentionnées.

Processus de fourniture/demande de justificatifs d'identité:

a) Un point d'extrémité ESF cible peut demander les justificatifs d'identité des objets ESData/ESPrim à l'aide d'un identificateur correspondant obtenu dans les cadres ESData/ESPrim;

b) En fonction des informations d'autorisation fournies durant le processus d'enregistrement des justificatifs d'identité et de l'identificateur correspondant, la fonction TEF fournit les justificatifs d'identité appropriés et les paramètres cryptographiques associés au point d'extrémité ESF cible authentifié et autorisé.

Traitement des éléments ESData/ESPrim:

a) Le point d'extrémité ESF cible utilise les justificatifs d'identité fournis par la fonction TEF afin de traiter les objets ESData/ESPrim.

b) Le traitement des objets ESData/ESPrim comprend la vérification de l'intégrité et l'authentification des données d'application ou le déchiffrement des données et des messages.

8.6.2 Processus de configuration à distance de la sécurité pour les justificatifs d'identité de sécurité de bout en bout

Ce paragraphe décrit la configuration à distance de justificatifs d'identité symétriques de sécurité de bout en bout. Les justificatifs d'identité de sécurité de bout en bout doivent être générés, après la fin de la configuration à distance des justificatifs d'identité symétriques, à l'aide d'une clé symétrique configurée ou de l'établissement d'une association de sécurité de la clé symétrique fondée sur la fonction MAF, comme cela est décrit au § 8.3.

En fonction des exigences principales, les justificatifs d'identité de bout en bout appropriés peuvent être générés durant le processus de configuration à distance de la sécurité à l'aide de justificatifs d'identité préconfigurés. La Figure 8.6.2-1 illustre un processus de génération de clés de haut niveau.

Dans le cadre du mécanisme de génération de clés de bout en bout, l'entité inscrite et la cible d'inscription génèrent des justificatifs d'identité de bout en bout en utilisant le justificatif d'identité Kpsa en tant que clé principale afin de générer une clé principale de bout en bout. Si l'entité inscrite est une entité d'application (point d'extrémité ESF source) et que la cible d'inscription est une entité de services communs (point d'extrémité ESF cible), le justificatif d'identité principal de bout en bout Ke2e_master est généré. On trouvera ci-dessous un exemple de génération de clé de bout en bout effectuée selon les indications de [IETF RFC 5869].

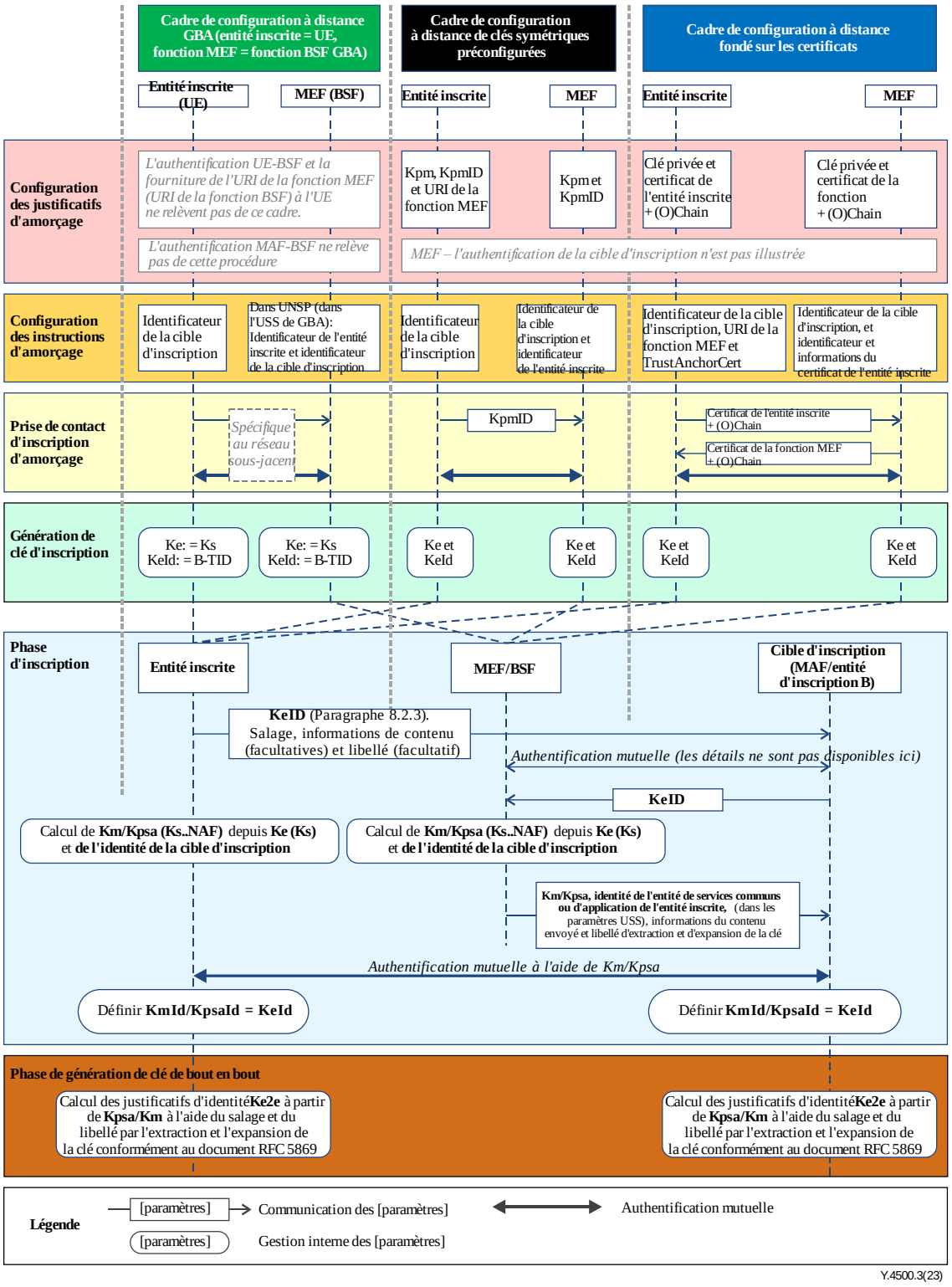


Figure 8.6.2-1 – Résumé des cadres de configuration à distance de la sécurité de bout en bout

Configuration des justificatifs d'amorçage: elle peut dépendre du type de cadre de configuration à distance utilisé. Durant le processus de configuration à distance de clés symétriques, la clé symétrique de l'entité inscrite (Kpm) et l'identificateur de la clé symétrique préconfigurée correspondante, notée KpmID, sont fournies à l'entité inscrite, qui peut être le point d'extrémité ESF source et les cibles d'inscription (point d'extrémité ESF cible). De plus, l'adresse (URI) de la fonction TEF est fournie au point d'extrémité ESF source. Le mécanisme est composé des procédures décrites au § 8.3.2.1.

Configuration des instructions d'amorçage: le point d'extrémité ESF source (entité inscrite) et la fonction TEF sont configurés à l'aide des informations nécessaires à l'autorisation de la configuration à distance:

Le point d'extrémité ESF source (entité inscrite) est configuré avec les arguments suivants afin de démarrer la configuration à distance:

a) Le profil de sécurité du point d'extrémité ESF cible: le profil de sécurité du point d'extrémité ESF cible et les fonctionnalités de sécurité associées décrites dans la ressource <*e2ESecurityCapabilities*> peuvent être utilisés pour identifier les types de mécanismes de sécurité pouvant servir à garantir la sécurité de bout en bout.

b) L'identité du point d'extrémité ESF cible: identification du point d'extrémité ESF cible auquel le point d'extrémité ESF source doit fournir les justificatifs d'identité de sécurité de bout en bout.

c) Le profil de sécurité du point d'extrémité ESF cible: le profil de sécurité du point d'extrémité ESF cible et les fonctionnalités de sécurité associées décrites dans la ressource <*e2ESecurityCapabilities*> peuvent être utilisés pour identifier les types de mécanismes de sécurité pouvant servir à garantir la sécurité de bout en bout.

d) Le point d'extrémité ESF source lie ces arguments à la fonction TEF. La fonction TEF peut être associée au point d'extrémité ESF source à l'aide de l'identificateur de la clé symétrique préconfigurée de l'entité inscrite (KpmID) et de son URI.

Afin d'être autorisée à configurer à distance le point d'extrémité ESF source pour un point d'extrémité ESF cible, la procédure d'inscription M2M ou la fonction TEF est configurée avec les arguments suivants:

a) L'identité du point d'extrémité ESF cible: identification du point d'extrémité ESF cible auquel le point d'extrémité ESF source doit fournir les justificatifs d'identité de sécurité de bout en bout.

b) L'identificateur de l'entité de services communs ou d'application assigné au point d'extrémité ESF source (identificateur du point d'extrémité ESF source). La fonction TEF doit fournir l'identificateur du point d'extrémité ESF source accompagné du justificatif d'identité Km ou Kpsa au point d'extrémité ESF cible quand ce dernier le demande.

c) Le profil de sécurité du point d'extrémité ESF source: le profil de sécurité du point d'extrémité ESF source indique le niveau de sécurité attendu décrit dans la ressource <*e2ESecurityCapabilities*> (voir § 9.6.3 de la Recommandation [ITU-T Y.4500.1]) associée au point d'extrémité ESF source.

d) Le profil de sécurité du point d'extrémité ESF cible: Le profil de sécurité du point d'extrémité ESF cible indique le niveau de sécurité attendu décrit dans la ressource <*e2ESecurityCapabilities*> associée au point d'extrémité ESF cible.

e) La fonction TEF fournit aux points d'extrémité ESF source et cible les paramètres détaillés d'extraction et d'extension de clés à utiliser lors du calcul des justificatifs d'identité de bout en bout à partir du justificatif d'identité Km ou Kpsa.

f) La fonction TEF fournit la portée et les paramètres de sécurité associés aux points d'extrémité ESF source et cible afin de déterminer les protocoles et les algorithmes cryptographiques à utiliser pour assurer la sécurité de bout en bout.

Prise de contact de sécurité d'amorçage: le point d'extrémité ESF source et la fonction TEF effectuent une prise de contact (D)TLS-PSK [IETF RFC 4279] afin d'établir une session sécurisée. Le mécanisme est composé des procédures décrites au § 8.3.2.

Génération de clé de bout en bout:

a) La clé d'inscription (Ke) et l'identificateur relativeKeID sont générés à l'aide des secrets de la session (D)TLS par le point d'extrémité ESF source et la fonction TEF grâce à l'exportation de clés par protocole TLS ([IETF RFC 5705]), comme cela est décrit au § 10.3.1 (Détails relatifs à l'exportation de clés TLS). De même, l'identificateur de la clé d'inscription (KeID) est généré à l'aide de l'identificateur relativeKeID et du nom de domaine complet (FQDN) de la fonction TEF par le point d'extrémité ESF source et la fonction TEF comme cela est décrit au § 10.3.4 (Génération de l'identificateur KeID). Le point d'extrémité ESF source et la fonction TEF stockent la clé Ke et l'identificateur KeID qui lui est associé.

b) La clé principale de bout en bout (Ke2e_master) et l'identificateur E2EKeyId sont générés de la même manière que le justificatif d'identité Kpsa et l'identificateur associé KpsaID. Si le point d'extrémité ESF source demande la fourniture de clés de bout en bout, des clés sont extraites en fonction des justificatifs d'identité Kpsa et Km.

c) La clé principale de bout en bout (Ke2e_master) est utilisée pour générer des clés de sécurité spécifiques, telles qu'une clé d'authentification de bout en bout, une clé de confidentialité de bout en bout et d'autres clés en fonction des paramètres d'extraction et d'extension renseignés. Les processus d'extraction et d'extension de clés reposent sur [IETF RFC 5869].

NOTE – Le processus de génération de clés de bout en bout du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite est identique au processus de génération de clés de bout en bout du cadre de configuration à distance de la sécurité fondé sur les certificats.

8.6.3 Description détaillée des justificatifs d'identité de bout en bout générés par la source

Ce paragraphe décrit la génération et l'enregistrement des justificatifs d'identité symétriques de sécurité de bout en bout. Les justificatifs d'identité de sécurité de bout en bout générés automatiquement par la source doivent être enregistrés auprès de la fonction TEF. Ce mécanisme est particulièrement utile quand des données (attributs) et des messages destinés à plusieurs cibles sont nécessaires. Pour sécuriser la ressource <*contentInstance*> utilisée par plusieurs entités finales, les justificatifs d'identité générés par la source doivent être utilisés.

Une source qui génère des données consommées par plusieurs entités finales peut générer les justificatifs d'identité appropriés de sorte que l'intégrité et la confidentialité d'un seul attribut (par exemple, la valeur de l'attribut de contenu d'une ressource <*contentInstance*> ou l'attribut *customAttribute* d'une ressource <*flexContainer*>) ou d'un seul élément adressable d'un attribut soient protégées par les cadres ESData et ESPrim. Dans le cas d'une autorisation dynamique, tout ou partie de la valeur d'un seul paramètre de primitive (par exemple, un jeton d'accès autonome et signé transmis dans une primitive de demande afin d'obtenir une autorisation dynamique) peut être également protégé à l'aide des cadres ESData et ESPrim. L'entité que génère le cadre ESData/ESPrim enregistre les justificatifs d'identité à l'aide d'une fonction TEF.

La Figure 8.6.3-1 illustre de manière schématique l'utilisation d'une fonction TEF en vue de distribuer les justificatifs d'identité générés par la source.

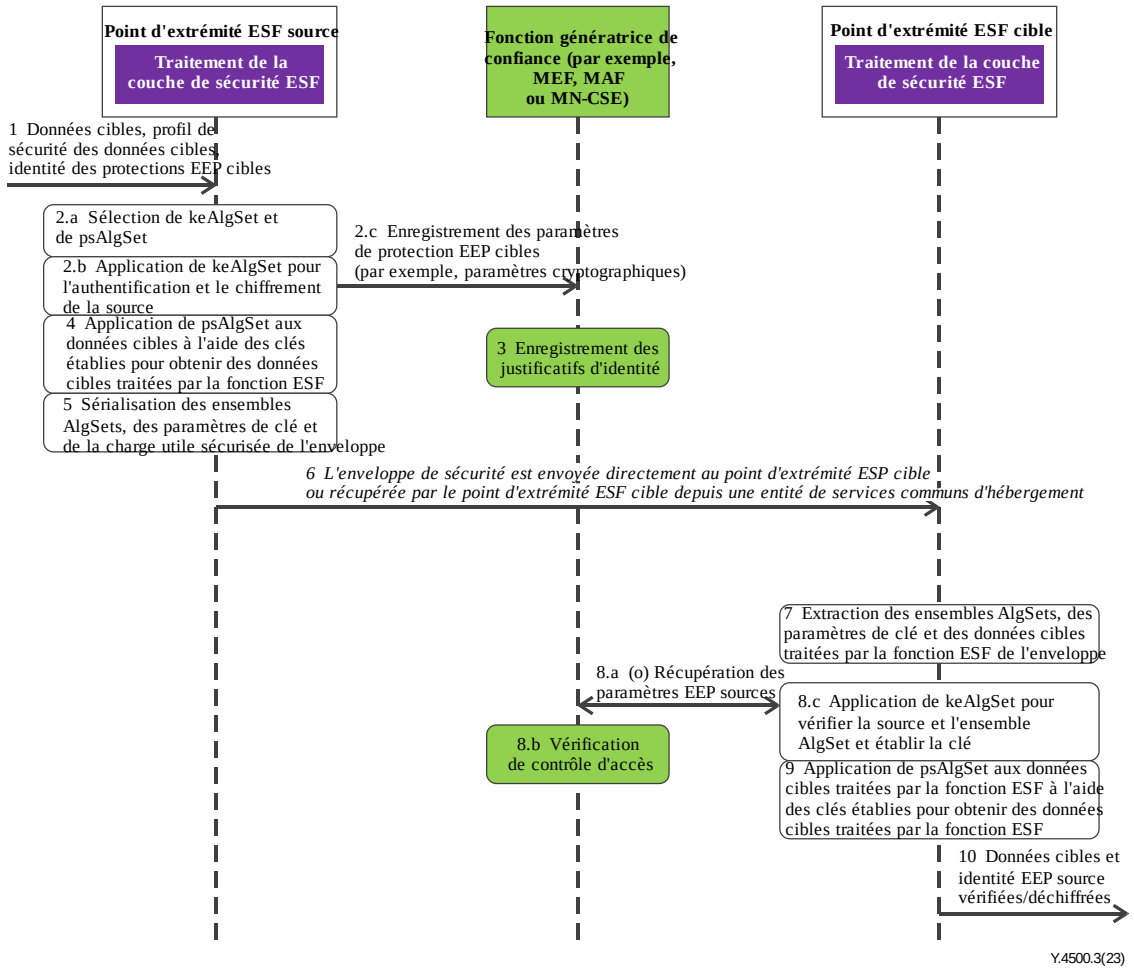


Figure 8.6.3-1 – Schéma d'utilisation d'une fonction TEF en vue de distribuer les justificatifs d'identité générés par la source

Configuration des justificatifs d'amorçage: on suppose que le point d'extrémité ESF source est configuré avec les éléments Ke et KeID qui ont été générés par le cadre de configuration à distance à l'aide d'une fonction TEF (par exemple, une fonction d'inscription M2M), comme cela est décrit au § 8.2. Le point d'extrémité ESF source peut être configuré avec les éléments Km et KmID qui ont été générés par le cadre de configuration à distance à l'aide d'une fonction TEF, comme cela est décrit au § 8.3. Le point d'extrémité ESF cible peut être fourni avec les éléments Ke et KeID si la fonction TEF est une fonction d'inscription M2M, ou avec les éléments Km et KmID si l'association de sécurité est établie à l'aide d'une fonction d'authentification M2M.

Configuration des instructions d'amorçage: le point d'extrémité ESF source et les points d'extrémité ESF cibles sont fournis avec l'URI de la fonction TEF qui prend en charge la fourniture et l'enregistrement des justificatifs d'identité de sécurité de bout en bout.

Le point d'extrémité ESF source est configuré avec les arguments suivants afin de démarrer la configuration à distance:

a) L'identité du point d'extrémité ESF cible qui doit être fourni avec le cadre ESData/ESPrim ainsi qu'avec les justificatifs d'identité de sécurité de bout en bout associés.

b) Les exigences de sécurité associées aux données (attributs): elles sont préconfigurées et fournies par l'application. En fonction des mécanismes de sécurité, les technologies de sécurité appropriées doivent être utilisées.

c) Argument préconfiguré qui comprend un tableau répertoriant les exigences de sécurité et la façon dont elles peuvent être respectées grâce aux mécanismes de sécurité adaptés (par exemple, des protocoles de sécurité ou des algorithmes de vérification de l'intégralité et de la confidentialité et de génération de clés).

d) Le profil de sécurité du point d'extrémité ESF cible (facultatif): le profil de sécurité du point d'extrémité ESF cible et les fonctionnalités de sécurité associées à celui-ci et décrites dans la ressource <*e2ESecurityCapabilities*> peuvent être utilisés pour identifier les types de mécanismes de sécurité pouvant servir à fournir le cadre ESData/ESPrim.

La fonction TEF est configurée avec les arguments suivants pour enregistrer les justificatifs d'identité de sécurité ESData/ESPrim du point d'extrémité ESF source et être autorisée à fournir les paramètres cryptographiques de sécurité ESData/ESPrim pertinents uniquement à un ensemble de points d'extrémité ESF cibles:

a) Paramètres cryptographiques: une liste de paramètres cryptographiques de bout en bout identifiés par un identificateur de justificatif d'identité et auxquels sont associées des valeurs cryptographiques, telles que des justificatifs d'identité, des algorithmes cryptographiques, des libellés et des valeurs aléatoires (par exemple, un nonce ou un vecteur d'initialisation) Ces paramètres sont fournis par le point d'extrémité ESF source durant l'enregistrement des justificatifs d'identité. Des justificatifs d'identité peuvent être associés à un ou plusieurs mécanismes de sécurité (qui assurent par exemple l'intégrité et la confidentialité des données). La liste peut également inclure la portée et l'utilisation des paramètres de sécurité de bout en bout afin que le point d'extrémité ESF cible puisse traiter les éléments ESData/ESPrim (par exemple, en vérifiant leur intégrité ou en les déchiffrant).

b) Identité du point d'extrémité ESF cible: elle doit être fournie avec les justificatifs d'identité demandés auquel est associé un identificateur. L'autorisation peut être effectuée et appliquée au moyen de politiques de contrôle d'accès (ACP).

Le point d'extrémité ESF cible est configuré avec les arguments suivants:

a) ESData/ESPrim: le point d'extrémité ESF cible reçoit les éléments ESData/ESPrim directement depuis un point d'extrémité ESF source, ou il les récupère depuis une entité d'hébergement (par exemple, une entité de services communs hôte).

b) Identificateur de justificatif d'identité: il est fourni au point d'extrémité ESF cible et peut faire partie des éléments ESData/ESPrim.

c) Paramètres cryptographiques: la fonction TEF les fournit après avoir vérifié les politiques de contrôle d'accès associées à la demande émise par le point d'extrémité ESF source.

Prise de contact de sécurité:

a) le point d'extrémité ESF source et la fonction TEF effectuent une prise de contact (D)TLS [IETF RFC 4279] afin d'établir une session sécurisée. Le mécanisme est composé des procédures décrites au § 8.3.2. Toutes les communications entre le point d'extrémité ESF source et la fonction TEF sont sécurisées à l'aide de la connexion (D)TLS établie.

b) le point d'extrémité ESF cible et la fonction TEF effectuent une prise de contact (D)TLS [IETF RFC 4279] afin d'établir une session sécurisée. Le mécanisme est composé des procédures décrites au § 8.3.2. Toutes les communications entre le point d'extrémité ESF source et la fonction TEF sont sécurisées à l'aide de la connexion (D)TLS établie.

Génération de clé de bout en bout:

a) Le point d'extrémité ESF source génère des justificatifs d'identité pouvant dépendre des éléments suivants:

- Les justificatifs d'identité générés à l'aide de la clé d'inscription et les éléments Ke et KeID produits durant le processus amorcé de configuration à distance des justificatifs d'identité;
- Les justificatifs d'identité générés aléatoirement par le point d'extrémité ESF source et enregistrés auprès de la fonction TEF.

8.7 Établissement d'une clé à l'aide d'un certificat de sécurité de bout en bout (ESCertKE)

8.7.1 Finalité du cadre ESCertKE

L'établissement de clés fondé sur les certificats de bout en bout (ESCertKE) constitue un cadre interopérable permettant à deux points d'extrémité d'établir, à l'aide de certificats, une clé symétrique secrète appelée *pairwiseE2EKey*, à partir de laquelle des clés symétriques sont calculées. Ces dernières sont ensuite utilisées dans d'autres cadres de sécurité de bout en bout, comme celui portant sur les données (ESData) ou celui qui a trait aux primitives (ESPrim).

Les cas d'utilisation et les exigences correspondants sont exposés dans le document oneM2M TR-0012 [b-oneM2M TR0012].

La présente Recommandation décrit les messages ESCertKE et la procédure de traitement associée. La transmission des messages ESCertKE est exposée dans la Recommandation [ITU-T Y.4500.1].

8.7.2 Architecture du cadre ESCertKE

8.7.2.1 Modèle de référence du cadre ESCertKE

Les entités du modèle de référence du cadre ESCertKE sont l'*extrémité initiatrice ESCertKE*, qui démarre la procédure, et l'unique *extrémité de destination ESCertKE*, avec laquelle l'extrémité initiatrice ESCertKE doit créer une clé pairwiseE2EKey.

NOTE – Tout au long du § 8.7, le mot "ESCertKE" qui compose certains termes peut en être retiré afin d'en faciliter la lisibilité. Par exemple, le terme "extrémité initiatrice ESCertKE" est souvent abrégé, donnant ainsi simplement "extrémité initiatrice".

Le cadre *ESCertKE* consiste en l'échange de *messages ESCertKE* entre une extrémité initiatrice et une extrémité de destination et comprend une phase de traitement fondée sur ces messages. Si les opérations du cadre ESCertKE réussissent, l'extrémité initiatrice et l'extrémité de destination exportent une clé pairwiseE2EKey en fonction des paramètres échangés par le biais des messages ESCertKE.

Il n'existe aucune restriction inhérente quant aux entités qui peuvent être des extrémités initiatrices. Il peut s'agir d'entités d'un système oneM2M (c'est-à-dire des entités d'application et des entités de services communs) ou d'entités se trouvant en dehors d'un tel système (par exemple si elles font partie d'un système interopérable avec l'architecture oneM2M).

En revanche, il existe une seule restriction quant aux entités qui peuvent être des extrémités de destination. En effet, celles-ci doivent pouvoir recevoir le message ESCertKE non sollicité qui démarre le cadre ESCertKE. En plus de décrire la transmission des messages ESCertKE, la Recommandation [ITU-T Y.4500.1] précise quelles entités peuvent être des extrémités de destination.

8.7.2.2 Flux de messages du cadre ESCertKE

Les messages ESCertKE sont transmis selon les spécifications de la Recommandation [ITU-T Y.4500.1]. Par exemple, la ressource <*e2EKeyCSE*> peut être utilisée.

Les messages ESCertKE doivent contenir les messages TLS v1.2 [IETF RFC 5246] définis dans le Tableau 8.7.2.2-1 (Définitions des messages ESCertKE).

Le flux de messages du cadre ESCertKE est illustré dans la Figure 8.7.2.2-1 ci-dessous.

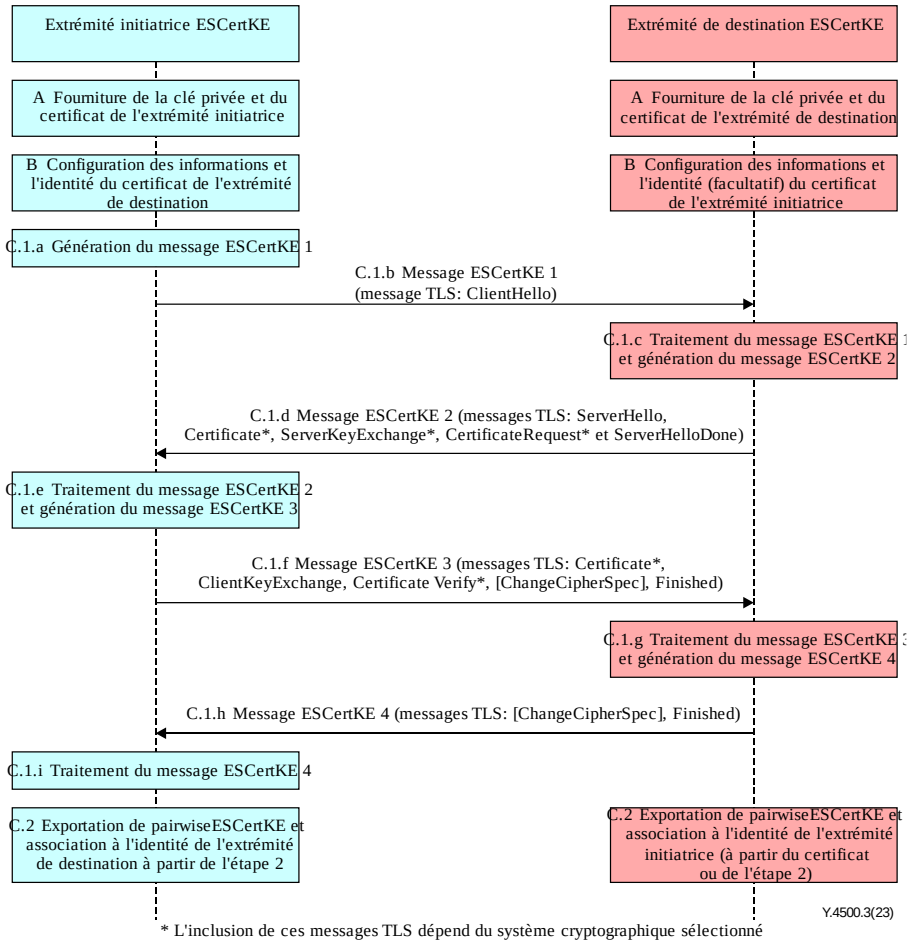


Figure 8.7.2.2-1 – Flux de messages du cadre ESCertKE

A **Fourniture des certificats**: Les extrémités ESCertKE doivent recevoir une clé privée et les certificats décrits au § 8.1.2.3 (Configuration des justificatifs d'identité pour le cadre de sécurité fondé sur les certificats). Les certificats en question doivent être conformes aux spécifications exposées au § 10.1 (Détails concernant le cadre de sécurité fondé sur les certificats).

B **Déclenchement**: les extrémités initiatrices et de destination doivent être configurées à l'aide des informations nécessaires à leur authentification et leur identification:

L'extrémité initiatrice démarre le cadre ESCertKE. La commande permettant cela doit inclure les arguments suivants:

Les informations du certificat de l'extrémité de destination décrites au § 8.1.2.4 (Informations requises pour l'authentification par certificat d'une autre entité).

L'identité de l'extrémité de destination. Cette information est utilisée aux fins suivantes:

– identification de la destination du message ESCertKE 1; et

– association à la clé pairwiseE2EKey établie.

L'extrémité de destination doit être configurée à l'aide des arguments suivants, qui décrivent l'entité initiatrice autorisée à exécuter la procédure:

Les informations du certificat de l'extrémité initiatrice décrites au § 8.1.2.4 (Informations requises pour l'authentification par certificat d'une autre entité).

Si le certificat de l'extrémité initiatrice est un certificat de clé publique brute, l'extrémité de destination doit également être configurée avec une identité à associer à la clé pairwiseE2EKey établie.

Les points d'extrémité peuvent être configurés dans n'importe quel ordre.

C Établissement de la clé *pairwiseE2EKey*:

C.1 Les extrémités initiatrices et de destination échangent une séquence de quatre messages ESCertKE. Les messages ESCertKE doivent être générés et traités conformément au protocole de prise de contact TLS v1.2 [IETF RFC 5246]. Les systèmes cryptographiques TLS utilisés pour le cadre ESCertKE doivent être conformes aux indications du § 10.2.3 (Systèmes cryptographiques des protocoles TLS et DTLS destinés aux cadres de sécurité fondés sur les certificats).

C.1.a L'extrémité initiatrice génère le message ESCertKE 1.

C.1.b L'extrémité initiatrice envoie le message ESCertKE 1 à l'extrémité de destination identifiée à l'étape 2.

C.1.c L'extrémité de destination traite le message ESCertKE 1 et génère le message ESCertKE 2.

C.1.d L'extrémité de destination envoie le message ESCertKE 2 à l'extrémité initiatrice.

C.1.e L'extrémité initiatrice traite le message ESCertKE 2 et génère le message ESCertKE 3.

C.1.f L'extrémité initiatrice envoie le message ESCertKE 3 à l'extrémité de destination.

C.1.g L'extrémité de destination traite le message ESCertKE 3 et génère le message ESCertKE 4.

C.1.h L'extrémité de destination envoie le message ESCertKE 4 à l'extrémité initiatrice.

C.1.i L'extrémité initiatrice traite le message ESCertKE 4.

C.2 Si le protocole de prise de contact TLS réussit, les extrémités initiatrices et de destination exportent et mettent en cache la clé pairwiseE2EKey selon les spécifications de l'exportateur TLS [IETF RFC 5705] exposées au § 10.3.1 (Détails concernant l'exportation de clés à l'aide du protocole TLS).

Tableau 8.7.2.2-1 – Définitions des messages ESCertKE

Message ESCertKE	Extrémité d'envoi	Messages TLS v1.2 possibles (cas de réussite) [IETF RFC 5246]	Description informative (la description normative est disponible dans la spécification TLS v1.2 [IETF RFC 5246])
1	Initiatrice	ClientHello	Liste des systèmes cryptographiques autorisés, valeur aléatoire et indicateur d'exportation de la clé pairwiseE2EKey.
2	Destination	ServerHello	Système cryptographique sélectionné, valeur aléatoire et indicateur d'exportation de la clé pairwiseE2EKey.
		Certificate*	Certificat de l'extrémité de destination (et éventuellement une chaîne de certificats).

Tableau 8.7.2.2-1 – Définitions des messages ESCertKE

Message ESCertKE	Extrémité d'envoi	Messages TLS v1.2 possibles (cas de réussite) [IETF RFC 5246]	Description informative (la description normative est disponible dans la spécification TLS v1.2 [IETF RFC 5246])
		ServerKeyExchange*	Paramètres d'échange de clés générés par l'extrémité de destination. Le contenu de ce paramètre dépend du système cryptographique sélectionné.
		CertificateRequest*	Ordonne à l'extrémité initiatrice de s'authentifier à l'aide d'un certificat.
		ServerHelloDone	Indique la fin du message.
3	Initiatrice	Certificate*	Certificat de l'extrémité initiatrice (et éventuellement une chaîne de certificats).
		ClientKeyExchange*	Paramètres d'échange de clés générés par l'extrémité initiatrice. Le contenu de ce paramètre dépend du système cryptographique sélectionné.
		CertificateVerify	Vérifie de manière explicite le certificat de l'extrémité initiatrice.
		[ChangeCipherSpec]	Informe l'extrémité de destination que les enregistrements consécutifs seront protégés conformément à la nouvelle spécification cryptographique CipherSpec et aux clés choisies.
		Finished	Code MIC de tous les paramètres précédemment échangés durant la procédure. Le code MIC est généré à l'aide de secrets de session créés grâce aux paramètres précédents.
4	Destination	[ChangeCipherSpec]	Voir ci-dessus.
		Finished	Code MIC de tous les paramètres précédemment échangés durant la procédure. Le code MIC est généré à l'aide de secrets de session.
NOTE – L'inclusion des messages TLS marqués par un astérisque ("*") dépend du système cryptographique choisi.			

8.8 Détails concernant le cadre de sécurité MAF

8.8.1 Introduction

Le paragraphe 8.8 expose les détails et les procédures des cadres de sécurité fondés sur la fonction MAF. Dans la présente spécification, ces cadres sont les suivants:

– le cadre d'établissement d'association de sécurité (SAEF) fondé sur la fonction MAF;

– le cadre de sécurité de bout en bout des primitives (ESPrim) fondé sur la fonction MAF;

– le cadre de sécurité de bout en bout des données (ESData) fondé sur la fonction MAF.

Ces cadres utilisent une fonction MAF pour authentifier et distribuer une clé symétrique dont un point d'extrémité source se sert pour démarrer l'établissement de la clé symétrique et d'au moins un point d'extrémité cible. Le Tableau 8.8.1-1 décrit la correspondance des rôles des cadres MAF génériques avec les rôles des cadres MAF spécifiques. Les clients MAF peuvent récupérer la clé symétrique produite à partir de la fonction MAF. La fonction MAF fournit ses services pour le compte des *parties prenantes administratrices*, telles que les fournisseurs de services M2M (M2M-SP) ou les éléments de confiance M2M tiers (MTE). Une partie prenante administratrice autorise la fonction MAF à fournir des services aux clients MAF et supervise l'autorisation de la distribution des clés symétriques. Le Tableau 8.8.1-1 décrit la correspondance des clients MAF source et cible avec les rôles utilisés dans les cadres MAF génériques ainsi que le nombre de clients MAF cibles autorisés.

Tableau 8.8.1-1 – Correspondance aux cadres de sécurité MAF spécifiques

Cadre de sécurité MAF	Client MAF source	Client MAF cible	Nombre de clients MAF cibles	Clé symétrique produite
Cadre d'établissement d'association de sécurité (SAEF)	Entité A	Entité B	1	Clé de connexion sécurisée M2M (Kc)
Sécurité de bout en bout des primitives (ESPrim)	Expéditeur	Destinataire	1	pairwiseESPrimKey
Sécurité de bout en bout des données (ESData)	Point d'extrémité ES Data source	Point d'extrémité ES Data cible	1..n	Clé ESData

Le paragraphe 8.8 décrit les *procédures MAF* s'exécutant entre les clients MAF et les messages associés. Les détails relatifs à l'exécution et à la gestion de la fonction MAF, au-delà des détails exposés à propos des procédures MAF, ne sont pas précisés dans la présente Recommandation.

La séquence générale d'utilisation des procédures MAF est illustrée dans la Figure 8.8.1-1 et peut être décrite comme suit:

Chaque client MAF doit créer séparément des justificatifs d'identité afin de s'authentifier mutuellement à l'aide de la fonction MAF, comme cela est décrit au § 8.8.3.1 relatif à la **configuration des justificatifs d'identité des clients MAF**.

Chaque client MAF doit être configuré séparément pour être enregistré sur la fonction MAF avec une partie prenante administratrice. Le paragraphe 8.8.3.2 (**Détails concernant la configuration de l'enregistrement du client MAF**) précise les paramètres nécessaires.

Chaque client MAF effectue une **procédure d'enregistrement des clients MAF** à l'aide de la fonction MAF. Cela permet de confirmer que le client MAF souhaite utiliser les services de la fonction MAF avec l'autorisation de la partie prenante administratrice. Le client MAF doit s'enregistrer séparément pour chaque partie prenante administratrice, même quand il le fait à l'aide d'une seule fonction MAF. Si le client MAF est configuré à distance pour authentification mutuelle avec la fonction MAF, cette dernière fournit le client MAF à l'aide de l'identificateur KmID, qui doit être utilisé pour une authentification consécutive avec la fonction MAF.

À un moment ultérieur ne s'inscrivant pas dans cette séquence d'événements, la **procédure de mise à jour de l'enregistrement des clients MAF** peut s'exécuter afin de confirmer que le client MAF va utiliser les services de la fonction MAF ou créer un nouveau justificatif d'identité Km et son identificateur KmID. De plus, la **procédure d'annulation de l'enregistrement des clients MAF** peut être effectuée pour indiquer que le client MAF cesse d'utiliser les services de la fonction MAF.

Le client MAF source doit être configuré pour établir une communication sécurisée à l'aide d'un cadre de sécurité (SAEF, ESPrim, ESData) et de clés symétriques établies au moyen de la fonction MAF. Les détails relatifs à cette configuration sont propres au cadre de sécurité utilisé, mais doivent inclure la **configuration d'enregistrement de clés MAF** (§ 8.4.4.3).

Le client MAF source effectue une **procédure d'enregistrement de clés MAF** pour établir une clé symétrique et un identificateur correspondant. Le client MAF source fournit également un identificateur d'utilisation de sécurité (SUID) qui limite la portée du justificatif d'identité en identificateur le cadre de sécurité (SAEF, ESPrim, ESData). Cette procédure doit comprendre une **prise de contact MAF** afin d'assurer l'authentification mutuelle du client MAF source et de la fonction MAF.

À un moment ultérieur ne s'inscrivant pas dans cette séquence d'événements, la **procédure de mise à jour de l'enregistrement des clés MAF** peut s'exécuter afin de mettre à jour la date d'expiration de la clé enregistrée ou mettre à jour la liste des clients MAF cibles. De plus, la **procédure d'annulation de l'enregistrement des clés MAF** peut être effectuée pour supprimer l'enregistrement de la clé de la fonction MAF.

Le client MAF source fournit aux clients MAF cibles l'identificateur de la clé symétrique établie durant la procédure d'enregistrement de clés MAF. Les détails de cette étape dépendent du cadre de sécurité identifié par l'identificateur SUID.

Le client MAF cible exécute la **procédure de récupération de clés MAF** pour récupérer la clé symétrique et les informations correspondantes. Cette procédure doit comprendre une **prise de contact MAF** afin d'assurer l'authentification mutuelle du client MAF cible et de la fonction MAF.

La clé symétrique doit être utilisée durant le protocole de sécurité qui s'exécute entre le client MAF source et le client MAF cible. Si le protocole de sécurité requiert une seule clé symétrique, on utilisera la première moitié de la clé symétrique distribuée. Si le protocole de sécurité requiert deux clés symétriques (par exemple, une clé de chiffrement et une clé d'intégrité distincte), on utilisera les deux moitiés de la clé symétrique distribuée, chacune constituant l'une des deux clés symétriques requises par le protocole de sécurité. Les détails de cette étape dépendent du cadre de sécurité.

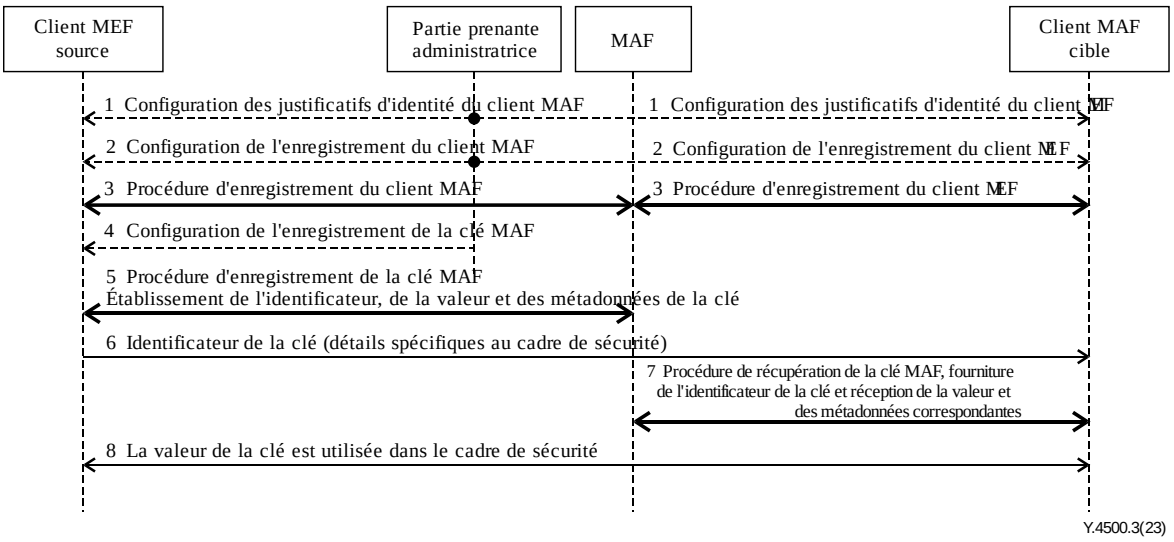


Figure 8.8.1-1 – Séquence d'événements liés à l'utilisation du cadre de sécurité MAF en tant que partie d'une fonctionnalité de sécurité

Le paragraphe 8.8 est organisé comme suit. Le paragraphe 8.8.2 décrit la phase de traitement et les flux d'information des procédures MAF. Le paragraphe 8.8.3 décrit les informations relatives à la configuration des justificatifs d'identité et de l'enregistrement du client MAF ainsi qu'à la configuration de l'enregistrement de la clé MAF.

8.8.2 Procédure de traitement et flux d'informations du cadre de sécurité MAF

8.8.2.1 Introduction

Le paragraphe 8.8.2 décrit la phase de traitement et les flux d'information des procédures MAF.

8.8.2.2 Procédure de prise de contact MAF

Finalité: une procédure de prise de contact MAF crée une session TLS ou DTLS mutuellement authentifiée pour protéger la communication entre un client MAF et une fonction MAF. Dans le cas de la procédure d'enregistrement de clé MAF, le client MAF source et la fonction MAF peuvent utiliser la session TLS ou DTLS pour établir la valeur de la clé.

Conditions préalables: l'une des conditions suivantes doit être respectée:

Le client MAF et la fonction MAF se voient remettre des certificats, comme cela est décrit dans les détails relatifs à la configuration des justificatifs d'identité du client MAF exposés au § 8.8.3.1. De plus, ils sont configurés avec des certificats de CA pour valider les certificats conformément aux détails de configuration de l'enregistrement du client MAF exposés au § 8.8.3.2.

Le client MAF et la fonction MAF créent un justificatif d'identité symétrique principal (Km) et son identificateur correspondant (KmID). Le justificatif d'identité Km et l'identificateur KmID peuvent être préconfigurés. Autrement, Km peut être créé à l'aide d'un cadre de configuration à distance de la sécurité et de l'identificateur KmID créé durant la procédure d'enregistrement du client MAF.

NOTE – Dans le cas de la création du justificatif d'identité Km par configuration à distance, la prise de contact MAF ne peut pas être effectuée pendant l'enregistrement du client MAF car (a) la fonction MAF ne connaît pas le justificatif d'identité Km avant l'enregistrement du client MAF; et (b) l'identificateur KmID n'a pas été attribué avant l'enregistrement du client MAF.

Description de la procédure

Si le client MAF et la fonction MAF ont créé un justificatif d'identité symétrique principal (Km) avec l'identificateur correspondant (KmID), le client MAF et la fonction MAF doivent établir la session TLS ou DTLS au moyen d'une prise de contact TLS-PSK conformément au § 10.2.2 et en donnant les précisions suivantes:

La valeur du paramètre "psk_identity" [IETF RFC 4279] doit être égale à la valeur de l'identificateur du justificatif d'identité principal (KmID).

La valeur du paramètre "psk" [IETF RFC 4279] doit être égale à la valeur du justificatif d'identité principal (Km).

Si le client MAF et la fonction MAF doivent s'authentifier à l'aide de certificats, ils doivent établir la session TLS ou DTLS au moyen d'une prise de contact TLS fondée sur les certificats conformément au § 10.2.2 et avec les précisions suivantes:

Le certificat du serveur TLS doit être le certificat de la fonction MAF. Le client MAF vérifie le certificat de la fonction MAF en le comparant à l'ensemble de ses ancres de confiance des certificats MAF fournis, conformément au § 8.1.2.5.

Le certificat du client TLS doit être le certificat du client MAF. La fonction MAF vérifie le certificat du client MAF en le comparant aux informations du certificat du client MAF fourni, conformément au § 8.1.2.5.

8.8.2.3 Procédure d'enregistrement du client MAF

Finalité: le client MAF s'enregistre auprès de la fonction MAF pour confirmer qu'il est disposé à utiliser les services de cette dernière avec l'autorisation de la partie prenante administratrice. Si la procédure de configuration à distance est utilisée pour établir une clé symétrique entre un client MAF et la fonction MAF, le client MAF exécute la fonction MAF (durant la procédure de prise de contact TLS) pour récupérer le justificatif d'identité Km depuis la fonction MEF, et la fonction MAF

fournit au client MAF l'identificateur du justificatif d'identité principal (KmID) à utiliser dans les procédures ultérieures de prise de contact MAF.

NOTE – La procédure d'enregistrement du client MAF est équivalente celle de l'entité de services communs ou de l'entité d'application, à ceci près que le client MAF s'enregistre auprès de la fonction MAF et non pas auprès de l'entité de services communs d'enregistrement.

Conditions préalables: les paramètres décrits aux paragraphes 8.8.3.1 et 8.8.3.2 sont fournis au client MAF, à la fonction MAF ainsi qu'à la fonction MEF (le cas échéant).

Description de la procédure

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF.

Si la procédure de configuration à distance est utilisée, les étapes (b) et suivantes de la section relative à l'utilisation des justificatifs d'identité configurés du § 8.3.2.1 doivent être effectuées par le client MAF (agissant en tant qu'entité inscrite), la fonction MEF et la fonction MAF (agissant en tant que cible d'enregistrement). L'identificateur SUID de la clé fournie à distance doit être "21" (clé symétrique fournie par un cadre de configuration à distance de la sécurité [RSPF] et destinée à être partagée avec une fonction MAF, comme cela est décrit dans la Recommandation [UIT-T Y.4500.4]). La fonction MAF récupère le justificatif d'identité Km de la fonction MEF au cours de cette procédure.

Sinon, le client MAF et la fonction MAF effectuent les procédures de prise de contact MAF (§ 8.8.2.2).

Elles fournissent à la fonction MAF une identité authentifiée pour le client MAF.

Le client MAF envoie une demande d'enregistrement de client MAF comprenant les informations indiquées dans le Tableau 8.8.2.3-1.

Tableau 8.8.2.3-1 – Informations du message de demande d'enregistrement du client MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
expirationTime	Date et heure d'expiration proposées de l'enregistrement.	1
labels	Libellé facilitant la découverte du relevé de l'enregistrement du client MAF.	0 ou 1
adminFQDN	Nom de domaine complet de la partie prenante administratrice fourni durant la configuration de l'enregistrement du client MAF.	1

Dès la réception de la demande, la fonction MAF traite cette dernière. En cas d'erreur, la fonction MAF envoie un message d'erreur. La fonction MAF peut attribuer différentes valeurs aux paramètres envoyés par le client MAF suivant les instructions de la partie prenante administratrice. Si la demande est correctement traitée, la fonction MAF compose une demande de réponse d'enregistrement du client MAF comprenant les informations indiquées dans le Tableau 8.8.2.3-2.

Tableau 8.8.2.3-2 – Informations du message de réponse d'enregistrement du client MAF

Paramètre	Description	Multiplicité
mafClientRegID	Identificateur du nouvel enregistrement du client MAF.	1
labels	Libellé facilitant la découverte du relevé de l'enregistrement du client MAF.	0 ou 1
expirationTime	Date et heure d'expiration de l'enregistrement du client MAF.	1

Tableau 8.8.2.3-2 – Informations du message de réponse d'enregistrement du client MAF

Paramètre	Description	Multiplicité
MAF Client ID	Identificateur du client MAF.	1
adminFQDN	Nom de domaine complet de la partie prenante administratrice.	1
assignedSymmKeyID	Identificateur du justificatif d'identité principal attribué par la fonction MAF (KmID) dans le cas où Km est configuré à distance.	0 ou 1

La fonction MAF envoie la réponse au client MAF.

Le client MAF et la fonction MAF stockent les paramètres. Si le paramètre assignedSymmKeyID a été inclus, le client MAF l'utilise en tant qu'identificateur de justificatif d'identité principal (KmID) lors de la création des sessions TLS (ou DTLS) avec la fonction MAF.

8.8.2.4 Procédure de récupération de la configuration du client MAF

Finalité: cette procédure permet à un client MAF de récupérer ses configurations, lesquelles sont fournies à la fonction MAF par la partie prenante administratrice.

Conditions préalables

Le client MAF a exécuté au préalable la procédure d'enregistrement de client MAF en vue de créer son relevé d'enregistrement.

Le relevé d'enregistrement du client MAF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF comme cela est décrit à l'étape 1 du § 8.8.2.3.

Le client MAF envoie une demande de récupération de sa configuration comprenant les informations indiquées dans le Tableau 8.8.2.4-1.

Tableau 8.8.2.4-1 – Informations du message de demande de récupération de la configuration du client MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
mafClientRegID	Identificateur du relevé d'enregistrement du client MAF qui est mis à jour.	1

Dès la réception de la demande, la fonction MAF traite cette dernière. En cas d'erreur, y compris si aucune configuration de client MAF n'est actuellement associée à l'enregistrement de client MAF identifié, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF essaie de récupérer la configuration de client MAF actuellement associée au relevé d'enregistrement de client MAF identifié.

La fonction MAF compose une réponse de récupération de la configuration du client MAF contenant les paramètres suivants.

Tableau 8.8.2.4-2 – Informations du message de réponse de récupération de la configuration du client MAF

Paramètre	Description	Multiplicité
mafClientCfg	Configuration de client MAF actuellement associée au relevé d'enregistrement de client MAF identifié.	1

La fonction MAF envoie la réponse au client MAF.

Le client MAF applique sa configuration.

8.8.2.5 Procédure de mise à jour de l'enregistrement du client MAF

Finalité: cette procédure permet à un client MAF de mettre à jour l'enregistrement du client MAF en repoussant sa date d'expiration (paramètre *expirationTime)*, en mettant à jour le paramètre *labels* ou en créant un nouveau justificatif d'identité Km et un nouvel identificateur KmID.

Conditions préalables

Le client MAF a exécuté au préalable la procédure d'enregistrement de client MAF en vue de créer son relevé d'enregistrement.

Le relevé d'enregistrement du client MAF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF comme cela est décrit à l'étape 1 du § 8.8.2.3.

Le client MAF envoie une demande de mise à jour de l'enregistrement de client MAF comprenant les informations indiquées dans le Tableau 8.8.2.5-1.

Tableau 8.8.2.5-1 – Informations du message de demande de mise à jour de l'enregistrement du client MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
mafClientRegID	Identificateur du relevé d'enregistrement du client MAF qui est mis à jour.	1
expirationTime	Date et heure d'expiration proposées du relevé d'enregistrement du client MAF.	0 ou 1
labels	Libellé facilitant la découverte du relevé de l'enregistrement du client MAF.	0 ou 1
NOTE – Au moins l'un des paramètres *expirationTime* et *labels* doit être inclus.		

Dès la réception de la demande, la fonction MAF traite cette dernière. En cas d'erreur, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF met à jour le relevé d'enregistrement du client MAF avec les valeurs proposées si la partie prenante administratrice l'autorise. La fonction MAF peut attribuer différentes valeurs aux paramètres envoyés par le client MAF suivant les instructions de la partie prenante administratrice.

La fonction MAF compose une réponse de mise à jour de l'enregistrement du client MAF contenant les paramètres indiqués dans le Tableau 8.8.2.5-2.

Tableau 8.8.2.5-2 – Informations du message de réponse de mise à jour de l'enregistrement du client MAF

Paramètre	Description	Multiplicité
expirationTime	Date et heure d'expiration mises à jour de l'enregistrement du client MAF.	0 ou 1
labels	Libellé mis à jour facilitant la découverte du relevé de l'enregistrement du client MAF.	0 ou 1
assignedSymmKeyID	Identificateur du justificatif d'identité principal attribué par la fonction MAF (KmID) dans le cas où un nouveau justificatif d'identité Km est configuré à distance.	0 ou 1
NOTE – La réponse comprend uniquement les paramètres *expirationTime* et/ou *labels* s'ils sont présents dans la demande correspondante.		

La fonction MAF envoie la réponse au client MAF.

Le client MAF et la fonction MAF stockent les paramètres. Si le paramètre *assignedSymmKeyID* a été inclus, le client MAF l'utilise en tant qu'identificateur de justificatif d'identité principal lors de la création des sessions TLS (ou DTLS) avec la fonction MAF.

8.8.2.6 Procédure d'annulation de l'enregistrement du client MAF

Finalité: cette procédure permet à un client MAF de mettre fin à son enregistrement auprès de la fonction MAF.

Conditions préalables: le client MAF a exécuté au préalable la procédure d'enregistrement de client MAF en vue de créer son relevé d'enregistrement.

Le relevé d'enregistrement du client MAF n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF comme cela est décrit à l'étape 1 du § 8.8.2.3.

Le client MAF envoie une demande d'annulation de l'enregistrement du client MAF comprenant les informations indiquées dans le Tableau 8.8.2.6-1.

Tableau 8.8.2.6-1 – Informations du message de demande d'annulation de l'enregistrement du client MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
mafClientRegID	Identificateur du relevé d'enregistrement du client MAF qui est supprimé.	1

Dès la réception de la demande, la fonction MAF traite cette dernière. En cas d'erreur, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF supprime les informations associées au relevé d'enregistrement de client MAF identifié.

La fonction MAF compose une réponse de mise à jour de l'enregistrement du client MAF indiquant la réussite de l'opération. La fonction MAF envoie la réponse au client MAF.

8.8.2.7 Procédure d'enregistrement de la clé MAF

Finalité: cette procédure permet à un client MAF source d'établir, à l'aide de la fonction MAF, une clé symétrique qui peut être récupérée en vue d'être utilisée par un ou plusieurs clients MAF cibles.

Cette procédure s'effectue entre le client MAF source et la fonction MAF.

Conditions préalables: les informations de configuration de l'enregistrement de la clé MAF (§ 8.8.3.3) sont fournies au client MAF source (ou bien ce dernier les détermine).

Le client MAF source a effectué la procédure d'enregistrement de client MAF (§ 8.8.2.3) auprès de la fonction MAF pour la partie prenante administratrice identifiée dans la configuration d'enregistrement de la clé MAF.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF source établit une session TLS (ou DTLS) avec la fonction MAF au moyen d'une prise de contact MAF, comme cela est décrit au § 8.8.2.2. Un des résultats de la prise de contact MAF est que la fonction MAF crée une identité authentifiée pour le client MAF source.

Le client MAF source sélectionne la valeur de la clé de connexion sécurisée M2M (Kc) que la fonction MAF distribuera. La valeur doit être l'une des suivantes:

Le client MAF source génère la valeur de la clé symétrique de sortie à partir des secrets de la session (D)TLS grâce à l'exportation de clés TLS [IETF RFC 5705], comme cela est décrit au § 10.3.1 (Détails relatifs à l'exportation de clés TLS).

La valeur de la clé symétrique de sortie est générée automatiquement par le client MAF source, indépendamment des secrets de la session (D)TLS.

Le client MAF source dresse une liste des clients MAF cibles auxquels la fonction MAF est autorisée à fournir la valeur de la clé symétrique de sortie:

Dans le cas d'un cadre SAEF ou ESPrim fondé sur la fonction MAF: la liste doit contenir exactement un identificateur AE-ID absolu ou CSE-ID absolu.

Dans le cas d'un cadre ESData fondé sur la fonction MAF: la liste doit contenir n'importe quel nombre non nul d'identificateurs AE-ID absolus ou d'identificateurs CSE-ID absolus.

NOTE 1 – La manière dont le client MAF source dresse la liste de clients MAF cibles dépend de l'application.

Le client MAF source envoie une demande d'enregistrement de clé MAF comprenant les informations indiquées dans le Tableau 8.8.2.7-1.

Tableau 8.8.2.7-1 – Informations du message de demande d'enregistrement de clé MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
expirationTime	Date et heure d'expiration proposées de l'enregistrement de la clé.	1
labels	Libellé facilitant la découverte du relevé de l'enregistrement de la clé.	0 ou 1
adminFQDN	Identificateur de la partie prenante administratrice.	1
SUID	Identificateur d'utilisation de sécurité indiquant dans quelle fonctionnalité de sécurité la clé peut être utilisée.	1
targetIDs	(Facultatif) Liste des identificateurs de l'ensemble initial des clients MAF cibles autorisés à récupérer la clé symétrique.	0 ou 1
Key Value	(Facultatif) S'il est renseigné, ce paramètre contient la valeur de la clé symétrique de sortie générée automatiquement par le client MAF source. Dans le cas contraire, le client MAF source et la fonction MAF génèrent la valeur de la clé symétrique de sortie à l'aide de l'outil d'exportation TLS.	0 ou 1

La fonction MAF traite la demande. En cas d'erreur, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF autorise l'établissement d'une valeur de clé en fonction de l'identité authentifiée du client MAF source.

NOTE 2 – La présente spécification ne fournit aucune précision quant à l'autorisation de cette demande.

Si la demande a attribué une valeur au paramètre Key Value, la fonction MAF la stocke. Autrement, la fonction MAF génère la valeur de la clé de la session (D)TLS grâce à l'exportation de clés par protocole TLS [IETF RFC 5705], comme cela est décrit au § 10.3.1 (Détails relatifs à l'exportation de clés à l'aide du protocole TLS).

La fonction MAF initialise la liste des clients MAF cibles autorisés (qui peuvent récupérer le justificatif d'identité) conformément à celle qui a été fournie dans la demande.

Dans le cas d'un cadre ESData fondé sur la fonction MAF: cette liste peut être ensuite mise à jour par les parties prenantes administratrices pendant ou après la procédure d'enregistrement de clé MAF.

NOTE 3 – La présente spécification ne fournissent aucune précision concernant la mise à jour, par les parties prenantes administratrices, de la liste des clients MAF cibles autorisés de la fonction MAF. La fonction MAF peut fournir sa propre logique et interface pour permettre aux parties prenantes administratrices de gérer cette liste.

La fonction sélectionne une valeur précédemment inutilisée de l'identificateur RelativeKeyID.

La fonction MAF peut attribuer différentes valeurs aux paramètres envoyés par le client MAF suivant les instructions de la partie prenante administratrice.

La fonction MAF envoie une réponse au client MAF source comprenant les informations indiquées dans le Tableau 8.8.2.7-2.

Tableau 8.8.2.7-2 – Informations du message de réponse d'enregistrement de clé MAF

Paramètre	Description	Multiplicité
RelativeKeyID	Partie relative de l'identificateur de la clé associé à l'enregistrement de la clé.	1
expirationTime	Date et heure d'expiration de l'enregistrement de la clé.	1
Source MAF Client ID	Identificateur du client MAF source.	1
labels	Libellé facilitant la découverte du relevé de l'enregistrement de la clé.	0 ou 1
adminFQDN	Identificateur de la partie prenante administratrice.	1
SUID	Identificateur d'utilisation de sécurité indiquant dans quelle fonctionnalité de sécurité la clé peut être utilisée.	
targetIDs	Liste des identificateurs de l'ensemble initial des clients MAF cibles autorisés à récupérer la clé symétrique. Cette liste peut avoir été modifiée par rapport à celle fournie par le client MAF, ou créée par la fonction MAF (si le client MAF ne l'a pas fournie).	1

Le client MAF source et la fonction MAF stockent la valeur de la clé symétrique de sortie et l'identificateur correspondant.

L'identificateur de la clé est généré à l'aide de l'identificateur RelativeKeyID et du nom de domaine complet (FQDN) de la fonction d'authentification M2M par le client MAF source et la fonction MAF, comme cela est décrit au § 10.3.5 (Génération de l'identificateur KcID).

8.8.2.8 Procédure de récupération de la clé MAF

Finalité: cette procédure permet à un client MAF cible de récupérer, à l'aide d'une fonction MAF, la valeur de la clé correspondant à un identificateur RelativeKeyID qu'il a reçu.

Conditions préalables

Le client MAF cible a effectué la configuration des justificatifs d'identité du client MAF (§ 8.8.2.1) à l'aide de la fonction MAF, y compris la configuration de l'identificateur URI de récupération de la clé MAF.

Le client MAF source a effectué la procédure d'enregistrement de clé MAF (§ 8.8.2.2) auprès de la fonction MAF, créant ainsi une valeur de clé enregistrée, un identificateur RelativeKeyID attribué à une partie prenante administratrice et un identificateur d'utilisation de sécurité (SUID).

Le client MAF cible a reçu un identificateur de clé de la part du client MAF initiateur dans une fonctionnalité de sécurité, ainsi que l'identificateur SUID que le client MAF source a fourni à la fonction MAF pendant la procédure d'enregistrement de clé MAF (§ 8.8.2.7). L'identificateur de clé doit être composé du nom de domaine complet de la fonction MAF et de l'identificateur RelativeKeyID attribué à la clé enregistrée.

Le client MAF cible peut être autorisé à obtenir la valeur de clé symétrique de sortie correspondante.

NOTE – Le client MAF cible n'a pas à répéter cette procédure s'il détient déjà la valeur de clé correspondante.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF cible établit une session TLS (ou DTLS) avec la fonction MAF au moyen d'une prise de contact MAF, comme cela est décrit au § 8.8.2.2. Un des résultats des prises de contact MAF est que la fonction MAF crée une identité authentifiée pour le client MAF cible.

Le client MAF cible envoie une demande de récupération de clé MAF à la fonction MAF comprenant les informations indiquées dans le Tableau 8.8.2.8-1.

Tableau 8.8.2.8-1 – Informations du message de demande de récupération de clé MAF

Paramètre	Description	Multiplicité
RelativeKeyID	Partie relative de l'identificateur de la clé envoyée par le client MAF source dans une fonctionnalité de sécurité.	1

La fonction MAF traite la demande. En cas d'erreur, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF identifie l'enregistrement de la clé à l'aide de l'identificateur *RelativeKeyID*.

La fonction MAF détermine si le client MAF cible est autorisé à récupérer la clé et les métadonnées enregistrées en comparant l'identificateur authentifié du client MAF cible avec la liste des identificateurs des clients MAF cibles autorisés. Si le client MAF cible n'est pas autorisé, la fonction MAF lui envoie un message d'erreur. Autrement, la fonction MAF passe à l'étape suivante.

La fonction MAF envoie une réponse au client MAF cible comprenant les informations indiquées dans le Tableau 8.8.2.8-2.

Tableau 8.8.2.8-2 – Informations du message de réponse d'enregistrement de clé MAF

Paramètre	Description	Multiplicité
expirationTime	Date et heure d'expiration de l'enregistrement de la clé.	1
Source MAF Client ID	Identificateur du client MAF source.	1

Tableau 8.8.2.8-2 – Informations du message de réponse d'enregistrement de clé MAF

Paramètre	Description	Multiplicité
labels	Libellé facilitant la découverte du relevé de l'enregistrement de la clé.	0 ou 1
adminFQDN	Identificateur de la partie prenante administratrice.	1
SUID	Identificateur d'utilisation de sécurité indiquant dans quelle fonctionnalité de sécurité la clé peut être utilisée.	
Key Value	Valeur enregistrée de la clé symétrique de sortie	1

Le client MAF cible associe les paramètres à l'identificateur de clé.

8.8.2.9 Procédure de mise à jour de l'enregistrement de la clé MAF

Finalité: cette procédure permet à un client MAF source de mettre à jour les métadonnées associées à une clé enregistrée.

Cette procédure s'effectue entre le client MAF source et la fonction MAF.

Conditions préalables: le client MAF a exécuté au préalable la procédure d'enregistrement de clé MAF en vue de créer l'enregistrement de la clé.

L'enregistrement de clé n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF comme cela est décrit à l'étape 1 du § 8.8.2.7.

Le client MAF source dresse une liste des clients MAF cibles auxquels la fonction MAF est autorisée à fournir la clé Kc:

Dans le cas d'un cadre SAEF ou ESPrim fondé sur la fonction MAF: La liste doit contenir exactement un identificateur AE-ID absolu ou CSE-ID absolu.

Dans le cas d'un cadre ESData fondé sur la fonction MAF: la liste doit contenir n'importe quel nombre non nul d'identificateurs AE-ID absolus ou d'identificateurs CSE-ID absolus.

NOTE 1 – La présente spécification ne fournit aucune précision quant à la manière dont le client MAF source dresse la liste des clients MAF cibles.

Le client MAF source envoie une demande de mise à jour de l'enregistrement de la clé MAF comprenant les informations actualisées indiquées dans le Tableau 8.8.2.9-1.

Tableau 8.8.2.9-1 – Informations du message de la demande de mise à jour de l'enregistrement de la clé MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
RelativeKeyID	Partie relative de l'identificateur de la clé associé à l'enregistrement de la clé.	1
expirationTime	Date et heure d'expiration proposées de l'enregistrement de la clé.	0 ou 1
labels	Libellés proposés pour faciliter la découverte de la clé enregistrée.	0 ou 1
targetIDs	(Facultatif) Liste proposée des identificateurs de l'ensemble des clients MAF cibles autorisés à récupérer la clé symétrique.	0 ou 1
NOTE – Au moins l'un des paramètres expirationTime, labels et targetIDs doit être fourni.		

La fonction MAF traite la demande. En cas d'erreur, la fonction MAF envoie le message d'erreur correspondant. Si la demande est traitée correctement, la fonction MAF met à jour les métadonnées avec les valeurs proposées si la partie prenante administratrice l'autorise. La fonction MAF peut attribuer différentes valeurs aux paramètres envoyés par le client MAF suivant les instructions de la partie prenante administratrice.

La fonction MAF envoie une réponse au client MAF source comprenant les informations indiquées dans le Tableau 8.8.2.9-2.

Tableau 8.8.2.9-2 – Informations du message de mise à jour de l'enregistrement de la clé MAF

Paramètre	Description	Multiplicité
expirationTime	Date et heure d'expiration en vigueur de l'enregistrement de la clé, si cette information a été modifiée depuis qu'elle a été fournie au client MAF pour la dernière fois.	0 ou 1
labels	Liste actualisée des libellés facilitant la découverte de l'enregistrement de la clé, s'ils existent.	0 ou 1
targetIDs	Liste actuelle des identificateurs de l'ensemble initial des clients MAF cibles autorisés à récupérer la clé symétrique. Cette liste peut avoir été modifiée par rapport à celle fournie par le client MAF.	0 ou 1
NOTE – La réponse comprend uniquement les paramètres présents dans la demande correspondante.		

8.8.2.10 Procédure d'annulation de l'enregistrement de la clé MAF

Finalité: cette procédure permet à un client MAF source de demander à la fonction MAF d'arrêter la distribution de la clé enregistrée.

Cette procédure s'effectue entre le client MAF source et la fonction MAF.

Conditions préalables: le client MAF a exécuté au préalable la procédure d'enregistrement de clé MAF en vue de créer l'enregistrement de la clé.

L'enregistrement de clé n'a pas expiré.

Description de la procédure: la procédure comprend les étapes suivantes:

Le client MAF établit une connexion TLS (ou DTLS) avec la fonction MAF comme cela est décrit à l'étape 1 du § 8.8.2.7.

Le client MAF envoie une demande d'annulation de l'enregistrement de la clé MAF comprenant les informations indiquées dans le Tableau 8.8.2.10-1.

Tableau 8.8.2.10-1 – Informations du message de demande d'annulation de l'enregistrement du client MAF

Paramètre	Description	Multiplicité
MAF-FQDN	Nom de domaine complet de la fonction MAF, à partir de la configuration des instructions MAF.	1
RelativeKeyID	Partie relative de l'identificateur de la clé associé à l'enregistrement de la clé.	1

Dès la réception de la demande, la fonction MAF traite cette dernière. En cas d'erreur, la fonction MAF envoie un message d'erreur. Si la demande est traitée correctement, la fonction MAF supprime les informations associées à l'enregistrement de la clé identifiée.

La fonction MAF compose une réponse d'annulation de l'enregistrement du client MAF indiquant la réussite de l'opération. La fonction MEF envoie la réponse au client MAF.

8.8.3 Détails concernant la configuration du client MAF

8.8.3.1 Détails de la configuration des justificatifs d'identité du client MAF

Le client MAF et la fonction MAF doivent être configurés à l'aide des justificatifs d'identité afin d'assurer leur authentification mutuelle.

Les justificatifs d'identité servant à l'authentification mutuelle doivent être préconfigurés ou configurés à distance grâce aux cadres de configuration à distance de la sécurité. Des justificatifs d'identité de clé symétrique ou de certificat peuvent être fournis. Des justificatifs d'identité de clé symétrique peuvent être utilisés pour l'authentification de certains clients MAF et des justificatifs d'identité de certificat peuvent être utilisés pour l'authentification d'autres clients MAF. La sélection peut dépendre des fonctionnalités du client MAF.

Les détails dépendent du type de justificatif d'identité (clé symétrique ou certificats) et, dans le cas des clés symétriques, du type de configuration (préconfiguration ou configuration à distance).

Détails spécifiques aux **clés symétriques préconfigurées (PPSK)**: le justificatif d'identité principal (Km) et l'identificateur correspondant (KmID) doivent être fournis au client MAF (agissant en tant qu'entité inscrite) et à la fonction MAF. Le format de l'identificateur KmID est défini au § 10.6 (Format de l'identificateur KmID).

Détails spécifiques aux **clés symétriques fournies à distance (RPSK)**: le client MAF et une fonction d'inscription M2M (MEF) doivent être dotés de justificatifs d'identité pour exécuter un cadre de configuration à distance de la sécurité (RSPF). Le client MAF doit être autorisé à utiliser les services de la fonction MEF. Pour plus de détails, voir le § 8.3.

NOTE 1 – Dans ce cas, le justificatif d'identité principal (Km) et son identificateur (KmID) sont créés pendant la procédure d'enregistrement du client MAF.

Détails spécifiques aux **certificats (préconfigurés ou configurés à distance)**: le client MAF est doté d'un certificat de client MAF avec une chaîne de certificats facultative. Le certificat du client MAF doit être un certificat de dispositif, un certificat AE-ID ou un certificat CSE-ID.

NOTE 2 – La configuration des certificats de la CA constituant l'ancre de confiance MAF est effectuée durant la configuration de l'enregistrement du client MAF et peut être exécutée séparément de la configuration des justificatifs d'identité du client MAF.

La spécification de configuration de dispositif oneM2M [UIT-T Y.4500.22] définit un ensemble de spécialisations <mgmtObj> qui doit être utilisé pour la configuration des justificatifs d'identité du client MAF lorsque ce dernier prend en charge la gestion des dispositifs (soit à distance, soit par entrée manuelle). La présente Recommandation ne précise pas comment la configuration du justificatif d'identité du client MAF est représentée lorsque ce dernier ne prend pas en charge la gestion des dispositifs.

8.8.3.2 Détails concernant la configuration de l'enregistrement du client MAF

Finalité: la configuration d'enregistrement du client MAF décrit les informations fournies à un client MAF pour lui permettre d'exécuter les procédures MAF autorisées par une partie prenante administratrice. La partie prenante administratrice prend les dispositions nécessaires pour que la configuration d'enregistrement du client MAF soit fournie au client MAF.

Conditions préalables: le client MAF et la fonction MAF ont été configurés à l'aide de justificatifs d'identité d'authentification mutuelle (voir les détails relatifs à la configuration des justificatifs d'identité du client MAF au § 8.8.3.1).

Si le client MAF et la fonction MAF utilisent des certificats pour la procédure d'authentification mutuelle:

– la partie prenante administratrice (ou toute autre partie prenante agissant au nom de celle-ci) détient une copie des informations du certificat du client MAF définies au § 8.1.2.4. La fonction MAF est dotée d'une copie des informations du certificat du client MAF. La présente Recommandation ne précise pas comment ces informations sont fournies à la fonction MAF par la partie prenante administratrice (ou par toute autre partie prenante agissant au nom de celle-ci);

– la partie prenante administratrice (ou toute autre partie prenante agissant au nom de celle-ci) possède une copie des certificats de la CA constituant l'ancre de confiance MAF. Le client MAF est doté d'une copie des certificats de la CA constituant l'ancre de confiance MAF.

La partie prenante administratrice prend les dispositions nécessaires pour que la fonction MAF permette au client MAF d'effectuer son enregistrement. Cela peut inclure une préautorisation ou une autorisation en temps réel.

Détails: la configuration d'enregistrement du client MAF (*mafClientRegCfg*) inclut les informations indiquées dans le Tableau 8.8.3.2-1 et présente le type de données sec:ClientRegCfg (voir le § 12.4.2).

Tableau 8.8.3.2-1 – Informations contenues dans la configuration de l'enregistrement du client MAF

Nom de l'élément	Multiplicité	Notes
expirationTime	0 ou 1	Date et heure d'expiration de la configuration
labels	0 ou 1	Liste de libellés permettant la découverte du relevé de l'enregistrement du client MAF.
fqdn	1	Nom de domaine complet de la fonction MAF (aussi appelé MAF-ID)
adminFQDN	1	Nom de domaine complet de la partie prenante administratrice.
httpPort	0 ou 1	Numéro de port ouvert lors de l'utilisation du protocole HTTP [b-IETF RFC 7730]
coapPort	0 ou 1	Numéro de port ouvert lors de l'utilisation du protocole CoAP [b-IETF RFC 7252]
websocketPort	0 ou 1	Numéro de port ouvert lors de l'utilisation du protocole WebSocket [b-IETF RFC 6455]

8.8.3.3 Détails concernant la configuration de l'enregistrement de la clé MAF

Finalité: la configuration d'enregistrement de la clé MAF décrit les informations fournies à un client MAF pour lui permettre d'exécuter les procédures MAF autorisées par une partie prenante administratrice. La partie prenante administratrice prend les dispositions nécessaires pour que la configuration d'enregistrement du client MAF soit fournie au client MAF.

Conditions préalables: le client MAF a effectué la procédure d'enregistrement de client MAF auprès de la fonction MAF pour la partie prenante administratrice.

Le client MAF possède des justificatifs d'identité valides pour effectuer l'authentification mutuelle à l'aide de la fonction MAF.

Détails: la configuration d'enregistrement de clé MAF (*mafKeyRegCfg*) contient les informations indiquées dans le Tableau 8.8.3.3-1 et présente le type de données sec:keyRegCfg (voir le § 12.4.3).

Tableau 8.8.3.3-1 – Informations contenues dans la configuration de l'enregistrement de la clé MAF

Nom de l'élément	Multiplicité	Notes
expirationTime	0 ou 1	Date et heure d'expiration
labels	0 ou 1	Liste de libellés permettant la découverte de l'enregistrement de la clé
adminFQDN	1	Nom de domaine complet de la partie prenante administratrice
SUID	1	Identificateur SUID contraignant l'utilisation de la valeur de clé établie pendant la procédure d'enregistrement de clé MAF
targetIDs	0 ou 1	Liste des identificateurs des clients MAF cibles autorisés

9 Procédures et paramètres des cadres de sécurité

9.0 Introduction

Le présent paragraphe décrit les procédures et les paramètres des phases des cadres d'établissement d'association de sécurité (voir le § 8.2) et de configuration à distance de la sécurité (voir le § 8.3).

9.1 Procédure et paramètres du cadre d'établissement d'association de sécurité (SAEF)

9.1.1 Paramètres de configuration des justificatifs d'identité

9.1.1.0 Introduction

Les procédures de configuration de justificatifs d'identité suivantes sont décrites dans le présent paragraphe:

– Configuration des justificatifs d'identité de l'entité A et de l'entité B (voir § 9.1.1.1).

– Configuration des justificatifs d'identité des fonctions d'authentification M2M (voir § 9.1.1.2).

9.1.1.1 Configuration des justificatifs d'identité de l'entité A et de l'entité B

Le Tableau 9.1.1.1-1 énumère le paramètre qui peut être configuré pour l'entité A pendant la phase de configuration des justificatifs d'identité et qui est commun à tous les cadres d'établissement d'association de sécurité.

Tableau 9.1.1.1-1 – Paramètre qui peut être configuré pour l'entité A pendant la phase de configuration des justificatifs d'identité et qui est commun à tous les cadres d'établissement d'association de sécurité

Paramètre commun à tous les cadres d'établissement d'association de sécurité
(Si l'entité A est une entité de services communs) Identificateur CSE-ID de l'entité A

Le Tableau 9.1.1.1-2 énumère les paramètres propres au cadre d'établissement d'association de sécurité configurés pour les points d'extrémité d'association de sécurité du domaine de terrain durant la phase de configuration des justificatifs d'identité.

Tableau 9.1.1.1-2 – Paramètres spécifiques au cadre d'établissement d'association de sécurité configurés pour les points d'extrémité d'association de sécurité du domaine de terrain durant la phase de configuration des justificatifs d'identité

Cadre d'établissement d'association de sécurité		Paramètre
Clé symétrique fournie		Kpsa
		KpsaID
Fondé sur les certificats	L'entité s'authentifie à l'aide d'un certificat de clé publique brute	Clé privée de l'entité
		Certificat de clé publique brute de l'entité
	L'entité s'authentifie à l'aide d'un certificat de dispositif	Clé privée de l'entité
		Certificat et chaîne de l'entité
	L'entité s'authentifie à l'aide d'un certificat CSE-ID	Identificateur CSE-ID de l'entité
		Clé privée de l'entité
		Certificat et chaîne de l'entité
	L'entité s'authentifie à l'aide d'un certificat AE-ID	Identificateur AE-ID de l'entité
		Clé privée de l'entité
		Certificat et chaîne de l'entité
Fondé sur la fonction MAF	Entité A	Identificateur de la fonction MAF (MAF-ID)
		Justificatif d'identité principal (KmID)
		Identificateur du justificatif d'identité principal (KmID)
	Entité B	Les entités B et la fonction MAF doivent être en mesure d'établir des communications sécurisées mutuellement authentifiées. Les détails à cet égard ne sont pas indiqués dans la présente Recommandation.

La configuration des justificatifs d'identité de l'entité A et de l'entité B pour le cadre d'établissement d'association de sécurité par clé symétrique fournie, ou pour le cadre d'établissement d'association de sécurité fondé sur la fonction MAF, est réalisée par:

– préconfiguration grâce à des mécanismes qui ne sont pas précisés dans la présente Recommandation; ou

– configuration à distance à l'aide d'un des cadres de configuration à distance de la sécurité définis au § 8.3.

La configuration des justificatifs d'identité de l'entité A et de l'entité B pour les cadres d'établissement d'association de sécurité de certificats s'effectue par préconfiguration au moyen de mécanismes qui ne sont pas décrits dans la présente Recommandation.

9.1.1.2 Configuration des justificatifs d'identité des fonctions d'authentification M2M

Le Tableau 9.1.1.2-1 énumère les paramètres configurés pour les fonctions d'authentification M2M dans la phase de configuration des justificatifs d'identité. L'identificateur de la fonction d'authentification M2M (MAF-ID) est supposé avoir été configuré avant la phase de configuration des justificatifs d'identité.

Tableau 9.1.1.2-1 – Paramètres configurés pour les fonctions d'authentification M2M durant la phase de configuration des justificatifs d'identité

Cadre d'établissement d'association de sécurité		Paramètre
Fondé sur la fonction MAF	A vers authentification MAF	Justificatif d'identité principal (Km)
		Identificateur du justificatif d'identité principal (KmID)
	B vers authentification MAF	Les entités B et la fonction MAF doivent être en mesure d'établir des communications sécurisées mutuellement authentifiées. Les détails à cet égard ne sont pas indiqués dans la présente Recommandation.

La configuration des justificatifs d'identité du cadre d'authentification M2M doit être effectuée à l'aide de l'un des éléments suivants:

– la logique métier de la partie prenante qui exécute la fonction d'authentification M2M, qui n'est pas décrite en détail dans la présente Recommandation;

– configuration à distance à l'aide d'un des cadres de configuration à distance de la sécurité définis au § 8.3.

9.1.2 Procédures et paramètres de configuration des associations

9.1.2.0 Introduction

Les procédures de configuration des associations suivantes sont décrites dans le présent paragraphe:

– Configuration de l'association de l'entité A (voir § 9.1.2.1.1).

– Configuration de l'association de l'entité B (voir § 9.1.2.1.2).

– Configuration de l'association des fonctions d'authentification M2M (voir § 9.1.2.2).

9.1.2.1 Configuration de l'association de l'entité A et de l'entité B

9.1.2.1.1 Configuration de l'association de l'entité A

Le Tableau 9.1.2.1.1-1 énumère le paramètre configuré pour l'entité A pendant la phase de configuration de l'association et qui est commun à tous les cadres d'établissement d'association de sécurité.

Tableau 9.1.2.1.1-1 – Paramètre configuré pour l'entité A pendant la phase de configuration de l'association et qui est commun à tous les cadres d'établissement d'association de sécurité

Paramètre commun à tous les cadres d'établissement d'association de sécurité
Identificateur CSE-ID de l'entité B

Le Tableau 9.1.2.1.1-1 énumère les paramètres configurés pour l'entité A pendant la phase de configuration de l'association et qui sont spécifiques à un cadre d'établissement d'association de sécurité.

Tableau 9.1.2.1.1-2 – Paramètres configurés pour l'entité A pendant la phase de configuration de l'association et qui sont spécifiques à un cadre d'établissement d'association de sécurité

Cadre d'établissement d'association de sécurité		Paramètres spécifiques aux différents cadres d'établissement d'association de sécurité
Clé symétrique fournie		Aucun
Fondé sur les certificats	L'entité B est authentifiée à l'aide d'un certificat de clé publique brute	Identificateur de clé publique de l'entité B
	L'entité B est authentifiée à l'aide d'un certificat de dispositif	Identificateur unique mondialement d'instance matérielle de l'entité B
		Informations de l'ancre de confiance de l'entité B
	L'entité B est authentifiée à l'aide d'un certificat CSE-ID	Informations de l'ancre de confiance de l'entité B
	L'entité B est authentifiée à l'aide d'un certificat AE-ID	Informations de l'ancre de confiance de l'entité B
Fondé sur la fonction MAF		Aucun

Les mécanismes de configuration d'association de l'entité A doivent authentifier la source de configuration et assurer la protection de l'intégrité des informations configurées transmises par la source de configuration à l'entité.

9.1.2.1.2 Configuration de l'association de l'entité B

Le Tableau 9.1.2.1.1-1 énumère les paramètres configurés pour le registre (entité B) pendant la phase de configuration de l'association.

Tableau 9.1.2.1.2-1 – Paramètres configurés pour l'entité B pendant la phase de configuration de l'association

Cadre d'établissement d'association de sécurité		Paramètres spécifiques aux différents cadres d'établissement d'association de sécurité
Clé symétrique fournie		Aucun
Fondé sur les certificats	L'entité B est authentifiée à l'aide d'un certificat de clé publique brute	Aucun
	L'entité B est authentifiée à l'aide d'un certificat de dispositif, d'un certificat CSE-ID ou d'un certificat AE-ID	Informations de l'ancre de confiance de l'entité A
Fondé sur la fonction MAF		Aucun

Les mécanismes de configuration d'association de l'entité B doivent authentifier la source de configuration et assurer la protection de l'intégrité des informations configurées transmises par la source de configuration à l'entité.

9.1.2.2 Configuration de l'association des fonctions d'authentification M2M

Le Tableau 9.1.2.2-1 indique le paramètre configuré pour les fonctions d'authentification M2M dans la phase de configuration d'association.

Tableau 9.1.2.2-1 – Paramètres configurés pour les fonctions d'authentification M2M durant la phase de configuration d'association

Cadre d'établissement d'association de sécurité		Paramètre
Fondé sur la fonction MAF	A vers authentification MAF	Identificateur CSE-ID ou AE-ID de l'entité B (IdB)

Dans la présente Recommandation, il est supposé que la configuration d'association des fonctions d'authentification M2M utilisera la logique métier de la partie prenante qui exécute la fonction d'authentification M2M et qui n'est pas décrite en détail dans la Recommandation.

9.2 Procédures et paramètres des cadres de configuration à distance de la sécurité

9.2.1 Procédures et paramètres de la configuration des justificatifs d'amorçage

9.2.1.0 Introduction

Les procédures de configuration des justificatifs d'amorçage suivantes sont décrites dans le présent paragraphe:

– Configuration des justificatifs d'amorçage des entités inscrites et des cibles d'inscription (sauf dans le cas de l'architecture GBA, comme indiqué ci-dessous), voir le § 9.2.1.1.

– Configuration des justificatifs d'amorçage des fonctions d'inscription M2M (sauf dans le cas de l'architecture GBA, comme indiqué ci-dessus), voir le § 9.2.1.2.

Les procédures de configuration des justificatifs d'amorçage suivantes sont définies par d'autres organisations:

– Configuration des justificatifs d'amorçage des serveurs d'authentification du fournisseur de services du réseau sous-jacent (par exemple HLR, HSS ou AAA) pour le cadre d'établissement d'association de sécurité fondé sur l'architecture GBA. Les détails y afférents sont disponibles dans les spécifications 3GPP TS 33.220 [ETSI TS 133 220] et 3GPP2 S.S0109-A [TIA-1098-A].

– Configuration des justificatifs d'amorçage des entités inscrites pour le cadre d'établissement d'association de sécurité fondé sur l'architecture GBA. Les détails y afférents sont disponibles dans les spécifications 3GPP TS 33.220 [ETSI TS 133 220] et 3GPP2 S.S0109-A [TIA-1098-A].

9.2.1.1 Configuration des justificatifs d'amorçage des entités inscrites

Le Tableau 9.2.1.1-1 énumère les paramètres configurés pour les entités inscrites durant la phase de configuration des justificatifs d'amorçage, aux fins d'authentification avec la fonction d'inscription M2M dans le cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et dans le cadre de configuration à distance de la sécurité fondé sur les certificats.

Tableau 9.2.1.1-1 – Paramètres configurés pour les entités inscrites durant la phase de configuration des justificatifs d'amorçage

Cadre de configuration à distance de la sécurité	Paramètre
Authentification de la clé de connexion sécurisée M2M préconfigurée. Non applicable à la fonction MAF	Kpm
	KpmID
	URI de la fonction MEF

Tableau 9.2.1.1-1 – Paramètres configurés pour les entités inscrites durant la phase de configuration des justificatifs d'amorçage

Cadre de configuration à distance de la sécurité		Paramètre
Authentification fondée sur les certificats	L'entité inscrite s'authentifie à l'aide d'une clé publique brute	Clé privée de l'entité inscrite
		Certificat de clé publique brute de l'entité inscrite
	L'entité inscrite s'authentifie à l'aide d'un certificat de dispositif	Clé privée de l'entité inscrite
		Certificat et chaîne de l'entité inscrite
	L'entité inscrite s'authentifie à l'aide d'un certificat CSE-ID et AE-ID	Clé privée de l'entité inscrite
		Certificat et chaîne de l'entité inscrite

La configuration du justificatif d'amorçage d'une entité inscrite pour le cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et pour le cadre de configuration à distance de la sécurité fondé sur les certificats doit authentifier la source de configuration et assurer la protection de la confidentialité et de l'intégrité des informations configurées transmises par ladite source à l'environnement sécurisé de l'entité inscrite. La présente Recommandation ne décrit aucun mécanisme de ce genre.

Dans la présente Recommandation, aucun détail n'est fourni concernant la configuration des justificatifs d'amorçage d'une cible d'inscription d'un domaine d'infrastructure (y compris une fonction d'authentification M2M) censée utiliser la logique métier de la partie prenante qui effectue l'inscription dans le domaine d'infrastructure.

9.2.1.2 Configuration des justificatifs d'amorçage des fonctions d'inscription M2M

Il est supposé qu'une fonction d'inscription M2M connaît déjà son nom de domaine complet.

Le Tableau 9.2.1.2-1 énumère les paramètres configurés pour les fonctions d'inscription M2M durant la phase de configuration des justificatifs d'amorçage, aux fins de l'authentification mutuelle avec les entités inscrites et les cibles d'inscription à l'aide du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondé sur les certificats.

Tableau 9.2.1.2-1 – Paramètres configurés pour la fonction d'inscription M2M durant la phase de configuration des justificatifs d'amorçage, aux fins de l'authentification mutuelle avec les entités inscrites et les cibles d'inscription à l'aide du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondé sur les certificats

Cadre de configuration à distance de la sécurité	Paramètres spécifiques aux cadres de configuration à distance de la sécurité
Authentification de la clé symétrique d'inscription préconfigurée de l'entité inscrite ou de la cible d'inscription	Kpm
	KpmID
Authentification de l'entité inscrite ou de la cible d'inscription fondée sur les certificats	Clé privée de la fonction MEF
	Certificat et chaîne de la fonction MEF

La configuration des justificatifs d'amorçage des fonctions d'inscription M2M est censée utiliser la logique métier de la partie prenante qui exécute la fonction d'inscription M2M. Aucun détail à cet égard n'est fourni dans la présente Recommandation.

9.2.2 Procédures et paramètres de la configuration des instructions d'amorçage

9.2.2.0 Introduction

Les procédures de configuration des instructions d'amorçage suivantes sont décrites dans le présent paragraphe:

– Configuration des instructions d'amorçage des entités inscrites (voir § 9.2.2.1).

– Configuration des instructions d'amorçage des fonctions d'inscription M2M (voir § 9.2.2.3).

– Configuration des instructions d'amorçage des serveurs d'authentification du fournisseur de services du réseau sous-jacent (par exemple HLR, HSS ou AAA), voir § 9.2.2.4.

9.2.2.1 Configuration des instructions d'amorçage des entités inscrites.

Le Tableau 9.2.2.1-1 indique le paramètre configuré pour une entité inscrite au cours de la phase de configuration des instructions d'amorçage qui est commun à tous les cadres de configuration à distance de la sécurité.

Tableau 9.2.2.1-1 – Paramètre configuré pour une entité inscrite au cours de la phase de configuration des instructions d'amorçage qui est commun à tous les cadres de configuration à distance de la sécurité

Paramètre commun à tous les cadres de configuration à distance de la sécurité
Identificateur de la cible d'inscription (identificateur AE-ID, CSE-ID ou MAF-ID de l'entité inscrite B)

Le Tableau 9.2.2.1-2 énumère les paramètres spécifiques aux cadres de configuration à distance de la sécurité configurés pour une entité inscrite durant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité.

Tableau 9.2.2.1-2 – Paramètres spécifiques aux cadres de configuration à distance de la sécurité configurés pour une entité inscrite durant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité

Cadre de configuration à distance de la sécurité	Paramètres spécifiques aux cadres de configuration à distance de la sécurité
Clé symétrique d'inscription préconfigurée	Expiration de l'inscription
Fondé sur les certificats	URI de la fonction MEF
	Informations de l'ancre de confiance de la fonction MEF
Fondé sur l'architecture GBA	*Aucun*

Les mécanismes de configuration d'instructions d'amorçage de l'entité inscrite doivent authentifier la source de configuration et assurer au moins la protection de l'intégrité des informations configurées transmises par la source de configuration à l'entité inscrite.

9.2.2.2 Paragraphe intentionnellement laissé en blanc

9.2.2.3 Configuration des instructions d'amorçage des fonctions d'inscription M2M

Le Tableau 9.2.2.3-1 indique le paramètre configuré pour une fonction d'inscription M2M durant la phase de configuration des instructions d'amorçage qui est commun au cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondé sur les certificats.

Tableau 9.2.2.3-1 – Paramètre configuré pour les fonctions d'inscription M2M durant la phase de configuration des instructions d'amorçage et qui est commun au cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondée sur les certificats

Paramètre commun à tous les cadres de configuration à distance de la sécurité
Identificateur de la cible d'inscription (identificateur CSE-ID, AE-ID ou MAF-ID de l'entité inscrite B)

Le Tableau 9.2.2.3-2 énumère les paramètres configurés spécifiques au cadre de configuration à distance de la sécurité configurés pour des fonctions d'inscription M2M durant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondé sur les certificats.

Tableau 9.2.2.3-2 – paramètres configurés spécifiques au cadre de configuration à distance de la sécurité configurés pour la fonction d'inscription M2M durant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité fondé sur une clé symétrique préconfigurée d'entité inscrite et du cadre de configuration à distance de la sécurité fondé sur les certificats

Cadre de configuration à distance de la sécurité		Paramètres spécifiques aux cadres de configuration à distance de la sécurité
Clé symétrique d'inscription préconfigurée		Expiration de l'inscription
Fondé sur les certificats	L'entité inscrite est authentifiée à l'aide d'un certificat de clé publique brute	Identificateur de clé publique de l'entité inscrite
	L'entité inscrite est authentifiée à l'aide d'un certificat de dispositif	Identificateur de dispositif M2M de l'entité inscrite
		Informations de l'ancre de confiance de l'entité inscrite
	L'entité inscrite est authentifiée à l'aide d'un certificat CSE-ID et AE-ID	Informations de l'ancre de confiance de l'entité inscrite

Dans la présente Recommandation, il est supposé que la configuration des instructions d'amorçage des fonctions d'inscription M2M utilise la logique métier de la partie prenante qui exécute la fonction d'inscription M2M et qui n'est pas décrite en détail dans la Recommandation.

9.2.2.4 Configuration des instructions d'amorçage d'un serveur d'authentification de l'UNSP

Le Tableau 9.2.2.4-1 indique le paramètre configuré pour un serveur d'authentification du fournisseur de services du réseau sous-jacent (UNSP; par exemple HLR, HSS ou AAA) pendant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité fondé sur l'architecture GBA.

Tableau 9.2.2.4-1 – Paramètre configuré pour les fonctions d'inscription M2M pendant la phase de configuration des instructions d'amorçage du cadre de configuration à distance de la sécurité fondé sur l'architecture GBA

Paramètre	Obligatoire/facultatif pour tous les cadres de configuration à distance de la sécurité
Identificateur de la cible d'inscription (identificateur CSE-ID, AE-ID ou MAF-ID de l'entité inscrite B)	Obligatoire

La configuration des instructions d'amorçage du serveur d'authentification du fournisseur de services du réseau sous-jacent est réalisée en mettant à jour les paramètres de sécurité de l'utilisateur GBA (GUSS) (3GPP TS 33.220 [ETSI TS 133 220]) de l'équipement d'utilisateur (UE) sur lequel l'entité inscrite est exécutée. Dans la présente Recommandation, il est supposé que cette configuration des instructions d'amorçage utilise la logique métier du fournisseur de services du réseau sous-jacent. Aucun détail à cet égard n'est fourni dans la présente Recommandation.

9.2.3 Procédures et paramètres de la configuration des justificatifs d'identité de bout en bout

9.2.3.0 Introduction

Les procédures de configuration des justificatifs d'identité de bout en bout suivantes sont décrites dans le présent paragraphe:

– Configuration des justificatifs d'identité de bout en bout des points d'extrémité ESF source et cible (voir § 9.2.3.1).

– Configuration des justificatifs d'identité de bout en bout des fonctions TEF (voir § 9.2.3.2).

– Paramètres de configuration permettant d'assurer la sécurité de bout en bout des points d'extrémité ESF sources et cibles (voir § 9.2.3.3).

9.2.3.1 Configuration des justificatifs d'identité de bout en bout des points d'extrémité ESF source et cible

Il est supposé que le point d'extrémité ESF source et les points d'extrémité ESF cibles sont configurés avec l'identificateur URI de la fonction TEF et qu'ils ont été configurés avec les paramètres appropriés propres aux cadres de configuration à distance de la sécurité décrits au § 9.2. En outre, les justificatifs d'identité de bout en bout sont configurés et les paramètres de sécurité appropriés sont fournis aux points d'extrémité ESF cibles, tandis que le point d'extrémité ESF source peut en déduire les justificatifs d'identité de bout en bout à l'aide des paramètres de sécurité pertinents qui ont été fournis. Le Tableau 9.2.3.1-1 dresse la liste des paramètres.

Tableau 9.2.3.1-1 – Justificatif de sécurité et paramètres fournis aux points d'extrémité ESF sources et cibles

Protection de la sécurité	Paramètres du cadre de configuration de sécurité de bout en bout	Description
Justificatifs de sécurité de bout en bout	KpsaID	Identificateur du justificatif d'identité fourni de la clé symétrique M2M fournie.
	Kpsa	Il s'agit de la clé symétrique M2M fournie. Elle est utilisée pour calculer le secret principal de bout en bout Ke2e_master décrit au § 10.3.6.
	URI de la fonction TEF	Identificateur URI de l'entité tierce de confiance (TEF) qui est utilisé comme générateur/registre de justificatifs d'identité et permet l'enregistrement et la génération des justificatifs de sécurité de bout en bout.
Paramètres cryptographiques	Salage (Salt)	Salage utilisé pour générer des justificatifs d'identité de bout en bout. Paramètre facultatif.
	Algorithme d'extraction de clé: hachage HMAC	Algorithme d'extraction de clés utilisé pour générer les diverses clés doit suivre les mécanismes décrits dans [IETF RFC 5869].
	Libellés cryptographiques	Libellés utilisés par les algorithmes cryptographiques. Ils doivent être utilisés conformément au § 10.3.6.1.
Types de justificatifs d'identité	Authenticité du message (primitive)	Clé utilisée pour l'authentification et la vérification de l'intégrité du message des primitives oneM2M. Si les éléments de calcul de clé sont fournis, la clé est générée par le point d'extrémité ESF cible.
	Confidentialité du message (primitive)	Clé utilisée pour la confidentialité du message des primitives oneM2M. Si les éléments de calcul de clé sont fournis, la clé est générée par le point d'extrémité ESF cible.
	Intégrité des données (attribut)	Clé utilisée pour assurer l'intégrité des données/attributs. Si les éléments de calcul de clé sont fournis, la clé est générée par le point d'extrémité ESF cible.
	Confidentialité des données (attribut)	Clé utilisée pour assurer la confidentialité des données/attributs. Si les éléments de calcul de clé sont fournis, la clé est générée par le point d'extrémité ESF cible.

9.2.3.2 Configuration des justificatifs d'identité de bout en bout dans les fonctions TEF M2M

Il est supposé que la fonction TEF est configurée avec l'identité des entités (points d'extrémité ESF sources et cibles) et les paramètres appropriés propres aux cadres de configuration à distance de la sécurité décrits au § 9.2.

En outre, la fonction TEF est dotée des paramètres de sécurité appropriés, de façon que les justificatifs de sécurité de bout en bout puissent être calculés et que l'ensemble des paramètres cryptographiques puisse être fourni aux points d'extrémité ESF cibles une fois qu'ils ont été authentifiés. Le Tableau 9.2.3.2-1 dresse la liste des paramètres.

Tableau 9.2.3.2-1 – Paramètres de sécurité configurés dans la fonction d'inscription M2M ou TEF et au point d'extrémité ESF source

Protection de la sécurité de bout en bout	Paramètres du cadre de configuration de sécurité de bout en bout	Description
Justificatifs de sécurité de bout en bout	Kpm	Justificatifs d'identité préconfigurés entre le point d'extrémité ESF source et la fonction TEF.
	KpmID	Identificateur des justificatifs d'identité préconfigurés.
	Identité du point d'extrémité ESF source (AE-ID/CSE-ID). Identité du point d'extrémité ESF cible (CSE-ID).	Identité de l'entité préconfigurée avec les justificatifs de sécurité de bout en bout.
Liste des éléments de protection et de niveau de sécurité requis	Authentification du message: (faible/élevée)	Indique le niveau de sécurité requis du mécanisme d'authentification du message.
	Confidentialité du message: (faible/élevée)	Indique le niveau de sécurité requis du mécanisme de confidentialité du message.
	Intégrité des attributs: (faible/élevée)	Indique le niveau de sécurité requis du mécanisme de vérification de l'intégrité des attributs.
	Confidentialité des attributs: (faible/élevée)	Indique le niveau de sécurité requis du mécanisme de confidentialité des attributs.

9.2.3.3 Paramètres de configuration permettant d'assurer la sécurité de bout en bout des points d'extrémité ESF sources et cibles

Les points d'extrémité ESF sources et cibles sont configurés avec les paramètres cryptographiques servant à activer et à vérifier la protection de la sécurité de bout en bout. Dans le cas du point d'extrémité ESF cible, la fonction TEF lui fournit les paramètres après l'authentification et le calcul réussis de la clé de connexion sécurisée (Kpsa). Dans le cas du point d'extrémité ESF source, les paramètres peuvent avoir été préconfigurés ou fournis de la même manière que pour le point d'extrémité ESF cible, c'est-à-dire une fois que la clé de connexion sécurisée (Kpsa) est calculée et partagée entre les points d'extrémité ESF source et cible. Le Tableau 9.2.3.3-1 dresse la liste des paramètres.

Tableau 9.2.3.3-1 – Paramètres de sécurité fournis aux points d'extrémité ESF source et cible

Protection de la sécurité de bout en bout	Paramètres du cadre de configuration de sécurité de bout en bout	Description
Justificatifs de sécurité de bout en bout	e2e_master	Justificatif d'identité principal de bout en bout.
	E2EKeyId	Identité du justificatif principal de bout en bout.
	Identité du point d'extrémité ESF cible (CSE-ID) Identité du point d'extrémité ESF source (AE-ID/CSE-ID)	Identité de l'entité d'extrémité avec laquelle le justificatif d'identité de bout en bout est associé.
Paramètres cryptographiques	Protocole: JWS/JWE et XML Sec	Type d'encodage et de représentation utilisé.
	Classe d'algorithmes cryptographiques: AEAD (clé unique) ou non AEAD	Définit la classe d'algorithmes cryptographiques à utiliser.
	Algorithme d'authenticité du message/taille: HMAC-SHA-256 et HMAC-SHA-512	Indique l'algorithme d'authentification du message et la taille de la clé.
	Algorithme de confidentialité du message/taille: AES-192/256	Indique l'algorithme de confidentialité du message et la taille de la clé.
	Algorithme de confidentialité des attributs: AES-192/256	Algorithme de confidentialité des attributs et taille de la clé.
	Algorithme d'authenticité des attributs/taille: HMAC-SHA-256	Algorithme de vérification de l'authenticité et de l'intégrité des attributs et taille de la clé.
Utilisation cryptographique	Authenticité du message/des attributs: nonce	Valeur aléatoire utilisée pour garantir une certaine actualité. Elle n'est stockée que temporairement, associée à une date d'expiration et transmise à l'autre point d'extrémité.
	Confidentialité du message/des attributs: vecteur d'initialisation	Valeur aléatoire utilisée en tant que vecteur d'initialisation pour l'algorithme de confidentialité.

NOTE – Pour la classe d'algorithmes AEAD où une seule clé est utilisée, une seule clé sera générée et un algorithme cryptographique associé (par exemple AES-GCM ou AES-CCM) sera identifié. En outre, pour la classe d'algorithmes AEAD, un vecteur d'initialisation et un nonce ne seraient pas générés ensemble. Seule une valeur aléatoire, un nonce, serait plutôt générée.

10 Détails concernant les protocoles et les algorithmes

10.1 Détails concernant le cadre de sécurité fondé sur les certificats

10.1.1 Profils de certificat

Les documents RFC qui identifient les algorithmes cryptographiques ne sont cités dans cette section qu'à titre informatif. Ils n'ont pas pour objectif d'être une liste exclusive ou définitive. De plus, ils ne constituent aucunement une partie normative de la présente Recommandation.

10.1.1.0 Généralités

NOTE – Ces profils de certificat sont conformes à la spécification CoAP [b-IETF RFC 7252].

10.1.1.1 Détails concernant les certificats communs

Tous les certificats doivent être conformes au profil suivant:

– Les certificats doivent être conformes à [IETF RFC 5280].

– Le certificat doit inclure un élément SubjectPublicKeyInfo indiquant un algorithme de l'élément id-ecPublicKey comprenant la courbe namedCurves secp256r1 [IETF RFC 5280]. Cette courbe est équivalente à la courbe P-256 de [NIST EC].

– Le format de la clé publique doit être non compressé conformément à [IETF RFC 5480].

– L'algorithme de hachage doit être SHA-256.

– L'extension d'utilisation de clé doit être incluse et indiquer au moins l'élément digitalSignature.

10.1.1.2 Profil des certificats de clé publique brute

Les certificats de clé publique brute doivent être conformes au § 10.1.1.1 (Détails concernant les certificats communs) et à [IETF RFC 7250].

10.1.1.3 Détails communs aux certificats dotés de chaînes de certificats

Les certificats dotés de chaînes de certificats doivent être conformes à la description suivante:

– Ces certificats doivent être conformes au § 10.1.1.1 (Détails concernant les certificats communs).

– Les certificats doit être signés à l'aide de l'algorithme ECDSA en utilisant la courbe secp256r1 et la signature devra utiliser le protocole SHA-256.

– Les chaînes de certificats devraient limiter le nombre de certificats de CA intermédiaires afin d'éviter tout impact négatif sur les environnements soumis à des contraintes.

10.1.1.4 Profil des certificats de dispositif et de leurs chaînes de certificats

10.1.1.4.1 Profil des certificats de dispositif

Les certificats de dispositif doivent être conformes à la description suivante:

– Les certificats de dispositif doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

– L'extension subjectAltName des certificats de dispositif doit inclure au moins un identificateur unique mondialement d'instance matérielle.

EXEMPLE: L'Annexe H (Identificateur de dispositif M2M fondé sur un identificateur d'objet; [UIT-T Y.4500.1]) définit un identificateur de dispositif M2M fondé sur un identificateur d'objet qui peut être utilisé pour fournir un ou plusieurs identificateurs uniques mondialement d'instance matérielle. Un identificateur de dispositif M2M fondé sur l'identificateur d'un objet peut être représenté dans un champ otherName de l'extension subjectAltName, où:

– l'élément "type-ID" du champ otherName est égal à l'arc d'identification du dispositif M2M (§ H.2.1 "Identificateur de dispositif M2M [ITU-T Y.4500.1]) de l'identificateur de dispositif M2M fondé sur l'identificateur d'un objet; et

– l'élément "value" du champ otherName est égal au reste de l'identificateur de dispositif M2M fondé sur l'identificateur d'un objet, à savoir l'arc d'identification du fabricant, l'arc d'identification du modèle, l'arc d'identification du numéro de série et l'arc d'identification étendu facultatif (voir l'Annexe H.2 "Identificateur de dispositif M2M fondé sur l'identificateur OID" [UIT-T Y.4500.1]).

NOTE – Dans certains cas, le fait de fournir l'identificateur de modèle comme partie de l'identificateur du dispositif M2M peut avoir des conséquences en matière de vie privée.

10.1.1.4.2 Profil des certificats d'autorité de certification destinés aux certificats de dispositif

Les certificats d'autorité de certification présents dans la chaîne de certificats d'un certificat de dispositif doivent être conformes à la description suivante:

Ces certificats doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

Il est recommandé que les certificats d'autorité de certification destinés aux certificats de dispositif utilisent l'extension de contraintes de nom (voir le § 4.2.1.10 "Contraintes de nom" de [IETF RFC 5280]) pour imposer des contraintes aux identificateurs uniques mondialement d'instance matérielle des certificats de dispositif subséquents dans un chemin de certification.

EXEMPLE: les contraintes de nom sont définies selon des sous-arbres répertoriant des noms autorisés ou interdits. Les sous-arbres d'un espace de noms d'un identificateur de dispositif M2M fondé sur un identificateur d'objet sont représentés par un champ otherName comprenant les éléments suivants:

– "type-ID", qui est égal à l'arc d'identification du dispositif M2M (§ H.2.1 "Identificateur de dispositif M2M" de [ITU-T Y.4500.1]) de l'espace de noms de l'identificateur de dispositif M2M fondé sur l'identificateur d'un objet correspondant; et

– "value", qui est égal au reste de l'identificateur d'objet qui identifie le sous-arbre.

10.1.1.5 Profil des certificats AE-ID et de leurs chaînes de certificats

Les certificats AE-ID et tout autre certificat de la chaîne de certificats correspondante doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

La représentation URI complète de l'identificateur AE-ID doit être incluse dans l'extension subjectAltName.

Le certificat utilisé pour signer le certificat AE-ID doit inclure les contraintes de nom nameConstraints auxquelles est conforme le nom d'hôte de la représentation URI complète de l'AE-ID.

Les certificats AE-ID ne doivent contenir aucun caractère générique.

10.1.1.6 Profil des certificats de nom de domaine complet (FQDN) et de leurs chaînes de certificats

Les certificats FQDN et tout autre certificat de la chaîne de certificats correspondante doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

Un certificat FQDN doit inclure le nom de domaine complet de la fonction d'inscription M2M sujet contenu dans l'extension subjectAltName.

Les certificats FQDN ne doivent contenir aucun caractère générique.

10.1.1.7 Profil des certificats CSE-ID et de leurs chaînes de certificats

Les certificats CSE-ID et tout autre certificat de la chaîne de certificats correspondante doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

L'extension subjectAltName doit inclure la représentation du nom de domaine public de l'identificateur CSE-ID, telle qu'elle est définie dans la Recommandation [UIT-T Y.4500.1].

Les certificats CSE-ID ne doivent contenir aucun caractère générique.

10.1.1.8 Profil des certificats d'identificateur de nœud (Node-ID) et de leurs chaînes de certificats

Les certificats Node-ID et tout autre certificat de la chaîne de certificats correspondante doivent être conformes au § 10.1.1.3 (Détails communs aux certificats dotés de chaînes de certificats).

L'extension subjectAltName doit inclure l'identificateur Node-ID, tel qu'il est défini dans la Recommandation [UIT-T Y.4500.1].

Les certificats Node-ID ne doivent contenir aucun caractère générique.

10.1.2 Identificateur de clé publique

L'identificateur de clé publique d'un certificat de clé publique brute doit être calculé comme indiqué dans [IETF RFC 7507] et [IETF RFC 6920] au moyen de l'algorithme de hachage SHA-256. L'identificateur de clé publique doit être généré à l'aide de l'un des algorithmes de hachage suivants détaillés dans [IETF RFC 6920]: SHA-256-120, SHA-256-128 ou SHA-256.

Il est recommandé que l'identificateur de clé publique soit aussi long que possible compte tenu des contraintes de déploiement.

L'identificateur de clé publique fiable (reçu pendant la configuration de l'association ou pendant la configuration des instructions d'amorçage) est comparé au certificat de clé publique brute (reçu pendant la prise de contact de sécurité) au moyen de la procédure suivante:

Une valeur de résumé de vérification est calculée conformément à la section 2 de [IETF RFC 6920] au moyen de l'algorithme de hachage identifié dans l'identificateur de clé publique de confiance.

La valeur de résumé de vérification est comparée à la valeur de résumé codée dans l'identificateur de clé publique de confiance. Si les valeurs sont identiques, le certificat de clé publique brute correspond alors à l'identificateur de clé publique de confiance. Sinon, le certificat de clé publique brute ne correspond pas à l'identificateur de clé publique de confiance.

10.1.3 Exigences relatives à la prise en charge de chaque variante de certificat de clé publique

Le Tableau 10.1.3 énumère, pour chacun des divers types d'entités (entité CSE du domaine de terrain, entité AE du domaine de terrain, entité IN-CSE, entité IN-AE, fonction d'authentification M2M et fonction d'inscription M2M), la variante d'un certificat qui peut être émis à destination de l'entité et la variante des certificats d'autres entités que ladite entité doit pouvoir traiter. Dans ce tableau, la lettre "F" signifie "facultatif", la lettre "O" signifie "obligatoire", l'expression "CA" indique que l'option est nécessaire si l'entité prend en charge le cadre d'établissement d'association de sécurité fondé sur les certificats. La mention "CB", quant à elle, indique que l'option dépend de la prise en charge, par l'entité, du cadre de configuration à distance de la sécurité fondé sur les certificats.

Tableau 10.1.3-1 – Applicabilité des variantes de certificats émis à destination d'une entité et des variantes d'autres certificats d'autres entités que l'entité doit pouvoir traiter

Entité	Variante de certificat qui peut être émis à destination d'une entité					Variante des certificats d'une autre entité que l'entité doit pouvoir traiter				
	Brut	Dispositif	CSE-ID	AE-ID	FQDN	Brut	Dispositif	CSE-ID	AE-ID	FQDN
CSE du domaine de terrain	F	F	F	–	–	CA	CA	CA	CA	CB
AE du domaine de terrain	F	F	–	F	–	CA	CA	CA	–	CB
IN-CSE	F	–	F	–	–	CA	CA	CA	CA	–

Tableau 10.1.3-1 – Applicabilité des variantes de certificats émis à destination d'une entité et des variantes d'autres certificats d'autres entités que l'entité doit pouvoir traiter

Entité	Variante de certificat qui peut être émis à destination d'une entité					Variante des certificats d'une autre entité que l'entité doit pouvoir traiter				
	Brut	Dispositif	CSE-ID	AE-ID	FQDN	Brut	Dispositif	CSE-ID	AE-ID	FQDN
IN-AE	F	–	–	F	–	CA	–	CA	–	–
MAF	–	–	–	–	O	–	–	–	–	O
MEF	–	–	–	–	O	CB	CB	–	–	O

L'authentification mutuelle entre les serveurs de gestion à distance et les clients de gestion à distance n'est pas abordée dans la présente Recommandation. Lorsqu'elle est prise en charge, l'administration à distance de la sécurité peut être utilisée pour fournir les certificats.

10.1.4 Profil des demandes de signature de certificats

Une demande de signature de certificat (CSR) est un objet signé fourni au serveur de configuration de certificat (serveur EST ou SCEP) pour demander l'émission d'un certificat. La configuration de certificat telle que définie au § 8.3.6 peut être utilisée pour émettre un certificat à destination d'un nœud, d'une entité CSE ou d'une entité AE. La demande de signature de certificat doit comporter l'information de clé publique sujet (subjectPublicKeyInfo) suivante: la clé publique et l'algorithme d'utilisation de la clé.

subjectAltName: ce champ doit contenir l'identificateur AE-ID, CSE-ID ou Node-ID qui utilise le type de nom défini pour chaque type de certificat aux paragraphes 10.1.1.5, 10.1.1.7 et 10.1.1.8.

La demande de signature de certificat peut inclure des champs et des extensions supplémentaires fournis par le serveur de fourniture de certificat, par exemple en utilisant la demande d'attributs de demande de signature de certificat (CSR) EST décrite au § 2.6 de [IETF RFC 7030].

10.2 Détails concernant les protocoles TLS et DTLS

10.2.1 Versions des protocoles TLS et DTLS

Si les charges utiles TCP doivent être sécurisées, le protocole TLS v1.2 [IETF RFC 5246] doit être utilisé.

Si les charges utiles UDP doivent être sécurisées, le protocole DTLS v1.2 [IETF RFC 6347] doit être utilisé. Il convient toutefois de noter que les systèmes cryptographiques du protocole DTLS v1.2 sont identiques à ceux du protocole TLS v1.2.

Toutes les mises en œuvre doivent prendre en charge l'indication de nom de serveur (SNI) pour indiquer leur autorité dans le champ HostName de la SNI, tel que cela est défini au paragraphe 3 de [IETF RFC 6066]. Ceci est nécessaire pour qu'un hôte qui agit en tant que serveur virtuel pour plusieurs autorités sache quelles clés doivent être utilisées pour la session TLS ou DTLS, lorsqu'il reçoit une nouvelle connexion TLS ou DTLS.

Les clients (D)TLS d'un nœud quelconque et les serveurs (D)TLS des nœuds intermédiaires prennent en charge au moins l'un des systèmes cryptographiques TLS indiqués au § 10.2.2 (Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité TLS-PSK) ou au § 10.2.3 (Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité fondés sur les certificats).

NOTE – Les serveurs (D)TLS du nœud intermédiaire doivent pouvoir prendre en charge les systèmes cryptographiques TLS pour les clients (D)TLS avec lesquels ils peuvent être amenés à interagir.

Les serveurs (D)TLS des nœuds d'infrastructure prennent en charge tous les systèmes cryptographiques TLS indiqués au § 10.2.2 (Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité TLS-PSK) ou au § 10.2.3 (Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité fondés sur les certificats).

10.2.2 Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité TLS-PSK

Les cadres de sécurité suivants:

– cadre d'établissement d'association de sécurité de clé symétrique fournie;

– cadre d'établissement d'association de sécurité fondé sur la fonction MAF;

– cadre de configuration à distance de la sécurité de clé prépartagée;

– cadre de configuration à distance de la sécurité fondé sur l'architecture GBA;

doivent utiliser l'un des algorithmes d'échange de clés définis dans [IETF RFC 4279].

Les mises en œuvre du protocole TLS dans les entités qui prennent en charge ces cadres de sécurité doivent mettre en œuvre au moins le système cryptographique TLS suivant:

– TLS_PSK_WITH_AES_128_CBC_SHA256 ([IETF RFC 5487]).

Les mises en œuvre du protocole DTLS qui prennent en charge ces cadres de sécurité doivent mettre en œuvre au moins le système cryptographique TLS suivant:

– TLS_PSK_WITH_AES_128_CCM_8 ([IETF RFC 6655]).

Les considérations relatives à la sécurité exposées dans la section 7 de [IETF RFC 4279] s'appliquent. Les applications doivent notamment déterminer si elles doivent garantir une confidentialité totale vers l'avant (PFS) et sélectionner un système cryptographique adapté (section 7.1 de [IETF RFC 4279]).

10.2.3 Systèmes cryptographiques TLS et DTLS destinés aux cadres de sécurité fondés sur les certificats

Les cadres de sécurité suivants:

– Cadre d'établissement d'association de sécurité fondé sur les certificats;

– Cadre d'amorçage de sécurité fondé sur les certificats;

doivent utiliser la prise de contact TLS standard ([IETF RFC 5246]) à l'aide du système d'échange de clés ECDHE_ECDSA ([IETF RFC 4492]).

Les mises en œuvre du protocole TLS qui prennent en charge ces cadres de sécurité doivent mettre en œuvre au moins le système cryptographique suivant:

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, [IETF RFC 5289].

Les mises en œuvre du protocole DTLS qui prennent en charge ces cadres de sécurité doivent mettre en œuvre au moins le système cryptographique TLS suivant:

TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, [IETF RFC 7251].

Les mises en œuvre prenant en charge ces cadres de sécurité doivent prendre en charge l'authentification d'autres entités à l'aide de tous les certificats de clé publique disponibles (voir le § 8.1.2.1 "Variantes de certificats de clé publique"):

Certificat de clé publique brute: la mise en œuvre doit prendre en charge la réception et le traitement des clés publiques brutes conformément à la section 9.1.3.2 "Certificats de clé publique brute" de [b-IETF RFC 7252], à l'aide du mécanisme défini dans [IETF RFC 7250].

Tous les autres certificats, c'est-à-dire ceux qui sont définis dans la Recommandation ITU-T X.509 et qui comprennent un identificateur de dispositif: la mise en œuvre doit prendre en charge la réception et le traitement des clés publiques brutes conformément à la section 9.1.3.3 "Certificats de clé publique brute" de [b-IETF RFC 7252].

10.3 Détails concernant l'exportation et le calcul de clés

10.3.1 Détails concernant l'exportation de clés à l'aide du protocole TLS

Détails concernant l'exportation de clés d'inscription par protocole TLS

Après l'authentification TLS réussie entre l'entité inscrite et la fonction d'inscription M2M (voir le § 8.3.1.2), la clé d'inscription (Ke) et l'identificateur RelativeKeID sont générés à partir des secrets de session (D)TLS par l'entité inscrite et la fonction d'inscription M2M en exportant des clés TLS ([IETF RFC 5705]) à l'aide du libellé "EXPORTER-oneM2M-Bootstrap" et du paramètre "length" égal à 48. La valeur des 32 octets de plus faible poids est attribuée à la clé d'inscription (Ke), tandis que la valeur des 16 octets de plus fort poids est attribuée à l'identificateur RelativeKeID.

Détails concernant l'exportation de clés TLS de connexion sécurisée M2M

Après l'authentification TLS réussie entre l'entité A et la fonction d'authentification M2M (MAF; voir le § 8.8.2.7), la clé de connexion sécurisée M2L (Kc) et son identificateur (KcID) sont générés à partir des secrets de session (D)TLS par l'entité A et la fonction MAF en exportant des clés TLS ([IETF RFC 5705]) à l'aide du libellé "EXPORTER-oneM2M-Connection" et du paramètre "length" égal à 48. La valeur des 32 octets de plus faible poids est attribuée à la clé de connexion sécurisée M2M (Kc), tandis que la valeur des 16 octets de plus fort poids est attribuée à son identificateur (KcID).

Détails concernant l'exportation de clés pairwiseE2EKey par protocole TLS

Une fois l'authentification TLS réussie entre l'extrémité initiatrice ESCertKE et l'extrémité de destination ESCertKE (voir le § 8.7.2.2), la clé pairwiseE2EKey et son identificateur pairwiseE2EKeyID sont générés à partir des secrets de session (D)TLS par l'entité inscrite et la fonction d'inscription M2M en exportant des clés à l'aide du protocole TLS ([IETF RFC 5705]), au moyen du libellé "EXPORTER-oneM2M-ESCertKE" et du paramètre "length" égal à 48. La valeur des 32 octets de plus faible poids est attribuée à la clé pairwiseE2EKeyID, tandis que la valeur des 16 octets de plus fort poids est attribuée à son identificateur (pairwiseE2EKeyID).

10.3.2 Calcul du justificatif d'identité principal à partir d'une clé d'inscription

Le présent paragraphe décrit en détail la création d'un justificatif d'identité principal (Km) à partir d'une clé d'inscription (Ke) dans les cadres d'amorçage de sécurité.

Les informations suivantes doivent être utilisées lors de la génération du justificatif d'identité Km à partir de la clé Ke:

– la valeur de la clé d'inscription (Ke);

– l'identificateur de la fonction d'authentification M2M (MAF-ID) doit être encodé en une chaîne d'octets conformément aux règles d'encodage UTF-8 indiquées dans [IETF RFC 3629] et appliquer la forme de normalisation KC (NFKC) telle qu'elle est définie dans [Unicode].

La valeur du justificatif d'identité Km doit être générée comme suit:

Km: = HMAC-SHA-256(Ke, "oneM2M Enrolment Key to master credential derivation" || MAF-ID)

où HMAC-SHA-256 est défini dans [IETF RFC 2104].

10.3.3 Calcul de la clé de connexion sécurisée fournie à partir d'une clé d'inscription

Le présent paragraphe décrit en détail l'établissement d'une clé de connexion sécurisée configurée (Kpsa) à partir d'une clé d'inscription (Ke) dans les cadres de configuration à distance.

Les informations suivantes doivent être utilisées lors de la génération du justificatif d'identité Kpsa à partir de la clé Ke:

– La valeur de la clé d'inscription (Ke).

– L'identificateur CSE-ID ou AE-ID de l'entité inscrite B doit être encodé en une chaîne d'octets conformément aux règles d'encodage UTF-8 indiquées dans [IETF RFC 3629] et appliquer la forme de normalisation KC (NFKC) telle qu'elle est définie dans [Unicode].

La valeur du justificatif d'identité Kpsa doit être générée comme suit:

Kpsa: = HMAC-SHA-256(Ke, "oneM2M enrolment key to provisioned secure connection key derivation" || Enrolee-B-ID);

où HMAC-SHA-256 est défini dans [IETF RFC 2104].

10.3.4 Génération de l'identificateur KeID

La valeur KeID devra être formée comme suit:

KeID = base64encode(RelativeKeID)@MEF_FQDN

où:

base64encode(RelativeKeID) indique l'encodage base64 ([IETF RFC 3548]) de la valeur de l'identificateur RelativeKeID; et

MEF_FQDN se réfère au nom de domaine complet de la fonction d'inscription M2M.

10.3.5 Génération d'identificateur de clé pour le cadre de sécurité MAF

La valeur de l'identificateur de clé devra être formée comme suit:

Identificateur de clé = base64encode(RelativeKeyID)@MAF_FQDN

où:

base64encode(RelativeKeID) indique l'encodage base64 ([IETF RFC 3548]) de la valeur de l'identificateur RelativeKeID; et

MEF_FQDN se réfère au nom de domaine complet de la fonction d'authentification M2M.

10.3.6 Calcul de la clé principale de bout en bout à partir d'une clé de connexion sécurisée fournie

10.3.6.1 Introduction

Le présent paragraphe décrit en détail la génération d'une clé principale de bout en bout (Ke2e_master) fondée sur l'établissement réussi d'une association de sécurité entre un point d'extrémité ESF source et un point d'extrémité ESF cible au moyen d'un cadre de configuration à distance de la sécurité, tel que décrit au § 8.3. Les mécanismes permettant de générer la clé principale de bout en bout recourent ensuite à un processus d'extraction de clés au moyen de la clé de connexion sécurisée fournie (Kpsa).

Les informations suivantes doivent être utilisées lors de la génération de la clé Ke2e à partir de Kpsa:

– La valeur de la clé de connexion sécurisée fournie (Kpsa).

– L'identificateur CSE-ID ou AE-ID du point d'extrémité ESF source B doit être encodé en une chaîne d'octets conformément aux règles d'encodage UTF-8 indiquées dans [IETF RFC 3629] et appliquer la forme de normalisation KC (NFKC) telle qu'elle est définie dans [Unicode].

La valeur de la clé Ke2e_master doit être générée comme suit:

Ke2e_master = HMAC-Hash (Salt, Kpsa).

NOTE – Dans le cas de justificatifs d'identité générés par la source, une valeur aléatoire générée par le point d'extrémité ESF source est utilisée au lieu de la valeur Kpsa pour générer la clé Ke2e_master.

10.3.6.2 Extraction de clés et extension de clé principale de bout en bout

La clé principale de bout en bout (Ke2e_master) sert à générer les clés spécifiques de protection de la sécurité. Les paramètres d'extraction et d'extension de clés ainsi que la portée sont utilisés pour générer les diverses clés. Le processus d'extraction et d'extension de clés est effectué conformément aux spécifications de [IETF RFC 5869]. Le Tableau 10.3.6.2-1 dresse une liste de clés de bout en bout possibles.

Tableau 10.3.6.2-1 – Clés de sécurité de bout en bout

Protection de la sécurité	Clés symétriques générées
Authenticité du message (primitive)	Ke2e_msg_auth
Confidentialité du message (primitive)	Ke2e_msg_conf
Intégrité des données (attribut)	Ke2e_att_auth
Confidentialité des données (attribut)	Ke2e_att_conf

Les clés de protection de la sécurité de bout en bout sont produites par l'extension de la clé Ke2e_master au moyen des mécanismes indiqués dans [IETF RFC 5869]. À l'aide de la clé principale de bout en bout générée, les clés associées d'authentification de message de bout en bout et/ou de vérification de la confidentialité des messages de bout en bout ainsi que les clés d'attribut sont générées de la manière suivante:

– T(0) = chaîne de caractères vide (longueur nulle)

– Clé d'authenticité de message de bout en bout (Ke2e_msg_auth) = T(1) = HMAC-Hash (Ke2e_master, T(0) | "E2E Message Authentication Key"| 0x01)

– Clé de vérification de la confidentialité des messages de bout en bout Confidentialité du message Key (Ke2e_msg_conf) = T(2) = HMAC-Hash (Ke2e_master, T(1)| "E2E Message Confidentiality Key"|0x02)

– Clé d'authenticité d'attribut de bout en bout (Ke2e_att_auth) = T(3) = HMAC-Hash (Ke2e_master, T(2)| "E2E Attribute Authenticity Key"|0x03)

– Clé de vérification de la confidentialité des attributs de bout en bout (Ke2e_att_conf) = T(4) = HMAC-Hash(Ke2e_master, T(3)| "E2E Attribute Confidentiality Key"|0x04)

NOTE 1 – Si des algorithmes AEAD sont utilisés, lorsqu'une seule clé est utilisée, on peut alors calculer la clé Ke2e_msg_auth ou Ke2e_msg_conf et l'utiliser pour la vérification de l'authenticité et de la confidentialité du message.

NOTE 2 – Le point d'extrémité ESF cible peut recevoir toutes les clés requises ou uniquement la clé principale de bout en bout (Ke2e_master) et les paramètres cryptographiques associés (par exemple, des libellés ou des valeurs aléatoires) qui sont ensuite utilisés par le point d'extrémité ESF cible afin de générer les clés requises pour les cadres ESPrim et ESData.

10.3.7 Calcul des clés symétriques présentant des contraintes d'utilisation à partir d'une clé d'inscription

Le présent paragraphe décrit en détail la création d'une clé symétrique à contraintes d'utilisation à partir d'une clé d'inscription (Ke) dans les cadres de configuration à distance de la sécurité.

Les informations suivantes doivent être utilisées:

– la valeur de la clé d'inscription (Ke);

– l'identificateur d'utilisation de sécurité (SUID) approprié à l'utilisation de la clé symétrique;

– l'identificateur de l'entité inscrite cible, qui est un nom de domaine complet devant être encodé en une chaîne d'octets conformément aux règles d'encodage UTF-8 indiquées dans [IETF RFC 3629] et appliquer la forme de normalisation KC (NFKC) telle qu'elle est définie dans [Unicode].

Si la cible d'inscription est une entité de services communs ou entité d'application, on utilisera la représentation du nom de domaine complet de l'identificateur CSE ou AE absolu.

La valeur de la clé symétrique à contraintes d'utilisation doit être générée comme suit:

HMAC-SHA-256(Ke, "oneM2M enrolment key to usage-constrained symmetric key derivation" || SUID || Enrolment-Target-ID);

où HMAC-SHA-256 est défini dans [IETF RFC 2104].

10.3.8 Algorithmes de calcul de la clé sessionESPrimKey

10.3.8.1 Introduction

Le paramètre sessionESPrimKey est utilisé pour la sécurité de bout en bout des primitives (ESPrim) et est calculé à partir des paramètres pairiseESPrimKey, receiveESPrimRandObject et originatorESPrimRandObject (voir le § 8.4.2).

Le paragraphe 10.3.8 décrit les algorithmes permettant de calculer la clé sessionESPrimKey utilisée dans le cadre ESPrim. L'algorithme disponible est répertorié dans le Tableau 10.3.8.1-1.

Tableau 10.3.8.1-1 – Algorithme de calcul de la clé sessionESPrimKey

Algorithme	Obligatoire/facultatif	Paragraphe
HMAC-SHA256	O	10.3.8.2

10.3.8.2 Algorithme HMAC-SHA256 de calcul de la clé sessionESPrimKey

La clé sessionESPrimKey est calculée comme suit:

sessionESPrimKey = HMAC-SHA256 (pairwiseESPrimKey,
receiverESPrimRandObject || originatorESPrimRandObject ||
"oneM2M HMAC-SHA256 sessionESPrimKey derivation algorithm"),

où HMAC-SHA-256 est défini dans [IETF RFC 2104].

10.4 Détails concernant l'identificateur de justificatif d'identité

L'identificateur de justificatif d'identité contient deux parties:

– Un identificateur de type, qui est un nombre entier positif défini par le type de données sec:credIDTypeID.

– Une valeur, qui contient l'identificateur unique mondialement du justificatif d'identité de l'entité. Cette valeur peut contenir des caractères latins, des chiffres, des points ("."), des tirets bas ("_"), des tirets ("-") et des arobases ("@").

L'identificateur de justificatif d'identité est formé en joignant le type et la valeur par un tiret ("-").

NOTE – Un identificateur de justificatif d'identité est un identificateur unique mondialement servant à identifier les ressources *serviceSubscribedAppRule* ([UIT-T Y.4500.1]) et les justificatifs d'identité contenus dans les informations de configuration de la sécurité.

10.5 KpsaID

L'identificateur KpsaID doit être formé de la manière suivante:

KpsaID = Issuer_Relative_KpsaID@Issuer_FQDN;

où:

L'élément Issuer_Relative_KpsaID est composé de caractères latins, de chiffres, de points ("."), de tirets bas ("_") et des tirets ("-"). L'émetteur de l'identificateur KpsaID doit s'assurer que deux justificatifs d'identité Kpsa ne partagent pas le même élément Issuer_Relative_KpsaID.

L'élément Issuer_FQDN est un nom de domaine complet qui représente la partie prenante qui a fourni le justificatif d'identité Kpsa.

NOTE – Le format de la clé KpsaID permet d'extraire l'identité de l'émetteur à partir de l'identificateur KpsaID.

10.6 Format de l'identificateur KmID

L'identificateur KmID doit être formé de la manière suivante:

KmID = MAF_RELATIVE_KmID@MAF_FQDN;

où:

L'élément MAF_RELATIVE_ KmID est composé de caractères latins, de chiffres, de points ("."), de tirets bas ("_") et des tirets ("-"). Les majuscules et minuscules peuvent être utilisées indifféremment pour l'élément MAF_RELATIVE_KmID. La fonction MAF doit s'assurer que deux justificatifs d'identité Km ne partagent pas le même élément MAF_RELATIVE_ KmID.

MEF_FQDN se réfère au nom de domaine complet de la fonction d'authentification M2M.

NOTE – Le format de l'identificateur KpsaID permet d'extraire l'identité de la fonction d'authentification M2M à partir de l'identificateur KmID.

10.7 Expiration de l'inscription

L'expiration de l'inscription est la durée de vie à appliquer à la clé générée, c'est-à-dire la clé Ke faisant partie du cadre de configuration à distance de la sécurité des clés symétriques préconfigurées. Les clés qui sont générées pour établir des associations de sécurité entre les entités inscrites et les cibles d'inscription (c'est-à-dire Km ou Kpsa) à partir de la clé d'inscription Ke ne seront pas valides après l'expiration de la durée de vie du justificatif d'inscription Ke. Par conséquent, la durée de vie de Km ou de Kpsa devrait être égale, au maximum, à la durée de vie associée à Ke. Une fois l'inscription expirée, l'entité inscrite doit réinitialiser la configuration à distance pour régénérer les clés, au moyen des cadres de configuration à distance de la sécurité, décrits au § 8.3.2.1.

11 Architecture de protection de la confidentialité à l'aide du gestionnaire de politique de confidentialité (PPM)

11.1 Introduction

Le présent paragraphe décrit une architecture pour le gestionnaire de politique de confidentialité (PPM). Ce dernier est une architecture de protection de la confidentialité des autorisations répartie qui utilise les préférences de l'utilisateur en matière de confidentialité.

Il s'agit également d'un cadre de gestion des données personnelles fondé sur ces préférences. Il crée des politiques de contrôle d'accès à partir des préférences de l'utilisateur en matière de confidentialité et protège les informations d'identification personnelle de celui-ci contre tout tiers non autorisé. Il peut être utilisé par un fournisseur de services M2M ou par une autre partie prenante agissant comme un tiers de confiance. Si le fournisseur de services ou toute autre partie prenante fournit à un tiers des informations d'identification personnelle concernant l'utilisateur, ce dernier doit donner son accord. Si l'utilisateur accepte une politique de confidentialité qui indique la fourniture à un tiers, le fournisseur de services peut fournir à un tiers les informations d'identification personnelle. Dans le cas contraire, le fournisseur de services doit mettre à jour la politique de confidentialité et obtenir l'accord de l'utilisateur.

11.2 Relation entre les composants du gestionnaire de politique de confidentialité et oneM2M

Le PPM doit comprendre les composants suivants. Chaque composant est décrit en détail dans le document oneM2M TR-0001-Use_Cases_Collection [b-oneM2M TR0002]. Le présent paragraphe décrit la relation entre lesdits composants et les composants de oneM2M:

Mécanisme de consentement sophistiqué pour la politique de confidentialité:

Lorsqu'un utilisateur final est abonné à un service qui utilise une application fournie par une entité IN-AE, celui-ci devient un sujet de données qui crée une préférence en matière de confidentialité et l'enregistre dans le PPM.

Cette fonction est décrite au § 11.4.1.2.

Fonctions du point de décision de politiques (PDP) et du point de stockage de politiques (PRP):

PDP:

Lorsqu'une application demande des données personnelles à une entité IN-CSE, le PPM répond par une décision d'accès qui est déterminée selon les politiques de contrôle d'accès fondées sur les préférences de l'utilisateur en matière de confidentialité.

PRP:

Lorsqu'une application demande des données personnelles à une entité IN-CSE, le PDP demande des politiques de contrôle d'accès au PPM. Le PPM fournit les politiques de contrôle d'accès fondées sur les préférences de l'utilisateur en matière de confidentialité.

Les composants du PDP et du PRP sont définis aux paragraphes 6.2.2 (Autorisation) et 7.1 (Mécanisme de contrôle d'accès).

Traçabilité de l'utilisation des données personnelles:

Le PPM doit stocker le journal d'accès, qui enregistre les types de données collectées auxquels ont eu accès les entités IN-AE.

Cette fonction doit faire l'objet d'un complément d'étude sur oneM2M, mais elle peut être mise en œuvre en utilisant des composants définis dans les normes oneM2M.

11.3 Gestion de la politique de confidentialité dans l'architecture oneM2M

11.3.1 Introduction

L'utilisation du PPM implique l'exécution de quatre procédures. Le présent paragraphe explique les relations entre les étapes du scénario PPM et les composantes de oneM2M:

– Un sujet de données rejoint une entité IN-CSE, telle qu'une plate-forme M2M.
– Un sujet de données s'abonne au service proposé par l'entité IN-AE.
– Une entité IN-AE demande les données personnelles stockées sur une entité ASN-CSE, MN-CSE ou IN-CSE.

Le sujet des données vérifie le journal d'accès à ses données personnelles et demande leur suppression à l'entité IN-AE.

11.3.2 Entités concernées

Sujet des données:

Un utilisateur final peut utiliser les services d'une entité IN-CSE en s'abonnant à un service d'une entité IN-AE qui fournit des fonctions de contrôle de l'accès aux informations de l'entité IN-CSE.

Lorsqu'un utilisateur final est abonné à un service proposé par une entité IN-AE, celui-ci devient le sujet des données.

Entité d'application:

Une entité d'application collecte différents types de données, comme les entrées de détecteurs.

Elle envoie ensuite ces données à une entité ASN-CSE, MN-CSE ou IN-CSE.

Entité ASN-CSE ou MN-CSE:

Point d'application de politique (PEP):

Le PEP constitue l'une des fonctions de l'entité de services communs.

Point de décision de politique (PDP):

Le PDP constitue l'une des fonctions de l'entité de services communs.

Données personnelles:

Les données personnelles sont des informations qui peuvent être utilisées individuellement ou avec d'autres informations pour identifier une personne et former des informations d'identification personnelle.

Une entité de services communs déployée sur le terrain collecte et stocke des données personnelles.

Exemples de données personnelles: données fournies par des détecteurs, consommation électrique, état de fonctionnement d'un climatiseur, etc.

IN-AE:

– Une entité IN-AE fournit des services à un utilisateur final qui rejoint une entité IN-CSE.
– Une entité IN-AE demande les données personnelles à une IN-CSE afin de fournir ses services.

IN-CSE:

Portail M2M:

Un portail M2M est une sorte de site Web ou d'application Web qui gère les services fournis par une plate-forme M2M (entité IN-CSE et entités IN-AE associées).

Un utilisateur final accède à un portail pour rejoindre une entité IN-CSE. Un sujet de données accède au portail pour s'abonner à un service proposé par l'entité IN-AE.

PPM:

Le PPM stocke des politiques de contrôle d'accès fondées sur les préférences de l'utilisateur en matière de confidentialité.

Étant donné qu'une entité MN-CSE utilise le PPM en tant que PDP ou PRP, le PPM doit pouvoir prendre en charge les fonctionnalités du PDP et du PRP:

Un portail PPM est une sorte de site Web ou d'application Web disponible dans le PPM.

Un utilisateur final accède à un portail PPM pour configurer ses préférences en matière de confidentialité.

11.3.3 Flux de gestion dans l'architecture PPM

11.3.3.0 Introduction

Ce paragraphe décrit le cas où une entité ASN-CSE ou MN-CSE stocke des données personnelles.

11.3.3.1 Rejoindre une entité IN-CSE

Lorsqu'un sujet de données rejoint une entité IN-CSE, il configure une préférence en matière de confidentialité au moyen du gestionnaire PPM. Une préférence en matière de confidentialité précise à quels types de données les entités IN-AE sont autorisées à accéder. La Figure 11.3.3.1-1 illustre un aperçu de ce processus.

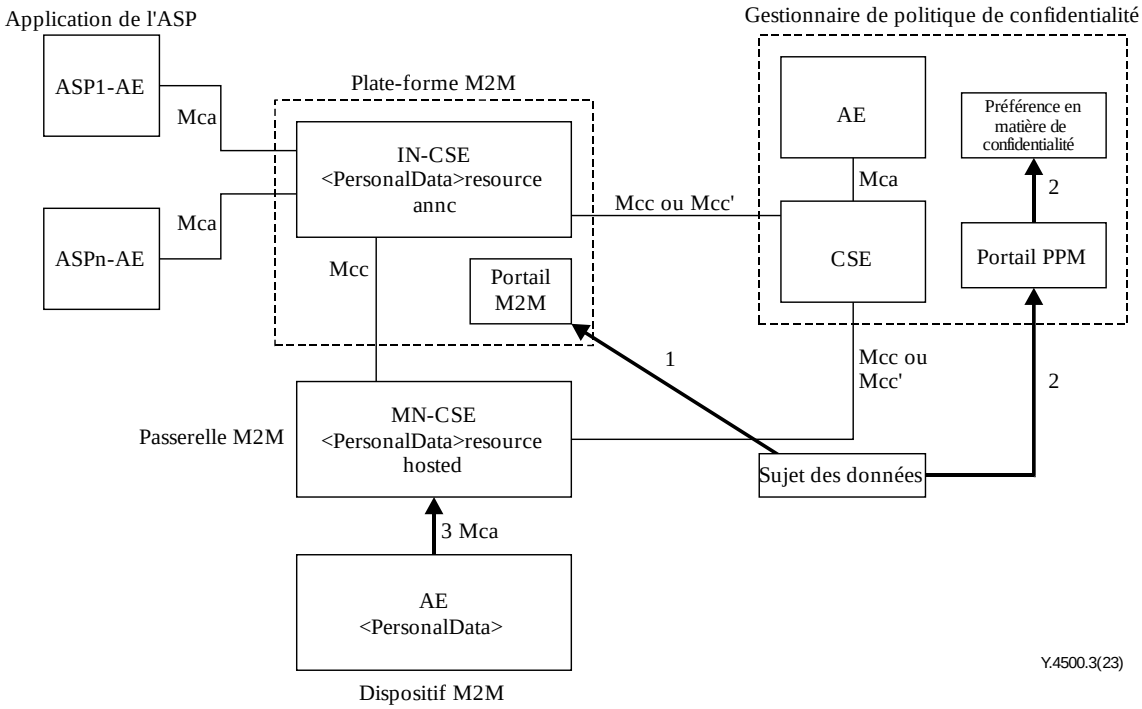


Figure 11.3.3.1-1 – Un sujet de données rejoint une entité IN-CSE

1 Un sujet de données accède au portail M2M d'une entité IN-CSE:

– Ce processus utilise généralement des protocoles d'accès au Web, tel que HTTP ou HTTPS.

– Ce processus est décrit au § 11.4.1.2.

2 Un sujet de données configure une préférence en matière de confidentialité et l'enregistre sur le portail PPM. Le PPM crée ensuite des politiques de contrôle d'accès fondées sur ces préférences:

– Un sujet de données accède au portail PPM ou le portail M2M l'y redirige. Ce processus utilise des protocoles d'accès au Web.

– Ce processus est décrit au § 11.4.1.2.

3 Une entité ASN-CSE ou MN-CSE recueille et stocke les données envoyées par l'entité d'application.

11.3.3.2 Abonnement à un service fourni par l'entité IN-AE

Le sujet des données peut s'abonner à divers types de services fournis par des entités IN-AE par l'intermédiaire de l'entité IN-CSE. Les listes de services sont enregistrées sur un portail M2M et le sujet des données peut choisir de s'abonner aux services. Lorsque le sujet des données s'abonne à un service, il doit accepter une politique de confidentialité. Pour que le sujet des données puisse facilement comprendre cette politique, le PPM doit créer une politique de confidentialité personnalisée fondée sur la politique de confidentialité fournie par l'entité IN-AE et sur les préférences de confidentialité du sujet des données. Par conséquent, le sujet des données peut contrôler les données personnelles et son accord implique la compréhension de la politique de confidentialité. La Figure 11.3.3.2-1 illustre un aperçu de ce processus.

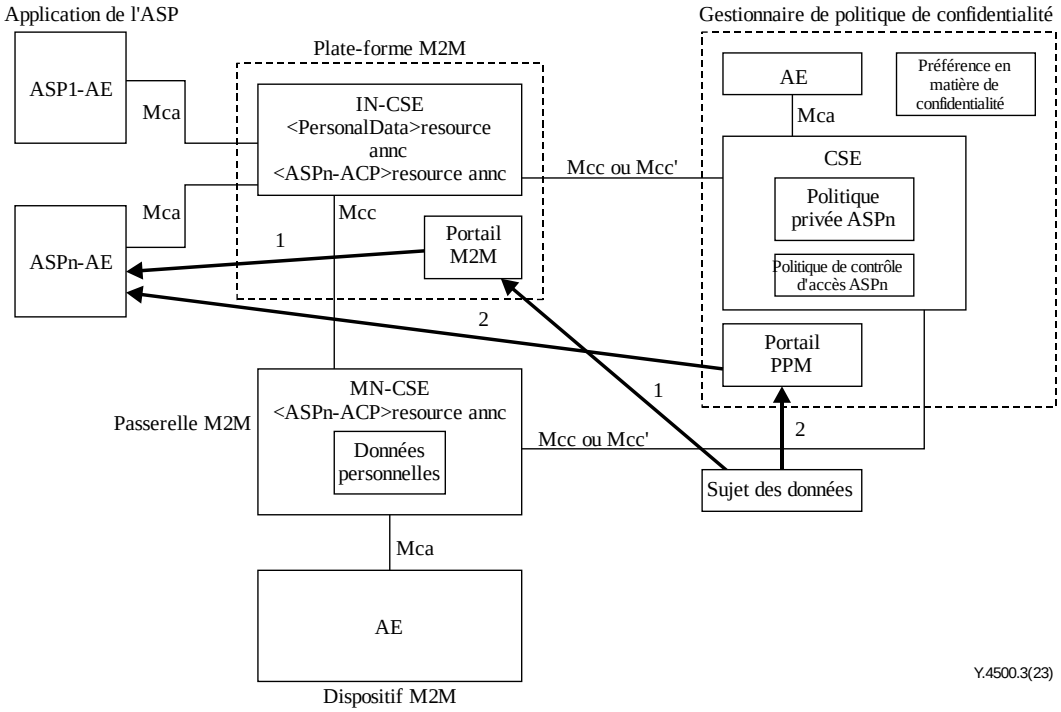


Figure 11.3.3.2-1 – Le sujet de données s'abonne à un service proposé par l'entité IN-AE

1 Le sujet de données accède au portail et sélectionne un service de l'entité IN-AE auquel il souhaite s'abonner.

Ce processus utilise généralement des protocoles d'accès au Web, tel que HTTP ou HTTPS.

Ce processus est décrit au § 11.4.1.1.

2 Le sujet des données doit accepter une politique de confidentialité pour s'abonner au service de l'entité IN-AE. Le PPM doit créer la politique de confidentialité personnalisée pour chaque sujet des données en fonction des préférences de confidentialité de ce dernier. Le sujet des données peut facilement confirmer les différences entre les préférences en matière de confidentialité et la politique de confidentialité et comprendre quel type de données personnelles est collecté par l'entité IN-AE. Une fois que le sujet des données accepte la politique de confidentialité, il peut s'abonner au service de l'IN-AE:

La fonction de création d'une politique de confidentialité personnalisée est décrite au § 11.4.1.3.

3 Le PPM doit créer ou mettre à jour des politiques de contrôle d'accès en utilisant la politique de confidentialité que le sujet des données a acceptée:

La fonction de création ou de mise à jour des politiques de contrôle d'accès dans le PPM peut s'appuyer sur les mécanismes d'autorisation exposés aux paragraphes 7.3 et 7.4. Les détails concernant le processus de synchronisation ne sont pas disponibles dans la présente Recommandation.

11.3.3.3 Demande de données personnelles à l'entité IN-CSE

Si l'entité IN-AE a besoin de données personnelles pour fournir le service, elle demande les données à l'entité IN-CSE. Le PPM fonctionne en tant que PDP ou PRP.

Si le PPM fonctionne en tant que PDP, le PEP reçoit la décision d'accès de la PPM et contrôle l'accès aux données en les utilisant. La Figure 11.3.3.3-1 illustre un aperçu de ce processus.

Si le PPM fonctionne en tant que PRP, le PDP récupère les politiques de contrôle d'accès de la PPM et contrôle l'accès aux données en les utilisant. La Figure 11.3.3.3-2 illustre un aperçu de ce processus.

A Le PPM fonctionne en tant que PDP

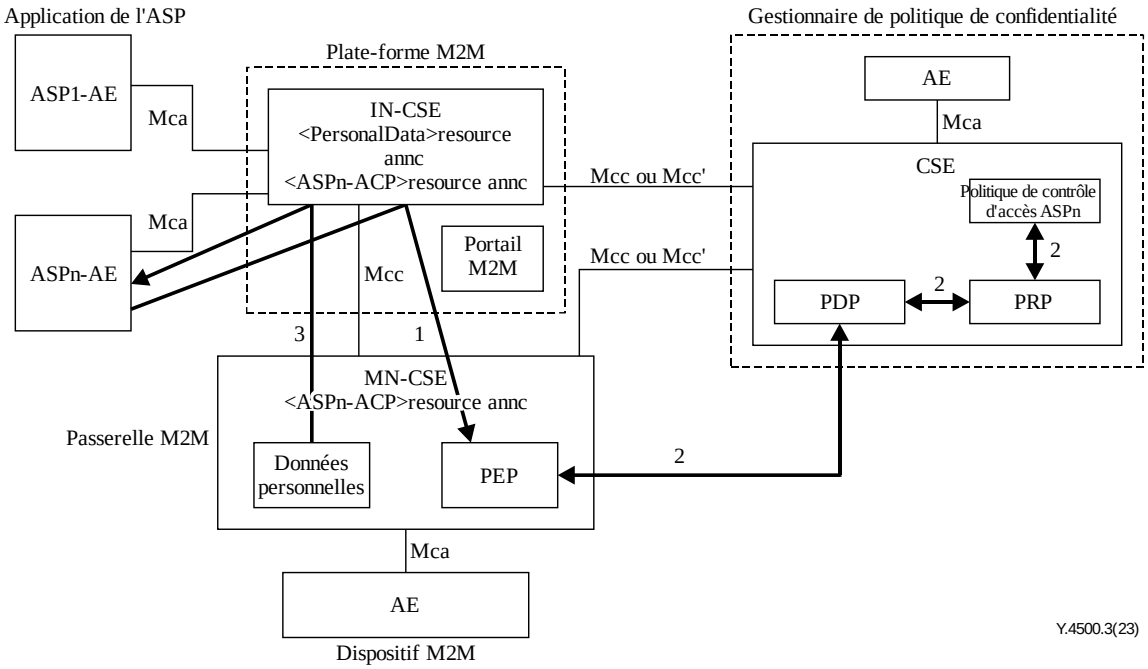


Figure 11.3.3.3-1 – Demande de données personnelles formulée à l'entité IN-CSE (le PPM fonctionne en tant que PDP)

1 Une entité IN-AE demande des données personnelles à l'entité IN-CSE.

2 Le PEP de l'entité ASN-CSE ou MN-CE demande la décision prise par le PPM. Le PPM décide d'autoriser ou de refuser l'accès aux données personnelles à l'aide des politiques de contrôles d'accès. Le PPM envoie ensuite sa décision à l'entité ASN-CSE ou MN-CSE.

Dans ce cas, la PPM doit fournir une interface qui permette de contrôler l'accès aux données personnelles en utilisant le PPM comme PDP.

3 Si l'accès à des données personnelles est autorisé, le PEP y accède et les envoie à l'entité IN-AE en tant que réponse.

B Le PPM fonctionne en tant que PRP

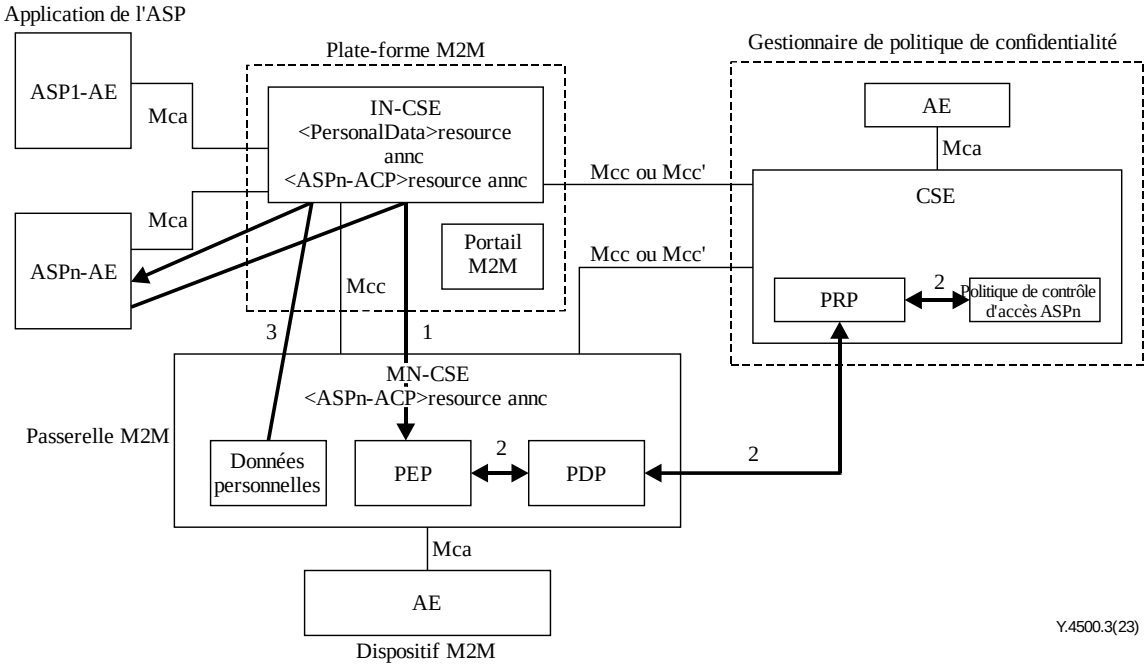


Figure 11.3.3.3-2 – Demande de données personnelles formulée à l'entité IN-CSE (le PPM fonctionne en tant que PRP)

1 Une entité IN-AE demande des données personnelles à l'entité IN-CSE.

2 Le PEP de l'entité ASN-CSE ou MN-CE demande la décision prise par le PDP. Le PDP demande les politiques au PPM. La PPM doit collecter les politiques de contrôle d'accès concernant la demande de politique et les envoyer en tant que réponse au PDP de l'entité ASN-CSE ou MN-CSE. Le PDP décide alors d'autoriser ou de refuser l'accès aux données personnelles en utilisant les politiques de contrôle d'accès et envoie un résultat comme "Réponse de décision" au PEP:

Dans ce cas, la PPM doit fournir une interface qui permette de contrôler l'accès aux données personnelles en utilisant le PPM comme PRP. Dans la version actuelle, l'autorisation dynamique directe doit être utilisée, le PPM jouant le rôle de serveur d'autorisation dynamique (DAS). Pour plus précisions, voir le § 7.3.3.2.

3 Si l'accès à des données personnelles est autorisé, le PEP y accède et les envoie à l'entité IN-AE en tant que réponse.

11.4 Modèles de mise en œuvre du gestionnaire de politique de confidentialité

11.4.1 Utilisation du langage de balisage de conditions générales

11.4.1.0 Introduction

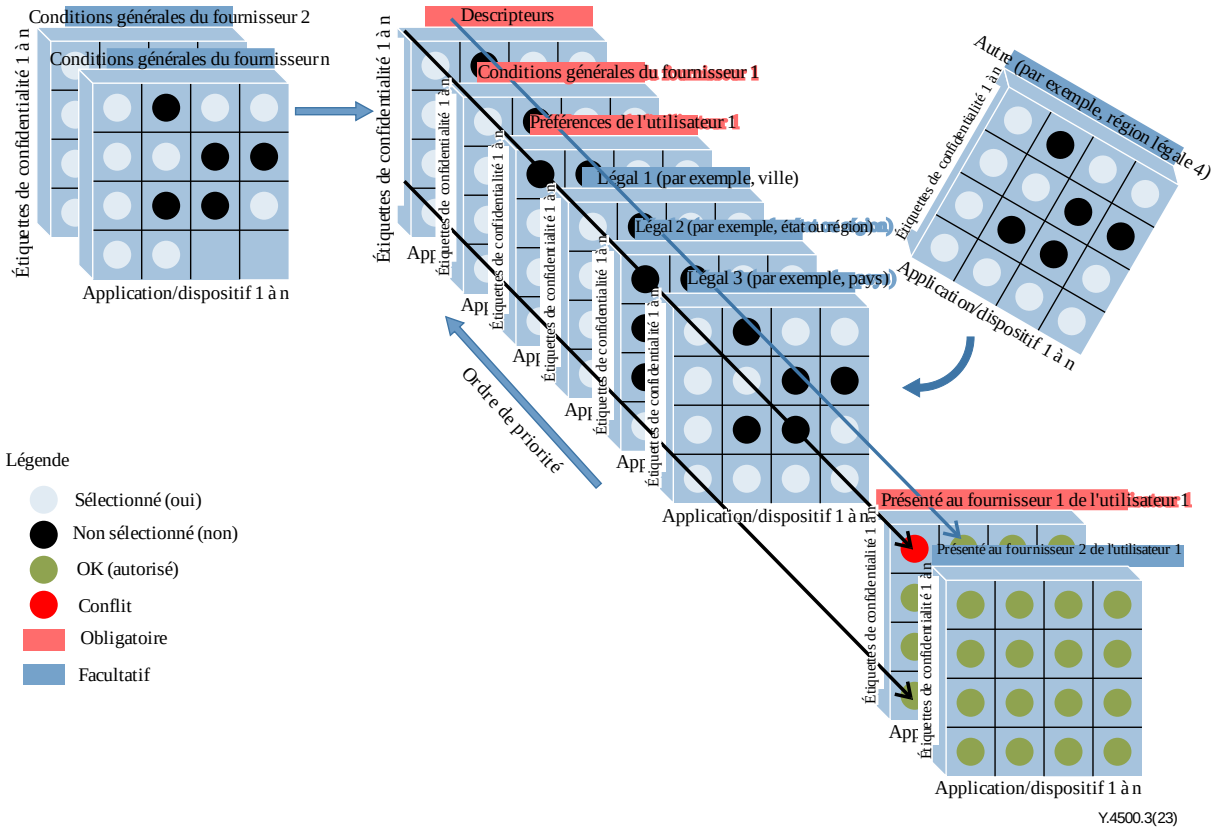


Figure 11.4.1.0-1 – Modèle de mise en œuvre du gestionnaire de politique de confidentialité utilisant le langage de balisage de conditions, pour un utilisateur final (#1) et un fournisseur de services d'application (fournisseur 1)

Le modèle représenté dans la Figure 11.4.1.0-1 décrit les composants du gestionnaire de politique de confidentialité (PPM) pour un utilisateur final (#1) et un fournisseur de services d'application (fournisseur 1), représentés sous la forme d'un nombre de filtres de sélection dans une série de trames de filtrage empilées.

Quatre trames de filtrage obligatoires sont définies:

Trame de filtrage de descripteurs.

Au moins une trame de filtrage des conditions générales du fournisseur.

Trame de filtrage des préférences de l'utilisateur.

Au moins une trame de filtrage des éléments présentés à l'utilisateur.

Chaque trame de filtrage contient une grille représentant les étiquettes de confidentialité du langage de balisage (verticalement) et les applications ou dispositifs (horizontalement).

Pour les trames de filtrage concernant les conditions générales du fournisseur et les préférences de l'utilisateur, chaque attribut représenté par l'étiquette de confidentialité configurée comme étant "sélectionné"" ou "non sélectionné" pour une application ou un dispositif est modélisé par un disque de filtrage coloré correspondant.

Les disques situés aux mêmes positions dans une ou plusieurs trames de filtrage de même structure concernant les éléments présentés à l'utilisateur détectent les filtres identiques dans la pile de trames:

– lorsque les conditions générales du fournisseur et les préférences de l'utilisateur sont compatibles, ces disques sont verts;

– en cas de conflits, les disques sont rouges.

EXEMPLE: si le fournisseur de service d'application (ASP) attend de l'utilisateur qu'il accepte la collecte des informations d'emplacement et leur partage avec un tiers, l'ASP choisit ces deux attributs (disques clairs). Si l'utilisateur final a fait savoir qu'il ne souhaite pas que les informations d'emplacement soient collectées et partagées, des disques noirs seront insérés dans la trame de filtrage des préférences de l'utilisateur et le passage dans la pile sera bloqué.

Des trames de filtres supplémentaires facultatives peuvent être placées dans la pile pour sélectionner ou non ces mêmes caractéristiques en ajoutant un disque de filtre coloré approprié. Par exemple, une demande concernant la priorité des politiques ou le pays peut se substituer au choix d'un fournisseur de service d'application ou annuler la sélection d'un utilisateur final. La position de ces trames de filtrage facultatives détermine leur priorité, celles à l'avant supplantant celles à l'arrière.

Ce modèle suppose que la grande majorité des attributs de fournisseur choisis par le fournisseur de service d'application n'entrera pas en conflit avec les préférences de l'utilisateur et qu'elle s'affichera en vert. Toutefois, il existera un très grand nombre de dispositifs, d'applications, de mises à jour de logiciels et de dispositifs remplacés. Dans la plupart des cas, cela ne traduit pas par un conflit. Toutefois, dans le cas contraire, les éléments concernés sont identifiés instantanément par un ou plusieurs disques rouges visibles uniquement par l'utilisateur final, évitant ainsi toute relecture incessante de centaines de pages de conditions générales détaillées.

Une instance de cette pile doit exister pour chaque utilisateur final enregistré auprès du PPM et pour chaque fournisseur de service d'application auquel il s'est abonné. Cependant, la trame du filtrage de descripteurs et les trames facultatives de filtrage de ville/état/pays/région peuvent être des ressources partagées pour ces instances.

11.4.1.1 Enregistrement de la politique de confidentialité du fournisseur de service d'application

L'enregistrement facultatif d'une politique de confidentialité d'application doit faire partie du processus d'obtention d'un identificateur App-ID enregistré pour chaque application et version et du processus de présentation d'un certificat de sécurité à l'autorité d'enregistrement oneM2M qui est utilisé pour authentifier l'application et la version.

Le fournisseur de services d'application téléchargera un modèle d'importation des conditions générales de l'application à partir du serveur du registre d'identificateurs App-ID oneM2M, s'il ne possède pas déjà le modèle d'importation correct.

Le modèle d'importation des conditions générales d'application doit énumérer, par ordre numérique, les étiquettes figurant dans l'Annexe normative J.

NOTE – Le format du modèle d'importation de conditions générales dépend de la mise en œuvre, à condition qu'il puisse transmettre les informations indiquées dans l'Annexe J.

Pour chaque étiquette de la liste, le fournisseur de service d'application doit fournir une valeur à tous les dispositifs et toutes les applications du domaine d'application et l'enregistrer dans le format défini à l'Annexe normative J.

Le fournisseur de services d'application doit traiter le modèle d'importation de conditions générales d'application en utilisant ses systèmes et procédures locaux et des entrées provenant de fournisseurs de dispositifs et de tiers qui fournissent les composants de leur application pour créer une ou plusieurs conditions générales de fournisseur.

Le registre des identificateurs d'application oneM2M doit aussi, au minimum, remettre au fournisseur de services d'application la "liste des descripteurs" dans le langage du partenaire oneM2M pour aider ledit fournisseur à remplir les modèles d'importation de conditions générales et ainsi former l'ensemble des conditions générales qui lui sont spécifiques.

Le certificat de sécurité qui a été utilisé pendant le processus d'enregistrement de l'identificateur App-ID doit également être utilisé pour garantir l'intégrité et protéger le modèle d'importation de conditions générales d'application complété lors du stockage et de la transmission ultérieurs.

Le registre de l'identificateur App-ID oneM2M doit vérifier l'authenticité et l'intégrité des conditions générales du fournisseur de services d'application en vérifiant la signature au moyen du certificat de clé publique du fournisseur pendant l'enregistrement de l'identificateur App-ID.

Chacune des conditions générales du fournisseur de services d'application ou des logiciels doit être associée à l'identificateur App-ID du registre App-ID oneM2M.

11.4.1.2 Enregistrement des préférences de l'utilisateur en matière de confidentialité

Lorsqu'un utilisateur final s'abonne à un service fourni par un fournisseur de service d'application, il devient un sujet de données et télécharge ou consulte le modèle de préférences en matière de confidentialité depuis le portail PPM.

Le modèle utilisé par l'utilisateur final pour indiquer ses préférences en matière de confidentialité doit s'aligner sur le modèle utilisé par le fournisseur de services d'application, c'est-à-dire que les étiquettes énumérées dans l'Annexe normative J doivent s'afficher dans le même ordre.

L'utilisateur final sélectionne ou non des attributs pour indiquer ses préférences en matière de confidentialité, qui sont ensuite enregistrées sur le PPM au moyen du même portail.

11.4.1.3 Création d'une politique de confidentialité personnalisée pour chaque utilisateur final

Pour que le sujet des données confirme les différences entre les préférences en matière de confidentialité et la politique de confidentialité:

a) Si le choix de la fonctionnalité effectué par le fournisseur de services d'application et représenté par la valeur d'étiquette correspond à la préférence en matière de confidentialité choisie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir vert.

b) Si la fonctionnalité non sélectionnée par le fournisseur de services d'application et représentée par la valeur d'étiquette correspond à la préférence en matière de confidentialité définie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir vert.

c) Si la valeur sélectionnée pour la fonctionnalité par le fournisseur de services d'application et représentée par la valeur d'étiquette correspond à la préférence en matière de confidentialité choisie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir vert.

d) Si le choix de la fonctionnalité effectué par le fournisseur de services d'application et représenté par la valeur d'étiquette ne correspond pas à la préférence en matière de confidentialité choisie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir rouge.

e) Si la fonctionnalité non sélectionnée par le fournisseur de services d'application et représentée par la valeur d'étiquette ne correspond pas à la préférence en matière de confidentialité choisie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir rouge.

f) Si la valeur définie par le fournisseur de services d'application pour la fonctionnalité représentée par la valeur d'étiquette ne correspond pas à la préférence en matière de confidentialité définie par l'utilisateur pour une application ou un dispositif donnés, l'indicateur "Éléments présentés à l'utilisateur" correspondant doit devenir rouge.

Les règles ci-dessus doivent être contournées si un ou plusieurs profils de préférence facultatifs existent.

L'ordre de priorité est le suivant:

1) Priorité de politique: région.
2) Priorité de politique: pays.
3) Priorité de politique: ville.
4) Priorité de politique: état.
5) Contrôle parental.

12 Définitions des types de données oneM2M propres à la sécurité

12.1 Introduction

Le paragraphe 12 contient les définitions de types de données utilisées uniquement dans le cadre des spécifications de sécurité oneM2M.

Tous les types de données d'éléments XML définis pour être utilisés uniquement dans les spécifications de sécurité oneM2M doivent utiliser l'espace de noms suivant:

http://www.onem2m.org/xml/securityProtocols.

La présente Recommandation, et tout document de schéma XML ou document XML produit par oneM2M, utilise le préfixe "sec:" pour désigner cet espace de nom. Les documents de schéma XML produits par oneM2M et correspondant à la présente spécification sont disponibles dans [b-oneM2M.XML].

12.2 Types de données oneM2M simples propres à la sécurité

Le Tableau 12.2-1 décrit la définition de types de données simples spécifiques à la sécurité. Chacun des types figurant dans le Tableau 12.2-1 fait partie de l'une des catégories suivantes:

– Types de données atomiques calculés à partir des types de données de schéma XML par des restrictions autres que des énumérations.
– Types de données de listes construits à partir d'autres schémas XML ou de types de données atomiques définis par oneM2M.

Tableau 12.2-1 – Types de données oneM2M simples propres à la sécurité

Nom du type XSD	Usage	Exemples	Description
sec:relKeyID	Partie relative des identificateurs de clés symétriques	1he83he, ma-clé_nom, prénom.nom	Toute combinaison de caractères latins, de chiffres, de points ("."), de tirets bas ("_") et de tirets ("-").
sec:credentialID	Identificateur du justificatif d'identité	10-cetteclé@mafonctionmef.com	Valeurs sec:credIDTypeID et xs:anyURI séparées par un tiret ("-"). Voir le § 10.4. L'élément xs:anyURI est la valeur de l'identificateur du justificatif d'identité.

Tableau 12.2-1 – Types de données oneM2M simples propres à la sécurité

Nom du type XSD	Usage	Exemples	Description
sec:deviceConfigURI	Attribut *deviceConfigURI* de la ressource <*MEFBase*>, voir la Recommandation [ITU-T Y.4500.32].	1:http://server.dmprovider.com	Valeur de l'identificateur sec:devMgmtID (voir § 12.3.2.2) et URI d'un serveur de gestion de dispositifs séparées un deux-points (":").

12.3 Types de données oneM2M énumérés propres à la sécurité

12.3.1 Introduction

Les types de données oneM2M énumérés propres à la sécurité sont traités de la même manière que les types de données oneM2M énumérés définis au § 6.3.4 de la Recommandation [UIT-T Y.4500.4]. Ces types de données sont fondés sur le type <xs:integer>, les valeurs numériques étant interprétées conformément au § 12.3.2.

12.3.2 Définitions des types d'énumérations

12.3.2.1 sec:credIDTypeID

Le type d'énumération sec:credIDTypeID est utilisé dans l'identificateur sec:credentialID pour identifier le type du justificatif d'identité. Le Tableau 12.3.2.1-1 décrit l'interprétation du type d'énumération sec:credIDTypeID.

Tableau 12.3.2.1-1 – Interprétation du type d'énumération sec:credIDTypeID

Valeur	Interprétation	Note
10	Clé symétrique utilisée pour l'authentification auprès d'une fonction MEF (KpmID)	Voir le § 8.3.2.1
11	Clé symétrique utilisée pour l'authentification auprès d'une fonction MAF (KmID)	Voir le § 8.8.3.1
12	Clé symétrique utilisée pour l'authentification dans un cadre SAEF (KpsaID ou KcID)	Voir les § 8.2.2.1 et 8.2.2.3
13	Clé symétrique utilisée pour l'authentification dans le cadre ESPrim (pairwiseESPrimKeyID)	Voir le § 8.4.2
14	Clé symétrique utilisée pour le chiffrement direct dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (format d'identificateur de clé symétrique générique)	Voir le § 8.5.2
15	Clé symétrique utilisée pour l'enveloppement des clés symétriques dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (format d'identificateur de clé symétrique générique)	Voir le § 8.5.2
16	Clé symétrique utilisée pour le code HMAC dans la classe de sécurité ESData de signature seule (format d'identificateur de clé symétrique générique)	Voir le § 8.5.2
30	Certificat de clé publique brute utilisé dans le protocole TLS (identificateur de clé publique)	10.1.2
31	Certificat de dispositif utilisé dans le protocole TLS (identificateur unique mondialement d'instance matérielle)	10.1.1.4

Tableau 12.3.2.1-1 – Interprétation du type d'énumération sec:credIDTypeID

Valeur	Interprétation	Note
32	Certificat CSE-ID utilisé dans le protocole TLS (CSE-ID)	[ITU-T Y.4500.1]
33	Certificat AE-ID utilisé dans le protocole TLS (AE-ID)	[ITU-T Y.4500.1]
41	Certificat de clé publique brute utilisé pour la gestion de clés RSA ou ECDH dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (identificateur de clé publique)	10.1.2
42	Certificat de dispositif utilisé pour la gestion de clés RSA ou ECDH dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (identificateur unique mondialement d'instance matérielle)	10.1.1.4
43	Certificat CSE-ID utilisé pour la gestion de clés RSA ou ECDH dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (CSE-ID)	[ITU-T Y.4500.1]
44	Certificat AE-ID utilisé pour la gestion de clés RSA ou ECDH dans les classes de sécurité ESData de chiffrement seul ou de signature et de chiffrement imbriqués (AE-ID)	[ITU-T Y.4500.1]
51	Certificat de clé publique brute utilisé pour la gestion de clés RSA ou ECDH dans la classe de sécurité ESData de signature seule (identificateur de clé publique)	10.1.2
52	Certificat de dispositif utilisé pour les signatures RSA ou ECDSA dans la classe de sécurité ESData de signature seule (identificateur unique mondialement d'instance matérielle)	10.1.1.4
53	Certificat CSE-ID utilisé pour la gestion de clés RSA ou ECDH dans la classe de sécurité ESData de signature seule (CSE-ID)	[ITU-T Y.4500.1]
54	Certificat AE-ID utilisé pour la gestion de clés RSA ou ECDH dans la classe de sécurité ESData de signature seule (AE-ID)	[ITU-T Y.4500.1]
NOTE – La forme de l'identificateur du type de justificatif d'identité est décrite entre crochets.		

12.3.2.2 sec:devMgmtID

Le type d'énumération sec:devMgmtID est utilisé dans l'URI sec:deviceConfigURI comme identificateur de la technologie de gestion de dispositifs utilisée pour la configuration des dispositifs de terrain (voir la Recommandation [UIT-T Y.4500.22]). Le type d'énumération sec:devMgmtID est également utilisé dans l'élément *devMgmtID* de l'élément *devCfgArgs* des arguments *cmdArgs* de l'élément *cmdDescription* de la commande du client MEF (voir le § 8.3.9.8) pour indiquer le protocole de gestion des dispositifs à utiliser lors de la configuration des dispositifs [UIT-T Y.4500.22]. L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> figurant dans [UIT-T Y.4500.32]). Le Tableau 12.3.2.2-1 décrit l'interprétation du type d'énumération sec:devMgmtID.

Tableau 12.3.2.2-1 – Interprétation du type d'énumération sec:devMgmtID

Valeur	Interprétation	Note
1	OMA DMv1.3	Voir [b-UIT-T Y.4500.5]
2	OMA DMv2.0	Voir [b-UIT-T Y.4500.5]
3	OMA LwM2M	Voir [b-UIT-T Y.4500.5]
4	BBF TR-069	Voir [b-UIT-T Y.4500.6]

12.3.2.3 sec:cmdClassID

Le type d'énumération sec:cmdClassID est utilisé dans l'élément *cmdDescription* de la commande du client MEF (voir le § 8.3.9.4) pour indiquer la classe *cmdClass* de la description *cmdDescription*. L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32]. Le Tableau 12.3.2.3-1 décrit l'interprétation du type d'énumération sec:cmdClassID.

Tableau 12.3.2.3-1 – Interprétation du type d'énumération sec:cmdClassID

Valeur	Interprétation	Note
0	NO_MORE_COMMANDS	La classe de commande est définie au § 8.3.9.6.
1	CERT_PROV	La classe de commande est définie au § 8.3.9.7. La fourniture des certificats est expliquée au § 8.3.6.
2	DEV_CFG	La classe de commande est définie au § 8.3.9.8. La configuration des dispositifs est exposée dans la Recommandation [ITU-T Y.4500.22].
3	MO_NODE	La classe de commande est définie au § 8.3.9.9.

12.3.2.4 sec:cmdStatusCode

Le type d'énumération sec:cmdStatusCode est utilisé par l'élément *cmdStatusCode* afin d'indiquer l'état d'une commande de client MEF. L'élément *cmdStatus* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32]. Le Tableau 12.3.2.4-1 décrit l'interprétation du type d'énumération sec:cmdStatusCode.

Tableau 12.3.2.4-1 – Interprétation du type d'énumération sec:cmdStatusCode

Valeur	Interprétation	Note
10	MEF_CLIENT_CMD_ISSUED	Voir le § 8.3.9.5.2
11	MEF_CLIENT_CMD_REISSUED	Voir le § 8.3.9.5.3
20	MEF_CLIENT_CMD_OK	Voir le § 8.3.9.5.4
40	MEF_CLIENT_CMD_ REPEATED_CMD_ID	Voir le § 8.3.9.5.5
41	MEF_CLIENT_CMD_CLASS_NOT_SUPPORTED	Voir le § 8.3.9.5.6
42	MEF_CLIENT_CMD_BAD_ ARGUMENTS	Voir le § 8.3.9.5.7
43	MEF_CLIENT_CMD_UNACCEPTABLE_ARGUMENTS	Voir le § 8.3.9.5.8
100	MEF_CLIENT_CMD_CERT_PROV_SERVER_ERROR	Voir le § 8.3.9.5.9
101	MEF_CLIENT_CMD_CERT_PROV_CLIENT_ERROR	Voir le § 8.3.9.5.10
201	MEF_CLIENT_CMD_DEV_CFG_SERVER_ERROR	Voir le § 8.3.9.5.11
202	MEF_CLIENT_CMD_DEV_CFG_CLIENT_ERROR	Voir le § 8.3.9.5.12
300	MEF_CLIENT_CMD_MO_NODE_NOT_FOUND	Voir le § 8.3.9.5.13
301	MEF_CLIENT_CMD_MO_NODE_TYPE_CONFLICT	Voir le § 8.3.9.5.14
302	MEF_CLIENT_CMD_MO_NODE_BAD_ARGS	Voir le § 8.3.9.5.15
303	MEF_CLIENT_CMD_MO_NODE_UNACCEPTABLE_ARGS	Voir le § 8.3.9.5.16
304	MEF_CLIENT_CMD_MO_NODE_INCONSISTENT_CONFIG	Voir le § 8.3.9.5.17
305	MEF_CLIENT_CMD_MO_NODE_EXECUTION_ERROR	Voir le § 8.3.9.5.18

12.3.2.5 sec:certProvProtocolID

Le type d'énumération sec:certProvProtocolID est utilisé pour l'élément *certProvProtocolID* de l'élément *certProvCmdArgs* des arguments *cmdArgs* de l'élément *cmdDescription* de la commande du client MEF (voir le § 8.3.9.7) pour indiquer le protocole de fourniture de certificat à utiliser. L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32].

Tableau 12.3.2.5-1 – Interprétation du type d'énumération sec:certProvProtocolID

Valeur	Interprétation	Note
1	EST	Voir le § 8.3.6.2
2	SCEP	Voir le § 8.3.6.3

12.3.2.6 sec:certSubjectType

Le type d'énumération sec:certSubjectType est utilisé pour l'élément *certSubjectType* de l'élément *certProvCmdArgs* des arguments *cmdArgs* de l'élément *cmdDescription* de la commande du client MEF (voir le § 8.3.9.7) pour indiquer si le sujet du certificat fourni sera un nœud, une entité de services communs ou une entité d'application. L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32].

Tableau 12.3.2.6-1 – Interprétation du type d'énumération sec:certSubjectType

Valeur	Interprétation	Note
1	Identificateur de nœud	Voir la Recommandation [ITU-T Y.4500.1], § 7.1.5
2	CSE-ID	Voir la Recommandation [ITU-T Y.4500.1], § 7.2
3	AE-ID	Voir la Recommandation [ITU-T Y.4500.1], § 7.2

12.3.2.7 sec:objectTypeID

Le type d'énumération sec:objectTypelID est utilisé pour l'élément *objectTypelID* de l'élément *MONodeCmdArgs* des arguments *cmdArgs* de l'élément *cmdDescription* de la commande du client MEF (voir le § 8.3.9.9) pour indiquer le type d'un nœud MO. L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32].

Tableau 12.3.2.7-1 – Interprétation du type d'énumération sec:certProvProtocolID

Valeur	Interprétation	Note
1	[*authenticationProfile*]	Voir la Recommandation [ITU-T Y.4500.22], § 7.1.4 et 7.2.4.
2	[*trustAnchorCred*]	Voir la Recommandation [ITU-T Y.4500.22], § 7.1.6 et 7.2.6.
3	[*MAFClientRefCfg*]	Voir la Recommandation [ITU-T Y.4500.22], § 7.1.7 et 7.2.7.

12.4 Types de données oneM2M complexes propres à la sécurité

12.4.1 Données de configuration du client MAF et MEF

Le Tableau 12.4.1-1 définit l'attribution des types de données aux quatre conteneurs de données utilisés dans les procédures d'enregistrement et de configuration de clé de client MAF et MEF. Il convient de noter que ces conteneurs de données ne sont pas définis sous la forme de types de ressource car les informations ne sont pas accessibles à distance. Les éléments d'information de ces conteneurs sont gérés au moyen de procédures de gestion de dispositifs (voir la

Recommandation [UIT-T Y.4500.22]) ou par configuration manuelle. Le Tableau 12.4.1-1 énumère les types utilisés dans les procédures de configuration d'enregistrement des fonctions MAF et MEF.

Tableau 12.4.1-1 – Types utilisés dans les procédures de configuration d'enregistrement des fonctions MAF et MEF

Nom du conteneur de données	Usage	Type de données	Notes
mefClientRegCfg	Configuration de l'enregistrement du client MEF (voir le § 8.3.7.2)	sec:clientRegCfg	Voir le § 12.4.2
mafClientRegCfg	Configuration de l'enregistrement du client MEF (voir le § 8.8.3.2)		
mefKeyRegCfg	Configuration de l'enregistrement de la clé MEF (voir le § 8.3.7.3)	sec:keyRegCfg	Voir le § 12.4.3
mafKeyRegCfg	Configuration de l'enregistrement de la clé MAF (voir le § 8.8.3.3)		

12.4.2 sec:clientRegCfg

Le type de données sec:clientRegCfg s'applique aux conteneurs de données *mefClientRegCfg* et *mafClientRegCfg* utilisés dans la configuration d'enregistrement de client MEF et MAF (voir respectivement les § 8.3.7.2 et 8.8.3.2). Le Tableau 12.4.2-1 donne la définition du type sec:clientRegCfg.

Tableau 12.4.2-1 – Définition du type sec:clientRegCfg

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
expirationTime	m2m:timestamp	0 ou 1	[ITU-T Y.4500.4]
labels	m2m:labels	0 ou 1	[ITU-T Y.4500.4]
fqdn	xs:anyURI	1	
adminFQDN	xs:anyURI	1	
httpPort	xs:unsignedByte	0 ou 1	
coapPort	xs:unsignedByte	0 ou 1	
websocketPort	xs:unsignedByte	0 ou 1	

12.4.3 sec:keyRegCfg

Le type de données sec:keyRegCfg s'applique aux conteneurs de données *mefKeyRegCfg* et *mafKeyRegCfg* utilisés dans la configuration d'enregistrement de clé MEF et MAF (voir respectivement les § 8.3.7.3 et 8.8.3.3). Le Tableau 12.4.3-1 donne la définition du type sec:keyRegCfg.

Tableau 12.4.3-1 – Définition du type sec:keyRegCfg

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
expirationTime	m2m:timestamp	0 ou 1	[ITU-T Y.4500.4]
labels	m2m:labels	0 ou 1	[ITU-T Y.4500.4]
adminFQDN	xs:anyURI	1	
SUID	m2m:suid	1	[ITU-T Y.4500.4]
targetIDs	m2m:listOfM2MID	0 ou 1	[ITU-T Y.4500.4]

12.4.4 sec:cmdDescription

Le type complexe sec:cmdDescription est utilisé par l'élément cmdDescription afin de décrire une commande de client MEF (voir le § 8.3.9.5). L'élément *cmdDescription* est un attribut du type de ressource <*mefClientCmd*> défini dans la Recommandation [UIT-T Y.4500.32]. Le Tableau 12.4.4-1 donne la définition du type sec:cmdDescription.

Tableau 12.4.4-1 – Définition du type sec:cmdDescription

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
cmdClassID	sec:cmdClassID	1	Voir le § 12.3.2.3
cmdArgs	sec:cmdArgs	1	Voir le § 12.4.5
targetID	m2m:ID	1	[ITU-T Y.4500.4]

12.4.5 sec:cmdArgs

Le type complexe sec:cmdArgs est utilisé par l'élément cmdArgs du type de données sec:cmdDescription. Le Tableau 12.4.5-1 donne la définition du type sec:cmdArgs.

Tableau 12.4.5-1 – Définition du type sec:cmdArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
noMoreCmdArgs	sec:noMoreCmdArgs	0 ou 1	Voir le § 12.4.6
certProvCmdArgs	sec:certProvCmdArgs	0 ou 1	Voir le § 12.4.7
devCfgCmdArgs	sec:devCfgCmdArgs	0 ou 1	Voir le § 12.4.8
MONodeCmdArgs	sec:MONodeCmdArgs	0 ou 1	Voir le § 12.4.9

Ce type est un élément xs:choice. Il contient les éléments d'une seule ligne du Tableau 12.4.5-1.

12.4.6 sec:noMoreCmdArgs

Le type complexe sec:noMoreCmdArgs est utilisé dans l'élément sec:cmdDescription. Le Tableau 12.4.6-1 donne la définition du type sec:noMoreCmdArgs.

Tableau 12.4.6-1 – Définition du type sec: noMoreCmdArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
retryDuration	xs:duration	1	

12.4.7 sec:certProvCmdArgs

Le type complexe sec:certProvCmdArgs est utilisé dans l'élément sec:cmdDescription. Le Tableau 12.4.7-1 donne la définition du type sec:certProvCmdArgs.

Tableau 12.4.7-1 – Définition du type sec:certProvCmdArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
certProvProtocolID	sec:certProvProtocolID	1	Voir le § 12.3.2.5
URI	xs:anyURI	1	
certSubjectType	sec:certSubjectType	1	Voir le § 12.3.2.6
certSubjectID	xs:union de m2m:nodeID et m2m:ID	1	Voir la Recommandation [ITU-T Y.4500.4], § 6.3.3

12.4.8 sec:devCfgCmdArgs

Le type complexe sec:devCfgCmdArgs est utilisé dans l'élément sec:cmdDescription. Le Tableau 12.4.8-1 donne la définition du type sec:devCfgCmdArgs.

Tableau 12.4.8-1 – Définition du type sec:devCfgCmdArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
deviceConfigURI	sec:deviceConfigURI	1	Voir le § 12.2.

12.4.9 sec:MONodeCmdArgs

Le type complexe sec:MONodeCmdArgs est utilisé dans l'élément sec:cmdDescription. Le Tableau 12.4.9-1 donne la définition du type sec:MONodeCmdArgs.

Tableau 12.4.9-1 – Définition du type sec:MONodeCmdArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
objectPath	xs:anyURI	1	
objectTypeID	sec:objectTypeID	1	Voir le § 12.3.2.7
objectTypeSpecifcArgs	sec:authProfileMONodeArgs	0 ou 1	Voir le § 12.4.10

12.4.10 sec:authProfileMONodeArgs

Le type complexe sec:authProfileMONodeArgs est utilisé dans l'élément sec:MONodeCmdArgs. Le Tableau 12.4.10-1 donne la définition du type sec:authProfileMONodeArgs.

Tableau 12.4.10-1 – Définition du type sec:authProfileMONodeArgs

Chemin de l'élément	Type de l'élément	Multiplicité	Notes
SUID	m2m:suid	1	Voir la Recommandation [ITU-T Y.4500.4], § 6.3.4.2.39.

Annexe A

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe B

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe C

Protocoles de sécurité associés à des technologies d'environnement sécurisé spécifiques

(Cette annexe fait partie intégrante de la présente Recommandation.)

C.0 Introduction

L'environnement sécurisé prenant en charge les fonctions de sécurité définies par oneM2M fournit un niveau et un type de protection (par exemple protection de l'intégrité et de la confidentialité et résistance aux tentatives d'altération) aux informations qu'il contient, indépendamment de la méthode de protection (par exemple, carte UICC, élément de sécurité intégré, environnement d'exécution fiable, etc.). L'administration de leur contenu dépend de la mise en œuvre et s'appuie sur des normes existantes pour des technologies d'environnement sécurisé spécifiques. De plus amples informations les concernant sont disponibles ci-dessous.

C.1 UICC

Dans le cas d'une carte UICC (environnement sécurisé conforme à la spécification [ETSI TS 102 671]), les mécanismes hertziens (OTA), tels qu'ils sont définis dans [ETSI TS 102 225] et [ETSI TS 102 226], et leurs extensions [ETSI TS 131 115], [ETSI TS 131 116] relative aux réseaux sous-jacents 3GPP ou [ARIB STD-T64-C.S0078-0] et [ARIB STD-T64 C.S0079-0] portant sur les réseaux sous-jacents 3GPP2, doivent être pris en charge pour permettre l'administration de la sécurité des données sensibles de la couche de service M2M. La carte UICC fournit le niveau de protection 3 le plus élevé contre les attaques, selon la classification des niveaux de protection.

C.2 Autre élément de sécurité et élément de sécurité intégré doté d'une interface ISO 7816

Lorsque l'environnement sécurisé est mis en œuvre sous la forme d'un élément de sécurité ou d'un élément de sécurité intégré prenant en charge une interface ISO/CEI 7816 [ISO/CEI 7816-4], l'administration à distance peut, par exemple, être effectuée conformément à [b-GP DTSERAM]. Un élément de sécurité intégré fournit le niveau de protection 3 le plus élevé contre les attaques, selon la classification des niveaux de protection.

C.3 Environnement d'exécution fiable

Si l'environnement sécurisé est mis en œuvre comme un environnement d'exécution fiable (TEE) conformément à la norme GlobalPlatform [b-GP TEESystem], l'administration à distance doit être prise en charge selon ce qui est défini dans [b-GP TEEAdmin]. Un élément de sécurité intégré fournit le niveau de protection 2 intermédiaire contre les attaques, selon la classification des niveaux de protection.

C.4 Liaison de l'environnement sécurisé à l'entité de services communs

Si l'environnement sécurisé est mis en œuvre comme un élément de sécurité indépendant prenant en charge la spécification [ETSI TS 102 221], le canal sécurisé de plate-forme à plate-forme défini dans [ETSI TS 102 484] assure la liaison logique de l'environnement sécurisé avec une entité de services communs ou une entité d'application spécifique. Cela protège également les informations échangées entre l'environnement sécurisé et l'entité associée sur les interfaces physiquement exposées. Cela est donc recommandé pour les dispositifs exposés physiquement aux auteurs d'attaques.

Annexe D

Cadre de sécurité des cartes UICC pour la prise en charge des services oneM2M

(Cette annexe fait partie intégrante de la présente Recommandation.)

D.0 Introduction

La présente annexe s'applique lorsque la carte UICC (type d'élément de sécurité indépendant conforme à [ETSI TS 102 221] et [ETSI TS 102 671]) gère la sécurité de la couche des services M2M, qu'elle ne sert que pour la préconfiguration des données de la couche des services M2M dans les dispositifs ou les passerelles M2M, ou qu'elle est utilisée comme environnement sécurisé dans un dispositif ou une passerelle M2M.

Plus précisément, l'intégration de la carte UICC dans la sécurité oneM2M peut inclure l'une des étapes suivantes:

Préconfiguration des justificatifs d'identité initiaux des nœuds M2M selon l'une des méthodes suivantes:

Simple préconfiguration et administration des données de service M2M (justificatifs d'identité initiaux et autres paramètres préconfigurés), c'est-à-dire la configuration de services M2M fondée sur une carte UICC.

Prise en charge de l'amorçage assisté par l'infrastructure des justificatifs symétriques M2M par le biais du calcul des justificatifs symétriques du réseau d'accès stockés dans la carte UICC, au moyen de l'architecture GBA.

Calcul d'une clé d'association de sécurité directement obtenue à partir des justificatifs symétriques du réseau d'accès, au moyen de l'architecture GBA. Il convient de noter que ce processus peut être pris en charge par une application d'accès au réseau de la carte UICC indépendamment de la présence de la structure d'information décrite dans la présente annexe.

La prise en charge de la fourniture par carte UICC des informations d'abonnement aux services M2M doit être indiquée dans le tableau des services M2M pour l'abonnement aux services M2M correspondants, comme défini dans la présente annexe.

La prise en charge du calcul de clé au moyen de l'architecture GBA qui peut être utilisée pour l'amorçage ou l'association de sécurité doit toujours être indiquée dans le tableau des services de l'application de carte UICC de l'opérateur du réseau d'accès prenant en charge l'infrastructure GBA.

Au niveau le plus élémentaire, la préconfiguration M2M fondée sur la carte UICC nécessite un cadre interopérable pour stocker et administrer les informations associées de la carte UICC. Toute autre action nécessite un cadre de découverte des services disponibles proposés par la carte UICC pour le nœud de champ M2M hôte. La présente annexe a pour objet de définir ce cadre, qui permet à la fois la fourniture initiale des services et l'administration à distance de la sécurité des informations d'abonnement pendant la durée de celui-ci.

Il est courant qu'un nœud de champ M2M comprenne une application de carte UICC protégeant les justificatifs de sécurité du réseau d'accès. Ces justificatifs sont utilisés pour calculer les justificatifs de sécurité de la couche de service M2M utilisés pour amorcer des services M2M ou établir une association de sécurité dans la couche de service. Étant donné qu'un accord de confiance entre l'opérateur du réseau d'accès concerné et le fournisseur de services M2M est nécessaire dans ces cas-là, la prise en charge de la carte UICC pour les services M2M doit être traitée dans le contexte de l'application d'accès au réseau associée de la carte UICC. En particulier, la prise en charge de la carte UICC pour le calcul des justificatifs d'identité M2M au moyen de l'architecture GBA doit être indiquée dans l'application UICC de l'opérateur du réseau d'accès. Cela est décrit au § D.1.

Même lorsque les justificatifs d'identité de la couche de services M2M ne sont pas calculés à partir des justificatifs d'identité du réseau d'accès, la carte UICC peut être utilisée comme un environnement sécurisé qui protège le justificatif d'identité symétrique ou asymétrique utilisé pour assurer la sécurité de la racine d'un nœud de champ M2M. En pareil cas, les informations d'abonnement M2M et les méthodes connexes constituent une application indépendante qui réside sur une carte UICC, au sens de [ETSI TS 102 221]. En particulier, [ETSI TS 102 221] décrit les propriétés indépendantes de l'application de l'interface UICC/terminal, telles que les caractéristiques physiques et la structure logique.

NOTE – Un terminal, au sens de [ETSI TS 102 221], est la partie du nœud de champ M2M qui contient la carte UICC (par exemple, un modem de communication ou un environnement de traitement de nœud M2M).

Les propriétés spécifiques à l'application du module d'identité du fournisseur de services M2M détenant des justificatifs symétriques sont précisées au § D.2.

Le stockage des éléments d'information M2M dans la carte UICC et les procédures utilisées pour la communication entre le nœud de champ M2M hôte et la carte UICC doivent être conformes aux spécifications de la présente annexe. La présente annexe utilise les abréviations et les conventions de codage définies dans [ETSI TS 102 221].

D.1 Cadre des services oneM2M fondé sur une carte UICC dans un réseau d'accès

D.1.1 Caractéristiques du cadre de services oneM2M fondé sur une carte UICC du réseau d'accès

Un cadre de services oneM2M fondé sur une carte UICC du réseau d'accès est toujours associé à un seul abonnement à un service M2M et comprend un seul fichier dédié (DF), $DF_{1M2M,}$ conforme aux spécifications du § D.1.3 et mis en œuvre dans le fichier ADF d'une application d'accès au réseau stocké sur la carte UICC. Cette situation concerne le cas où une relation de confiance a été établie entre le fournisseur de services M2M et l'opérateur du réseau d'accès propriétaire du fichier dédié d'application hôte.

NOTE 1 – Cela n'implique pas nécessairement que les justificatifs d'identité du réseau d'accès du fichier dédié d'application (ADF) correspondant soient utilisés pour calculer les justificatifs de la couche des services M2M. Par exemple, un opérateur de réseau d'accès peut refuser qu'un fournisseur de services M2M les calcule à partir des justificatifs d'identité du réseau d'accès, tout en acceptant de réserver de l'espace sur sa carte UICC pour préconfigurer des justificatifs d'identité indépendants ou prendre en charge un amorçage assisté par l'infrastructure de service.

Il peut exister plusieurs cadres de services oneM2M (DF_{1M2M}) dans le fichier ADF d'un seul abonnement au réseau d'accès lorsque ce dernier est utilisé par plusieurs abonnements à des services M2M indépendants. Les identificateurs du fichier DF_{1M2M} contenus dans tout fichier ADF doivent être énumérés sous l'entrée correspondante dans le fichier EF_{DIR}, comme cela est décrit au § D.1.2.

NOTE 2 – Un abonnement unique à la couche des services M2M peut également utiliser plusieurs réseaux d'accès, ces abonnements étant mieux configurés dans un fichier ADF spécifique, comme indiqué au § D.2.

Le contenu de tout fichier DF_{1M2M} compris dans un fichier ADF d'application de réseau d'accès doit être tel que décrit au § D.1.3.

D.1.2 Découverte du cadre de services M2M pour les cartes UICC des réseaux d'accès

Lorsqu'une application d'accès au réseau UICC prend en charge un ou plusieurs abonnements à des services M2M à l'aide d'une DF_{1M2M}, l'entrée EF_{DIR} correspondant à cette application d'accès au réseau UICC contient les objets de données M2M suivants énumérés dans les Tableaux D.1 à D.3:

Objets de données du cadre de services oneM2M: définit l'association entre l'identificateur d'un abonnement à un service M2M configuré dans le fichier ADF et le fichier DF associé correspondant à cet abonnement M2M. De même, chaque abonnement à un service M2M est associé à un fichier DF. Chacun de ces fichiers est ci-après dénommé "DF_{1M2M}".

Il doit exister autant d'objets de données de cadre de services oneM2M que d'abonnements aux services M2M configurés dans le fichier ADF.

Tableau D.1 – Codage des objets de données oneM2M associés

Octets	Longueur	Description	État
1	1	Étiquette de modèle discrétionnaire = '73'	O
2	1	Longueur du modèle discrétionnaire = X	M
3 à (2 + X)	X	Modèle discrétionnaire	X

Tableau D.2 – Codage des objets de données oneM2M associés du modèle discrétionnaire

Octets	Longueur	Description	État
1	1	Étiquette de contenu de données propres au service oneM2M = 'A2'	M
2	1	Longueur du contenu de données propres au service M2M = Y	M
3 à (2 + Y)	Y	Contenu de données propres au service M2M	M

Tableau D.3 – Codage des objets de données associés au contenu des données propres au service oneM2M

Octets	Longueur	Description	État
1	1	Étiquette de fourniture de service prise en charge par oneM2M = '80'	M
2	1	Longueur de l'étiquette de fourniture de service prise en charge par oneM2M = A	M
3 ou 4	2	Identificateur de fichier dédié M2M destiné au suivi de l'abonnement au service M2M	M
5 à (A + 2)	(A – 2)	Identificateur d'abonnement M2M	M

Codage:

Identificateur de fichier M2M dédié:

Contient l'identificateur du fichier DF_{1M2M} associé à la configuration de l'abonnement au service M2M identifié dans l'objet de données.

Identificateur d'abonnement M2M:

L'identificateur de l'abonnement au service M2M configuré dans le fichier DF_{1M2M} est indiqué dans l'objet de données et encodé au format binaire.

D.1.3 Contenu des fichiers au niveau du fichier DF_{1M2M}

D.1.3.0 Introduction

Le présent paragraphe décrit les fichiers EF pour la fourniture de services M2M propres à un fournisseur de services M2M unique, qui définit les conditions d'accès, les éléments de données et le codage. Un élément de données est une partie d'un fichier EF qui représente une entité logique complète.

La structure du fichier DF_{1M2M} est illustrée dans la Figure D.1.

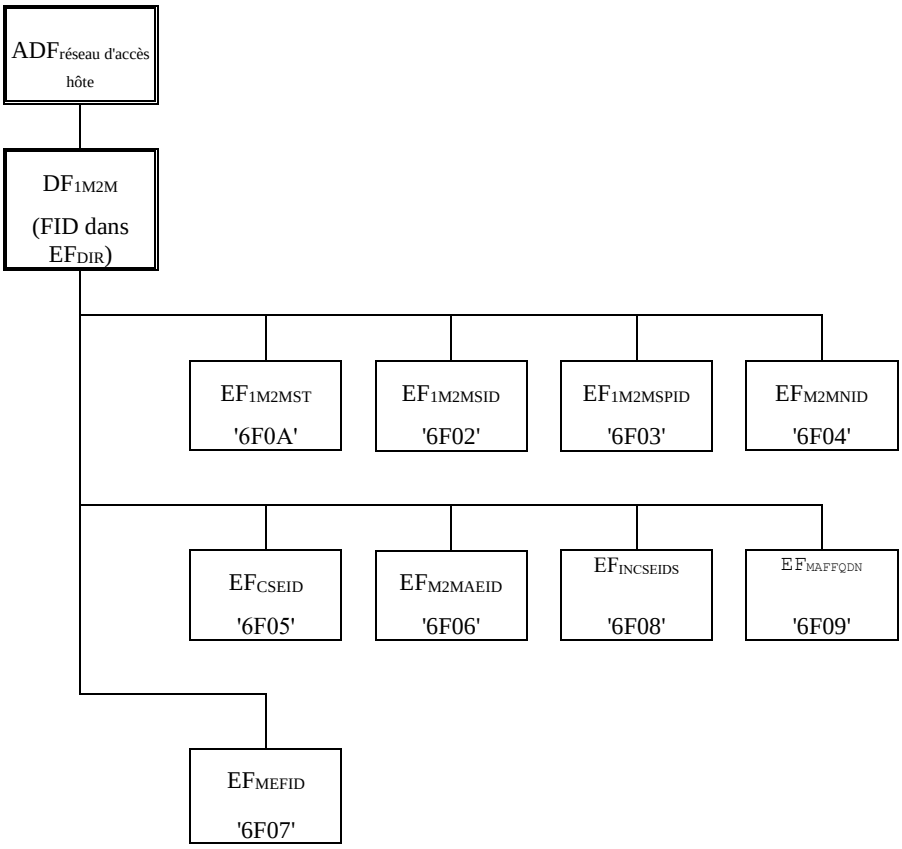


Figure D.1 – Identificateurs de fichier et structures de répertoire du fichier DF_{1M2M} d'un fichier ADF d'application de réseau d'accès hôte

D.1.3.1 EF_{1M2MST} (tableau des services oneM2M)

Ce fichier élémentaire (EF) indique quels services oneM2M facultatifs sont disponibles pour l'abonnement correspondant. Si un service n'est pas indiqué comme disponible dans le fichier DF_{1M2M}, le nœud de champ M2M hôte ne choisit pas ce service. La présence de ce fichier est obligatoire si des services facultatifs sont fournis par l'abonnement.

Identificateur: '6F0A'	Structure: transparente	Obligatoire
SFI: '0A'		
Taille du fichier: X octets, où X ≥ 1	Fréquence de mise à jour: faible	

Conditions d'accès:

LECTURE	ALW
MISE À JOUR	ADM
DÉSACTIVATION	ADM
ACTIVATION	ADM

Octets	Description	O/F	Longueur
1	Services N° 1 à 8	O	1 octet
2	Services N° 9 à 16	F	1 octet
3	Services N° 17 à 24	F	1 octet
4	Services N° 25 à 32	F	1 octet
etc.			
X	Services (8X – 7) à (8X)	F	1 octet

–Services

Contenus:		
	Service N° 1	Fourniture d'identificateur CSE-ID local
	Service N° 2	Fourniture de liste d'identificateurs IN-CSE-ID
	Service N° 3	Fourniture du nom de domaine complet de la fonction MAF
	Service N° 4	Fourniture de liste d'identificateurs AE-ID M2M locaux
	Service N° 5	Amorçage: fourniture d'adresse MEF
	Service N° 6	Information de l'identificateur de nœud M2M
	Service N° 7	Fourniture sécurisée GBA (voir la note)
	Service N° 8	Connexion sécurisée GBA (voir la note)

NOTE – Les services N° 7 et 8 ne peuvent être disponible que dans un tableau des services oneM2M disponible dans un fichier DF_{1M2M} stocké dans le fichier ADF de l'application d'accès au réseau à partir de laquelle les justificatifs de la couche de services M2M sont censés être calculés.

Le fichier EF contient au moins un octet. D'autres octets peuvent être inclus, mais si le fichier EF contient un octet facultatif, il doit aussi contenir tous les octets qui précèdent ce dernier. D'autres services seront possibles ultérieurement et seront codés sur d'autres octets enregistrés dans le fichier EF. Codage:

Un bit est utilisé pour coder chaque service:

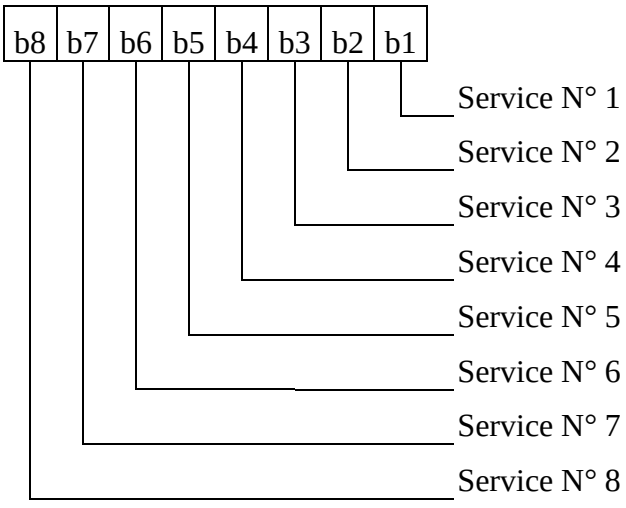
– Bit = 1: service disponible;

– Bit = 0: service non disponible.

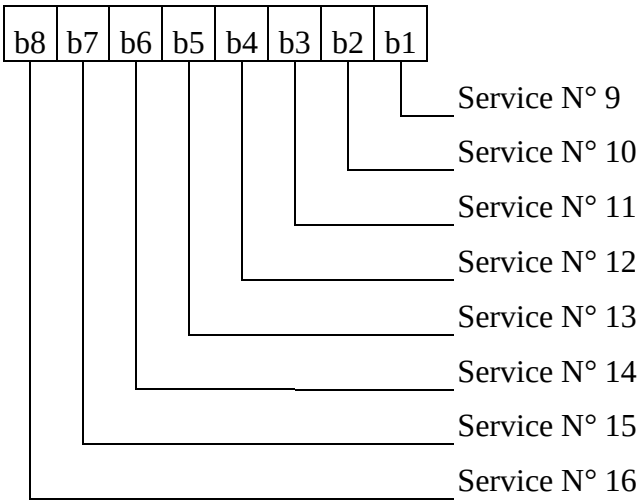
Quand le service est disponible, cela signifie que l'abonnement au service M2M configuré dans le fichier DF ou ADF actif permet de prendre en charge le service et qu'il est disponible pour l'utilisateur du service M2M.

Quand le service n'est pas disponible, cela signifie qu'il ne doit pas être utilisé par l'utilisateur de l'abonnement au service M2M, même si ledit abonnement prend en charge le service.

Premier octet:



Deuxième octet:



etc.

D.1.3.2 $EF_{1M2MSID}$ (identificateur d'abonnement oneM2M)

Ce fichier EF contient l'identificateur d'abonnement oneM2M, M2M-Sub-ID. Ce fichier EF doit inclure un seul objet comprenant une étiquette, une longueur et une valeur (TLV).

Identificateur: '6F02'	Structure: transparente		Obligatoire
SFI: '02'			
Taille du fichier: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1	Objet de données TLV d'identificateur d'abonnement M2M	O	X octets

Le champ de valeur de l'identificateur d'abonnement M2M doit contenir l'identificateur M2M-Sub-ID codé comme cela est décrit dans la Recommandation [UIT-T Y.4500.4]. La valeur d'étiquette de l'objet de données TLV d'identificateur d'abonnement M2M doit être égale à 80.

D.1.3.3 $EF_{1M2MSPID}$ (identificateur du fournisseur de services oneM2M)

Ce fichier EF contient l'identificateur du fournisseur de services oneM2M, M2M-SP-ID, lié à l'abonnement défini dans le fichier $EF_{1M2MSID}$. Ce fichier EF doit inclure un seul objet TLV.

Identificateur: '6F03'	Structure: transparente		Obligatoire
SFI: '03'			
Taille du fichier: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1	Objet de données TLV de l'identificateur M2M-SP-ID	O	X octets

Le champ M2M-SP-ID doit contenir l'identificateur M2M-SP-ID codé comme cela est décrit dans la Recommandation [UIT-T Y.4500.4]. La valeur d'étiquette de l'objet de données TLV de l'identificateur M2M-SP-ID doit être égale à 80.

D.1.3.4 EF_{M2MNID} (identificateur de nœud M2M)

Ce fichier EF contient l'identificateur de nœud M2M prenant en charge l'entité de services communs locale. Il peut être utilisé pour associer logiquement une carte UICC à un nœud M2M donné. Si le Service N° 6 est disponible, ce fichier doit être présent. Ce fichier EF doit inclure un seul objet TLV.

Identificateur: '6F04'	Structure: transparente		Facultatif
SFI: '04'			
Taille du fichier: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1 à X	Objet TLV de l'identificateur M2M-Node-ID	O	X octets

Le champ de valeur M2M-Node-ID doit contenir l'identificateur M2M-Node-ID codé comme cela est décrit dans la Recommandation [UIT-T Y.4500.4].

D.1.3.5 EF_{CSEID} (identificateur d'entité de services communs locale)

Ce fichier EF contient l'identificateur de l'entité de services communs locale, CSE-ID, destiné au nœud de champ M2M associé à l'abonnement défini dans le fichier $EF_{1M2MSID}$. S'il est présent, ce fichier est utilisé par le nœud du champ M2M pour préconfigurer l'identificateur CSE-ID. Si le Service N° 1 est disponible, ce fichier doit être présent. Ce fichier EF doit inclure un seul objet TLV.

Identificateur: '6F05'	Structure: transparente		Facultatif
SFI: '05'			
Taille du fichier: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1	Objet de données TLV de l'identificateur CSE-ID	O	X octets

Objet de données TLV de l'identificateur CSE-ID

Contenus:

Le champ de la valeur CSE-ID contient l'identificateur de l'entité de services communs locale formaté comme une URI.

Codage:

L'URI sera codée en une chaîne d'octets conformément aux règles de codage UTF-8 décrites dans [IETF RFC 3629]. La valeur d'étiquette de l'objet de données TLV de l'URI doit être égale à 80.

D.1.3.6 $EF_{M2MAE\text{-}ID}$ (liste d'identificateurs d'application M2M)

Ce fichier EF contient la liste des identificateurs des applications M2M locales (AE-ID) prises en charge par l'abonnement défini dans le fichier $EF_{1M2MSID}$. Si le Service N° 4 est disponible, ce fichier doit être présent.

Identificateur: '6F06'	Structure: linéaire fixe		Facultatif
SFI: '06'			
Longueur de l'enregistrement: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1 à X	Objet de données LV de l'identificateur AE-ID M2M	O	X octets

Objet de données LV de l'identificateur AE-ID M2M

Contenus:

Le champ de valeur doit contenir l'identificateur d'entité d'application M2M formaté comme une URI.

Codage:

L'URI sera codée en une chaîne d'octets conformément aux règles de codage UTF-8 décrites dans [IETF RFC 3629].

D.1.3.7 $EF_{INCSEIDS}$ (liste d'identificateurs IN-CSE M2M)

Ce fichier EF contient une liste d'identificateurs IN-CSE-ID préconfigurés, servant à déterminer le prochain point de contact après la configuration ou l'amorçage des services M2M. Si le Service N° 2 est disponible, ce fichier doit être présent.

Identificateur: '6F08'	Structure: linéaire fixe		Facultatif
Longueur de l'enregistrement: X octets		Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM			
Octets	Description	O/F	Longueur
1 à X	Objet de données LV de l'identificateur IN-CSE	O	X octets

Objet de données LV de l'identificateur IN-CSE

Contenus:

Le champ de la valeur contient l'identificateur IN-CSE formaté comme une URI.

Codage:

L'URI sera codée en une chaîne d'octets conformément aux règles de codage UTF-8 décrites dans [IETF RFC 3629].

D.1.3.8 $EF_{MAFFQDN}$ (nom de domaine complet de la fonction MAF)

Ce fichier EF sert à préconfigurer le nom de domaine complet de la fonction MAF à utiliser pour la connexion au service M2M après l'amorçage de service M2M. Si le Service N° 3 est disponible, ce fichier doit être présent. Ce fichier EF doit inclure un seul objet TLV.

Identificateur: '6F09'	Structure: transparente	Facultatif
Longueur: X octets	Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM		

Octets	Description	O/F	Longueur
1	Objet de données TLV du nom de domaine complet de la fonction MAF	O	X octets

Nom de domaine complet de la fonction MAF

Contenus:

Adresse du nom de domaine complet de la fonction MAF.

Codage:

Le nom de domaine complet de la fonction MAF sera codée en une chaîne d'octets conformément aux règles de codage UTF-8 décrites dans [IETF RFC 3629]. La valeur d'étiquette de l'objet de données TLV du nom de domaine complet de la fonction MAF doit être égale à 80.

D.1.3.9 EF_{MEFID} (identificateur de la fonction d'inscription M2M)

Ce fichier EF contient une ou plusieurs adresses de fonction d'inscription M2M. Le premier enregistrement du fichier EF doit être considéré comme ayant la priorité la plus élevée. Le dernier enregistrement du fichier EF doit être considéré comme ayant la priorité la plus faible. Si le Service N° 5 est disponible, ce fichier doit être présent.

Identificateur: '6F07'	Structure: linéaire fixe	Facultatif
Longueur de l'enregistrement: X octets	Fréquence de mise à jour: faible	
Conditions d'accès: LECTURE ALW MISE À JOUR ADM DÉSACTIVATION ADM ACTIVATION ADM		

Octets	Description	O/F	Longueur
1 à X	Objet de données LV de l'adresse de la fonction MEF	O	X octets

Objet de données LV de l'adresse de la fonction MEF

Contenus:

Adresse du MEF, sous la forme d'un nom de domaine complet, d'une adresse IPv4 ou IPv6.

Codage:

Le format de l'objet de données est le suivant:

Champ	Longueur (octets)
Longueur	1
Type d'adresse	1
Adresse MEF	Longueur de l'adresse

Type d'adresse: type de l'adresse de la fonction MEF.

Ce champ doit être égal au type de l'adresse de la fonction MEF conformément à ce qui suit:

Valeur	Nom
0x00	FQDN
0x01	IPv4
0x02	IPv6
Toutes les autres valeurs sont réservées	

Adresse de la fonction MEF: adresse de la fonction d'amorçage de service M2M.

Ce champ doit être égal à l'adresse de la fonction d'inscription M2M. Lorsque le type de fonction MEF est égal à 0x00, l'adresse de la fonction MEF correspondante doit être codée en une chaîne d'octets conformément aux règles de codage UTF-8 décrites dans [IETF RFC 3629].

La valeur "FF" est attribuée aux octets non utilisés.

D.2 Application du module de service oneM2M pour les justificatifs d'identité symétriques sur carte UICC (1M2MSM)

D.2.0 Introduction

Le présent paragraphe définit le module de service oneM2M (1M2MSM), une application utilisée pour les fonctionnalités de sécurité de la couche de services oneM2M et la fourniture d'abonnements sur la base de justificatifs d'identité symétriques. Cette application se trouve sur la carte UICC, carte à circuit intégré définie dans [ETSI TS 102 221]. En particulier, [ETSI TS 102 221] décrit les propriétés indépendantes de l'application de l'interface UICC/terminal, telles que les caractéristiques physiques et la structure logique. Il peut exister plusieurs fichiers ADF 1M2MSM sur une même carte UICC, correspondant à des abonnements indépendants aux services oneM2M.

D.2.1 Structure du fichier d'application du module de service oneM2M

Le présent paragraphe décrit les fichiers EF de la couche de services oneM2M qui définissent les conditions d'accès, les éléments de données et le codage. Un élément de données est une partie d'un fichier EF qui représente une entité logique complète.

D.2.1.1 Contenu des fichiers UICC au niveau du fichier principal

Les fichiers situés au niveau du fichier principal de la carte UICC sont indépendants de l'application, comme cela est décrit dans [ETSI TS 102 221]. Seuls les fichiers EF_{DIR} et EF_{ICCID} sont obligatoires sur la carte UICC pour les applications 1M2MSM. Dans tous les cas, tous les fichiers doivent être conformes à [ETSI TS 102 221].

D.2.1.2 Contenu des fichiers au niveau du fichier ADF 1M2MSM

Les fichiers EF du fichier ADF 1M2MSM contiennent les informations relatives à l'abonnement oneM2M qui sont requises pour les nœuds du champ M2M fonctionnant dans un environnement oneM2M. Ce fichier ADF doit être sélectionné à l'aide de son identificateur d'application (AID) et des informations du fichier EF_{DIR}. L'identificateur AID d'une application 1M2MSM doivent être formé conformément à [ETSI TS 101 220].

NOTE – L'identificateur de fournisseur d'application enregistré (RID) ETSI peut être utilisé pour les attributions en attente d'un identificateur RID oneM2M dédié dans [b-ISO/IEC 7816-5].

Les identificateurs de fichier "6F1X" (pour les fichiers EF), "5F1X" et "5F2X" (pour les fichiers DF), où X est compris entre "0" et "F", sont réservés au fichier ADF 1M2MSM pour l'usage administratif de l'émetteur de la carte.

La sous-structure du fichier $DF_{1M2M,}$ qui est utilisée pour isoler la configuration des informations liées au service M2M dépendant de l'accès au réseau et contenues dans un fichier ADF d'une application d'accès au réseau, n'est pas nécessaire pour la configuration indépendante du réseau d'accès d'un abonnement à un service M2M dans un fichier ADF 1M2MSM. Par conséquent, tous les fichiers EF définis au § D.1.3 doivent être présents au niveau du fichier ADF 1M2MSM. La structure du fichier du fichier ADF_{1M2MSM} est illustrée dans la Figure D.2.

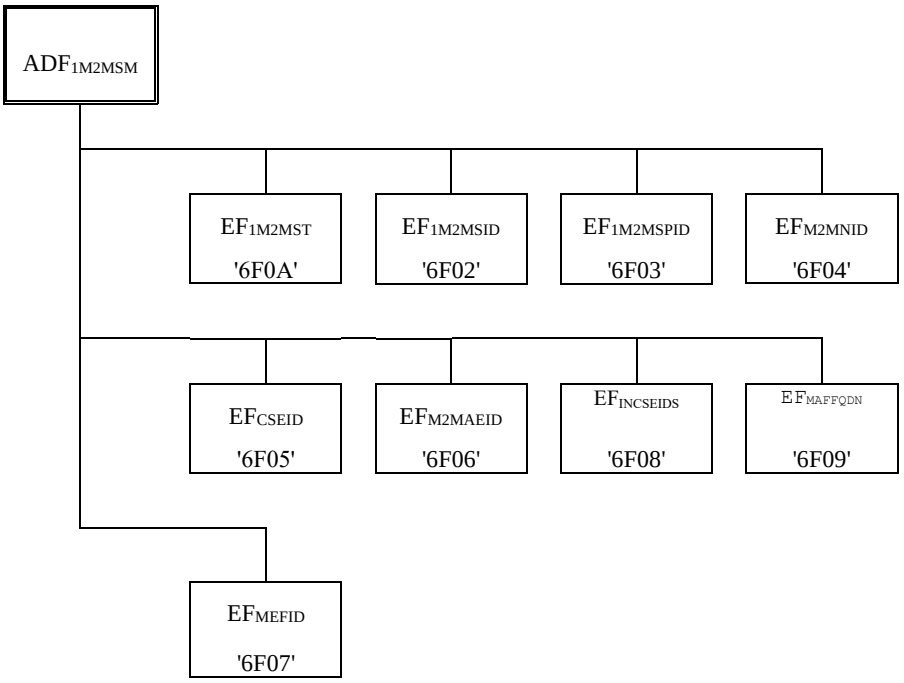


Figure D.2 – Identificateurs et structures de répertoire du fichier ADF_{1M2MSM}

D.2.2 Procédures relatives à l'abonnement au service M2M

D.2.2.0 Introduction

Le présent paragraphe décrit les procédures que doivent exécuter les nœuds du champ M2M pour interagir avec un abonnement à un service oneM2M sur une carte UICC. Sauf indication contraire, elles s'appliquent indépendamment de la structure de fichier prenant en charge l'abonnement au service oneM2M (fichier ADF 1M2MSM ou DF_{1M2M} dans le cadre d'un fichier ADF d'application d'accès au réseau).

D.2.2.1 Initialisation: sélection des applications 1M2MSM

Cette procédure ne s'applique qu'à un abonnement M2M pris en charge dans un fichier ADF 1M2MSM.

Si le nœud du champ M2M veut démarrer une opération M2M, il doit sélectionner une application 1M2MSM, après l'activation de la carte UICC (voir [ETSI TS 102 221]) et si l'application 1M2MSM est répertoriée dans le fichier EF_{DIR}, en utilisant la commande SELECT en fonction du nom du fichier DF comme défini dans [ETSI TS 102 221].

Après la sélection d'une application oneM2M, l'identificateur AID oneM2M sélectionné est stocké sur la carte UICC. Cette application est considérée comme la dernière application 1M2MSM sélectionnée. La dernière application 1M2MSM sélectionnée sera disponible sur la carte UICC une fois que cette dernière aura été désactivée, puis réactivée.

Si une application oneM2M est sélectionnée au moyen d'un nom de fichier DF partiel, le nom de fichier DF partiel fourni dans la commande doit identifier de manière univoque une application 1M2MSM. De plus, si une application 1M2M est sélectionnée au moyen d'un nom de

fichier DF partiel, comme cela est précisé dans [ETSI TS 102 221], en indiquant la dernière occurrence de la commande SELECT, la carte UICC sélectionnera l'application oneM2M stockée comme dernière application oneM2M. Si, dans la commande SELECT, les options "first" (premier), "next" (suivant) et "previous" (précédent) sont indiquées, ces dernières n'ont aucune signification si une application n'a pas été sélectionnée au préalable dans la même session et renvoient un code d'erreur correspondant.

D.2.2.2 Fermeture de session 1M2MSM

Cette procédure ne s'applique qu'à un abonnement M2M pris en charge dans un fichier ADF 1M2MSM.

La session UICC oneM2M est fermée par le nœud du champ M2M comme suit:

Le nœud du champ M2M doit indiquer à l'application UICC oneM2M que la procédure de fin de session est en train de démarrer, en envoyant une commande STATUS particulière.

Enfin, le nœud M2M supprime de sa mémoire tous les éléments d'information relatifs à l'abonnement M2M.

Pour mettre fin effectivement à la session, le nœud du champ M2M doit utiliser l'un des mécanismes décrits dans [ETSI TS 102 221].

D.2.2.3 Procédure de découverte de services oneM2M

Cette procédure sert à découvrir les services oneM2M associés offerts par une carte UICC oneM2M.

Le nœud de champ M2M exécute la procédure de lecture du tableau de services EF_{1M2MST}. Si aucun service associé à oneM2M n'est indiqué comme étant disponible, le nœud de champ M2M supposera que seule la fourniture de paramètres obligatoires est disponible dans ce fichier ADF.

D.2.2.4 Procédures de fourniture de services oneM2M

Ces procédures sont utilisées par un nœud de champ M2M pour amorcer un abonnement à un service M2M fourni sur la carte UICC.

Le nœud de champ M2M doit exécuter la procédure de lecture sur les éléments $EF_{1M2MSID}$, $EF_{1M2MSPID}$, et EF_{CSEID}, EF_{M2MNID}, $EF_{INCSEID}$ et $EF_{MAFFQDN}$ selon les services disponibles indiqués dans le tableau de services EF_{1M2MST}.

D.2.2.5 Procédure de fourniture d'identificateurs d'application oneM2M

Cette procédure fournit une liste d'identificateurs d'application M2M qui peuvent être activés sur le nœud M2M correspondant à l'abonnement au service oneM2M.

Condition: le service 4 doit être disponible dans le tableau de service oneM2M.

Dans ce cas, le nœud de champ M2M exécute la procédure de lecture du tableau de services $EF_{M2MAEID}$.

D.2.2.6 Procédures relatives à la fourniture sécurisée de l'architecture oneM2M

Ces procédures sont utilisées par le nœud de champ M2M pour effectuer une configuration sécurisée M2M à l'aide de la carte UICC, en fonction des services disponibles dans le fichier EF_{1M2MST} et des contextes de commandes AUTHENTICATE pris en charge (par exemple, prise en charge de l'architecture GBA par une application d'accès au réseau) indiqués pour le fichier ADF d'une application hôte.

Fourniture sécurisée: fourniture d'adresse MEF.

Condition: Le service 5 doit être disponible dans le tableau de service oneM2M.

Dans ce cas, le nœud de champ M2M exécute la procédure de lecture de l'élément EF_{MEFID} si le service associé est disponible.

Fourniture sécurisée GBA:

Cette procédure dépend du cadre d'authentification pris en charge par la carte UICC et est indiquée dans le tableau de services du fichier dédié d'application hôte.

Après avoir identifié le cadre d'authentification pris en charge, le nœud du champ M2M doit vérifier la disponibilité du service 7 dans le tableau de services $EF_{1M2MST.}$ Si le service est disponible, le nœud M2M D/G doit exécuter les procédures liées à l'architecture GBA dans un contexte de sécurité AUTHENTICATE-GBA (mode d'amorçage et de calcul) et avec les paramètres destinés à la fourniture sécurisée GBA.

D.2.2.7 Procédures relatives aux associations de sécurité oneM2M

Connexion sécurisée GBA:

Cette procédure dépend du cadre d'authentification pris en charge par la carte UICC et est indiquée dans le tableau de services du fichier dédié d'application hôte.

Après avoir identifié le cadre d'authentification pris en charge, le nœud du champ M2M doit vérifier la disponibilité du service 12 dans le tableau de services 12 dans le tableau de services EF_{1M2MST}. Si le service est disponible, le nœud de champ M2M doit exécuter les procédures liées à l'architecture GBA dans un contexte de sécurité AUTHENTICATE-GBA (mode d'amorçage et de calcul) et avec les paramètres destinés à l'association de sécurité GBA.

Annexe E

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe F

Obtention d'informations de localisation pour le contrôle d'accès fondé sur l'emplacement

(Cette annexe fait partie intégrante de la présente Recommandation.)

F.0 Introduction

Lorsqu'une demande (accès à une ressource) est évaluée par une entité de services communs (CSE) hôte et qu'un paramètre accessControlLocationRegions est défini dans l'attribut privileges des ressources <accessControlPolicy>, l'entité CSE hôte doit vérifier si l'emplacement de l'expéditeur d'une demande se trouve ou non dans les régions définies. Par conséquent, l'entité CSE hôte doit conserver l'emplacement de l'expéditeur ou bien l'obtenir ou refuser l'accès. La présente Annexe expose la manière de décrire les régions de localisation et d'obtenir l'emplacement de l'expéditeur.

F.1 Description d'une région

F.1.1 Description circulaire

La méthode pratique pour décrire la région ou la zone consiste à recourir à un cercle. Ce dernier se caractérise généralement par les coordonnées de son point central et par son rayon. Géographiquement, le point central et le rayon sont respectivement exprimés par une longitude et une latitude, d'une part, et en mètre, d'autre part. Pour cette description, le paramètre accessControlLocationRegions est représenté sous la forme d'un cercle, comme celui qui est illustré dans la Figure F.1.

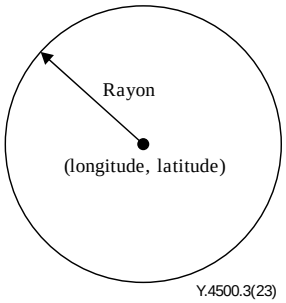


Figure F.1 – Représentation circulaire du paramètre accessControlLocationRegions

F.1.2 Description du pays

Si une description de pays est utilisée, il s'agira d'un code ISO-3166-1 alpha 2, tel que défini dans [b-ISO 3166-1]. Ces codes sont composés de deux lettres et utilisés pour représenter les pays et les régions présentant un intérêt géographique particulier. Par exemple, le code "KR" correspond à la Corée (République de).

NOTE – La norme [b-ISO 3166-1] dispose ce qui suit: "les éléments de code attribués par les utilisateurs sont des codes mis à la disposition de ces derniers s'ils ont besoin d'ajouter d'autres noms de pays, de territoires ou d'autres entités géographiques aux fins d'une application particulière de la norme ISO 3166-1. La norme ISO 3166/MA n'utilisera jamais ces codes si celle-ci venait à être mise à jour." L'utilisateur peut attribuer les codes suivants:

– *Alpha-2: AA, de QM à QZ, de XA à XZ, et ZZ;*

– *Alpha-3: de AAA à AAZ, de QMA à QZZ, de XAA à XZZ, et de ZZA à ZZZ;*

– *Numérique: de 900 à 999.*

L'architecture oneM2M n'empêchera pas les utilisateurs d'utiliser la fonctionnalité des codes qu'ils définissent, mais n'enregistrera pas ces codes. Il existe donc un risque de duplication dans les mises en œuvre, pouvant ainsi entraîner un fonctionnement incohérent et des problèmes d'interopérabilité entre les mises en œuvre.

F.2 Obtention d'informations de localisation

F.2.0 Introduction

Comme mentionné ci-dessus, lorsque le paramètre accessControlLocationRegions est défini, l'entité CSE hôte doit vérifier l'emplacement de l'expéditeur pour assurer le contrôle d'accès. Le présent paragraphe décrit comment l'entité CSE hôte vérifie ou obtient la localisation. Les procédures peuvent varier selon la description de la région, le cercle et le pays.

F.2.1 Description circulaire

Si la description circulaire est utilisée comme contrainte de contexte de localisation, l'entité CSE hôte vérifie si elle possède ou non l'emplacement actuel de l'expéditeur. Si ce n'est pas le cas, elle doit obtenir l'emplacement de l'expéditeur. La Recommandation [UIT-T Y.4500.1] définit un type de ressource destiné à l'acquisition de l'emplacement d'un nœud cible, à savoir <locationPolicy>. Par conséquent, afin d'obtenir l'emplacement de l'expéditeur, l'entité CSE hôte doit créer une ressource <locationPolicy> et définir les attributs pertinents comme suit:

locationSource: la fiabilité des informations de localisation étant cruciale, la localisation doit être obtenue auprès du réseau de confiance. Si la localisation est obtenue par les autres sources, elle peut être facilement usurpée (piratage GPS). Par conséquent, la valeur "network-based" doit être affectée à l'attribut locationSource.

locationTargetID: le nœud cible est l'expéditeur qui a besoin d'autoriser les demandes envoyées. L'identificateur de l'expéditeur doit être affecté à l'attribut locationTargetID.

Il est à noter que les autres attributs sont déterminés par les politiques locales de l'entité CSE hôte décrites au § 9.6.9 de la Recommandation [UIT-T Y.4500.1]. Afin d'obtenir la localisation à partir du réseau, l'entité CSE hôte transforme la demande de localisation formulée par l'architecture oneM2M en une demande formulée par le réseau.

NOTE 1 – L'Annexe F de la Recommandation [UIT-T Y.4500.4] décrit comment convertir la demande formulée par l'architecture oneM2M en une demande de localisation de terminal formulée par une interface de programmation d'application NetAPI REST OMA.

Étant donné que les informations de région (description circulaire) sont définies par le paramètre accessControlLocationRegions, l'entité CSE hôte peut utiliser les informations de région circulaire lorsqu'elle demande les informations de localisation au réseau. La spécification [b-OMA REST] relative à l'interface de programmation d'application NetAPI REST OMA destinée à la localisation des terminaux définit des types de ressources sous la forme d'un service de notification de localisation fondé sur une zone (région), nommé "CircleNotificationSubscription". Donc, si l'entité CSE hôte s'abonne au service de notification en précisant une région circulaire définie par le paramètre accessControlLocationRegions, elle peut toujours déterminer si l'expéditeur se trouve ou non dans les régions définies. La Figure F.2 montre comment obtenir l'emplacement de l'expéditeur lorsque le paramètre accessControlLocationRegions est défini.

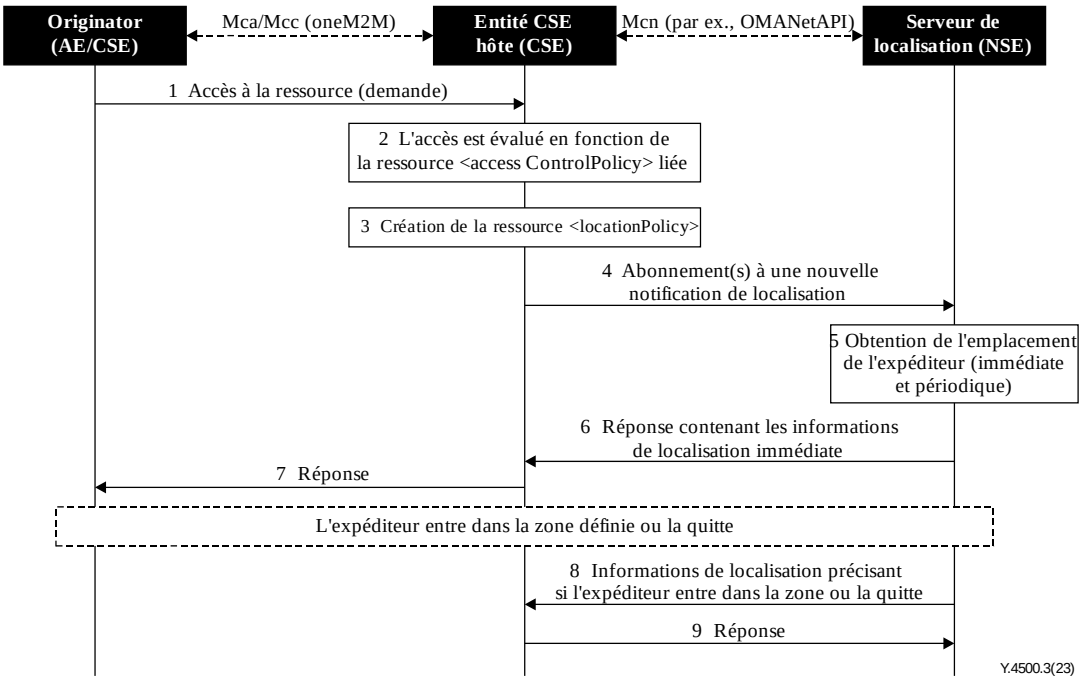


Figure F.2 – Emplacement de l'expéditeur lorsque le paramètre accessControlLocationRegions est défini

1 L'expéditeur envoie une demande d'accès à une ressource.

2 L'entité CSE hôte doit évaluer la demande reçue en la comparant à la ressource <accessControlPolicy> liée. Si l'un des tuples de règles concernant l'expéditeur de la demande contient le paramètre accessControlLocationRegions (description circulaire) et que l'entité CSE hôte ne mémorise pas l'emplacement de l'expéditeur, cette dernière passe à l'étape suivante ou refuse l'accès. Si l'entité CSE hôte détient l'emplacement de l'expéditeur, celui-ci est utilisé pour appliquer la politique de contrôle d'accès.

L'entité CSE hôte peut refuser l'accès si l'expéditeur n'est pas abonné au réseau ou pour toute autre raison (par exemple, à la suite d'une perte de connexion ou d'un dysfonctionnement du serveur).

3 L'entité CSE hôte crée la ressource <locationPolicy> et définit les attributs pertinents comme mentionné ci-dessus.

4 L'entité CSE hôte s'abonne à un nouveau service de notification de localisation de zone auprès du serveur de localisation du réseau. Les informations de zone reposent sur la zone définie par les paramètres accessControlLocationRegions. Si plusieurs régions sont définies, plusieurs abonnements doivent être activés.

5 Le serveur de localisation obtient immédiatement l'emplacement de l'expéditeur.

NOTE 2 – Après l'obtention immédiate de l'emplacement, le serveur de localisation obtient périodiquement l'emplacement de l'expéditeur pour vérifier si ce dernier se trouve ou non dans la zone. Des politiques locales peuvent définir la fréquence et la durée de cette procédure.

6 Le serveur de localisation répond en envoyant à l'entité CSE hôte les informations de localisation immédiate concernant l'expéditeur.

7 En fonction de l'emplacement de l'expéditeur reçu et d'autres politiques de contrôle d'accès, la demande est évaluée et peut être acceptée ou refusée. L'entité CSE hôte donne une réponse à la demande (étape 1).

8 Lorsque l'expéditeur pénètre dans la zone (entrée) ou la quitte (sortie), le serveur de localisation notifie à l'entité CSE hôte le changement d'emplacement. Ainsi, l'entité CSE hôte peut suivre l'emplacement de l'expéditeur et contrôler facilement l'accès en fonction de la contrainte de contexte d'emplacement.

9 L'entité CSE hôte répond à la notification.

F.2.2 Description du pays

Généralement, la localisation l'expéditeur à l'échelle du pays peut être déterminée par son adresse IP. Si l'entité CSE hôte peut distinguer le pays à l'aide de l'adresse IP de l'expéditeur et que le pays correspond au paramètre acccessControlLocationRegions défini, l'entité CSE hôte peut donner suite à la demande sous réserve d'une évaluation de l'ensemble des politiques de contrôle d'accès.

NOTE 1 – La manière de déduire le pays à partir de l'adresse IP ne relève pas du domaine d'application de la présente Recommandation.

Toutefois, si l'entité CSE hôte ne peut pas distinguer le pays à l'aide de l'adresse IP de l'expéditeur, le réseau lui envoie les coordonnées de l'emplacement (c'est-à-dire la longitude et la latitude) de l'expéditeur. L'entité CSE hôte peut ainsi distinguer le pays en utilisant l'emplacement s'il est disponible. La façon d'obtenir les coordonnées de l'emplacement est exposée dans l'Annexe F de la Recommandation [UIT-T Y.4500.4].

NOTE 2 – La manière de déduire le pays à partir de l'emplacement ne relève pas du domaine d'application de la présente Recommandation.

Annexe G

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe H

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe I

Annexe laissée en blanc

Cette annexe est intentionnellement laissée en blanc.

Annexe J

Liste des attributs de confidentialité

(Cette Annexe fait partie intégrante de la présente Recommandation.)

Voir le Tableau J.1.

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
1.0	Who	Valeur nulle	Valeur nulle	Nom de l'entité	Nom commercial du dispositif ou du fournisseur de services demandant l'accès aux dispositifs/réseaux/données intelligents des utilisateurs.	
		Variable	Texte	Nom de l'entreprise	Nom de l'entreprise qui demande l'accès aux dispositifs intelligents de l'utilisateur et qui précise leurs conditions.	
		Variable à deux lettres	Code de pays	Localisation	Pays où se trouve le dispositif ou le fournisseur de services.	
		Variable	Texte	Numéro d'enregistrement d'entreprise	Numéro d'enregistrement d'entreprise qui peut servir à vérifier l'authenticité de l'entreprise. Les autorités nationales des différents pays peuvent permettre de vérifier l'authenticité de l'entreprise qui demande à utiliser les données et sa fiabilité.	
1.1	ID				Options concernant l'identification univoque des applications et des dispositifs.	
			Texte	Numéro de modèle	Numéros de mobile des dispositifs inclus dans le service du fournisseur de services d'application (ASP).	
			Texte	Version	Numéros de version des dispositifs inclus dans le service de l'ASP	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
			Format défini de l'architecture one M2M	Identificateur enregistré des applications	Identificateur enregistré des applications incluses dans le service de l'ASP.	
			Codes de pays	Codes de pays pour lesquels une approbation a été accordée, le cas échéant	L'accréditation d'un dispositif ou d'une application ne peut être valable que dans certains pays.	
2.0	What			Type de classification de données	Type de données accessibles par le dispositif/service. Plus la valeur est élevée, plus les données sont sensibles.	
		Aucune donnée collectée	Oui/non	Aucune donnée collectée	Le dispositif ne collecte aucune donnée. Il peut s'agir d'un dispositif de sortie, tel qu'un commutateur lumineux, qui ne reçoit que des instructions.	
		Données non personnelles	Oui/non	Données qui ne sont liées à personne	Les données ne peuvent être liées à personne. Cela peut être le cas si le dispositif est, par exemple, un capteur de porte qui ne peut indiquer que si cette dernière est ouverte ou fermée.	
		Données anonymisées	Oui/non	Données anonymisées recueillies sur une personne	Données recueillies sur une personne, mais anonymisées de sorte à résumer toutes les données qui permettraient d'identifier un individu ou à les supprimer.	
		Données personnelles	Oui/non	Données qui peuvent être directement liées à un identificateur	Données collectées qui peuvent être liées à un identificateur propre à une personne ou à un groupe réduit de personnes (par exemple, aux membres d'un même foyer).	
		Données personnelles sensibles	Oui/non	Données plus sensibles qui peuvent être liées pour identifier une personne	L'autorité de l'Union européenne chargée de la protection des données définit plusieurs types de données personnelles sensibles. Il convient de considérer d'autres types de services, tels que les services bancaires, comme s'inscrivant dans ce domaine.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Données médicales	Oui/non	Données relatives, entre autres, à la santé d'une personne	Données concernant les maladies et l'état de santé général d'une personne ainsi que sur les soins qu'elle reçoit.	
3.0	When	Valeur nulle	Paramètre nul	Période de collecte des données	Fréquence de collecte des données.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Fondé sur les événements	Oui/non	Déclenchement par un événement	Le dispositif ne collecte des données qu'à la suite d'un événement (par exemple, un capteur de porte déclenche une caméra).	
		Mois	1 à 12	Les données sont envoyées de façon mensuelle	Les dispositifs/services ne collectent les données qu'au cours de transferts mensuels (par exemple, un réfrigérateur intelligent qui envoie un rapport de routine concernant son état de fonctionnement).	
		Hebdoma-daire	Oui/non	Les données sont envoyées toutes les semaines	Les dispositifs/services ne collectent les données qu'au cours de transferts hebdomadaires. Par exemple, un rapport sur l'état de diagnostic d'un système de détection d'incendie, y compris les résultats des tests du capteur, estime la durée de vie restante de la batterie.	
		Quotidien	Oui/non	Les données sont envoyées tous les jours	Les dispositifs/services ne collectent les données qu'au cours d'un transfert quotidien. Par exemple, un réfrigérateur intelligent envoie au détaillant sélectionné par l'utilisateur une liste énumérant les articles dont ce dernier a besoin afin que ledit détaillant puisse les répertorier et les inclure dans le panier de l'utilisateur.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Horaire	1 à 24	Données sont envoyées tous les X heures	Les dispositifs/services ne collectent les données qu'au cours d'un transfert horaire. Par exemple, une alarme domestique signale qu'elle est enclenchée et que tous les capteurs sont actifs. Ainsi, le service de surveillance de l'alarme et des applications qui y sont liées sait que le système est toujours opérationnel et que personne n'a désactivé la fonctionnalité d'envoi d'alertes.	
		Minutes	1 à 60	Données sont envoyées tous les X minutes	Les dispositifs/services ne collectent les données que toutes les 15 minutes (par exemple, compteurs intelligents qui rapportent les relevés de consommation).	
		Temps réel fondé sur un événement	Oui/non	Les données sont envoyées en continu à la suite d'un événement	Les données sont envoyées en temps réel lorsqu'un événement particulier se produit. Par exemple, l'alarme activée d'une maison qui signale l'ouverture d'une porte intérieure, déclenchant ainsi la diffusion du flux des caméras de sécurité.	
		Temps réel permanent	Oui/non	Les données sont envoyées en continu à tous moments	Les données sont envoyées en temps réel tant que le dispositif est actif. Par exemple, données d'un système de télévision en circuit fermé envoyées dans un espace de stockage externe.	
3.1	Time period	Valeur nulle		Période d'envoi des données	Moment auquel les données sont envoyées.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Résumé/état actuel	Oui/non	Indique si le dispositif envoie ou non les données relatives à son état actuel	Le dispositif envoie les données concernant son état actuel, sans historique. Par exemple, état actuel d'un capteur de porte (ouvert/fermé) et non le journal consignant l'ouverture et la fermeture de la porte.	
		Échantillon	Oui/non	Données collectées sur une période réduite	L'échantillon de données couvre une période réduite. Par exemple, échantillons de données d'une fréquence cardiaque collectés plusieurs fois par jour.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Historique complet	Oui/non	La totalité des données collectées par le dispositif est fournie.	La totalité des données collectées par le dispositif est fournie et envoyée en temps réel (3.0) ou bien l'historique est chargé rétrospectivement.	
3.2	Fréquence de collecte d'échantil-lon	Valeur nulle		Durée séparant chaque collecte d'échantillon	Durée exprimée en secondes s'écoulant entre chaque collecte d'échantillon.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Variable		Nombre de secondes s'écoulant entre chaque point de collecte de données	Durée en secondes s'écoulant entre chaque mesure du dispositif.	
		Streamed data	Oui/non	Les données sont collectées en continu	Les données sont collectées en continu (par exemple, caméra de sécurité intelligente pouvant diffuser son flux vidéo à l'utilisateur).	
4.0	Where – stored	Valeur nulle		Emplacement de stockage des données	Emplacement de stockage des données créées par le dispositif ou utilisées par le service.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Local	Oui/non	Les données ne sont stockées que localement.	Les données sont stockées au sein du réseau des dispositifs intelligents (par exemple, réseau domestique).	
		Variable		Pays/bloc	Indique le pays où sont stockées les données ou si elles font partie d'une organisation plus large (comme l'UE).	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
4.1	Where – collected (origine de la collecte)			Origine des données collectées	Emplacement à partir duquel les données sont collectées. Il convient de noter que ceci peut être redondant pour les consommateurs, mais peut être utilisé pour des flux externes tels que les bulletins météorologiques. Cela peut aussi être utile à des services afin qu'ils puissent déclarer les essais de dispositifs à partir desquels ils collectent des données, s'ils ne souhaitent pas avoir accès à tous les dispositifs intelligents sur le site.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Dispositif	Oui/non	Les données sont collectées uniquement sur le dispositif spécifique visé aux conditions générales.	Les conditions (ainsi que les paramètres de confidentialité des utilisateurs) ne sont comparées qu'aux données collectées par le dispositif en question.	
		Réseau de dispositifs intelligents	Oui/non	Les données sont recueillies à partir de tous les dispositifs du réseau de l'utilisateur.	Les données sont collectées auprès de tous les dispositifs* qui forment le réseau de dispositifs intelligents des utilisateurs.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Variable		Flux externe	Les données proviennent d'une source externe et sont combinées aux données collectées. Par exemple, prévisions météorologiques combinées à la fréquentation d'un bâtiment pour chauffer ce dernier de sorte qu'il soit à la température souhaitée lorsque l'usager y entre.	Cette valeur serait descriptive et l'utilisateur aurait deux options: désactiver ou remplacer (par exemple, s'il possède une station météorologique personnelle et ne récupère pas les prévisions auprès d'un service météorologique).
4.2	Where – Processed (lieu du traitement)	Valeur nulle		Lieu de traitement des données	Lieu physique où les données sont traitées. Il peut être différent de l'emplacement de stockage.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Local	Oui/non	Les données ne sont traitées que localement.	Les données ne sont traitées que sur le dispositif ou sur le réseau des dispositifs intelligents de l'utilisateur.	
		Variable		Pays/bloc	Indique le pays où sont stockées les données ou si elles font partie d'une organisation plus large (comme l'UE).	
4.3	Where – Accessed (lieu d'accès)	Valeur nulle		Emplacement à partir duquel les données sont accessibles	Emplacement à partir duquel le fournisseur ou les restrictions de priorité de politiques permettent d'accéder aux données stockées.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Aucune donnée collectée	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Local	Oui/non	Les données ne sont traitées que localement.	Les données ne sont traitées que sur le dispositif ou sur le réseau des dispositifs intelligents de l'utilisateur.	
		Variable		Pays/bloc	Indique le pays où sont stockées les données ou si elles font partie d'une organisation plus large (comme l'UE).	
5.0	Why (motif)	Valeur nulle			Première raison de la collecte des données personnelles. Cela permet de signaler à l'utilisateur tout changement d'usage de celles-ci.	
		Oui/non	Oui/non	Pour fourniture directe du service	Le fournisseur de service d'application (ASP) collecte les informations nécessaires à la fourniture directe du service.	Par exemple, en utilisant la localisation pour la radiomessagerie depuis une station de base à laquelle l'utilisateur est actuellement inscrit.
		Oui/non	Oui/non	Amélioration des produits et services de l'ASP et de ses partenaires	L'ASP collecte les informations nécessaires pour améliorer ses produits et services et ceux de ses partenaires.	
		Oui/non	Oui/non	Personnalisation des services	L'ASP collecte les informations nécessaires pour personnaliser ses produits et services et ceux de ses partenaires.	"Les clients qui ont sélectionné cela ont aussi sélectionné ces éléments."

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Oui/non	Oui/non	Exigence relative à la priorité des politiques	L'ASP collecte les informations nécessaires pour satisfaire à l'exigence relative à la priorité des politiques.	Par exemple, l'âge minimal requis de l'utilisateur afin de déterminer si ce dernier peut accéder aux ressources.
6.0	Retention (conser-vation)	Valeur nulle		Durée de conservation des données	Durée pendant laquelle les données (définies ci-dessus) sont conservées dans leur niveau de détail actuel.	
		Données non collectées	Oui/non	Aucune donnée collectée	Le dispositif/service, par exemple un dispositif terminal comme une lampe, ne collecte pas de données.	
		Aucune conservation	Oui/non	Aucune donnée n'est conservée	Une fois qu'elles ont été traitées, les données sont immédiatement supprimées.	
		Minutes	1 à 60	Les données sont conservées pendant X minutes	Les données sont conservées pendant 15 minutes avant d'être supprimées. Par exemple, le dispositif ne conserve que le dernier ensemble de mesures et en collecte de nouvelles toutes les 15 minutes.	
		Heure	1 à 24	Les données sont conservées pendant X heures.	Les données sont conservées pendant X heures.	
		Jour	1 à 7	Les données sont conservées pendant X jours		
		Semaine	1 à 4	Les données sont conservées pendant X semaines		
		Mois	1 à 12	Les données sont conservées pendant X mois		

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Année	1 à 10	Les données sont conservées pendant X ans		
		Conservation permanente		Les données sont conservées de façon permanente	Les données sont stockées sans politique de conservation ou de suppression définie.	
6.1	retention – anony-mized (conser-vation des données anony-misées)	Valeur nulle		Durée de conservation des données anonymisées	Durée pendant laquelle sont conservées les données anonymisées ou toute autre donnée calculée qui n'est pas directement liée à une identité unique. Par exemple, statistiques relatives à la consommation électrique en fonction d'un emplacement.	
		Aucune conservation	Oui/non	Aucune donnée n'est conservée	Une fois qu'elles ont été traitées, les données sont immédiatement supprimées.	
		Minutes	1 à 60	Les données sont conservées pendant X minutes.	Les données sont conservées pendant 15 minutes avant d'être supprimées. Par exemple, le dispositif ne conserve que le dernier ensemble de mesures et en collecte de nouvelles toutes les 15 minutes.	
		Heure	1 à 24	Les données sont conservées pendant X heures.	Les données sont conservées pendant X heures.	
		Jour	1 à 7	Les données sont conservées pendant X jours.		
		Semaine	1 à 4	Les données sont conservées pendant X semaines.		
		Mois	1 à 12	Les données sont conservées pendant X mois.		

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Année	1 à 10	Les données sont conservées pendant X ans.		
		Conservation permanente		Les données sont conservées de façon permanente.	Les données sont stockées sans politique de conservation ou de suppression définie.	
6.2	retention – summary (conservation du récapitulatif des données)	Valeur nulle		Durée de conservation du récapitulatif des données.	Durée de conservation du récapitulatif des données (par exemple, consommation électrique totale mensuelle calculée à l'aide de relevés effectués toutes les 15 minutes par le compteur).	
		Aucune conservation	Oui/non	Aucune donnée n'est conservée.	Une fois qu'elles ont été traitées, les données sont immédiatement supprimées.	
		Minutes	1 à 60	Les données sont conservées pendant X minutes.	Les données sont conservées pendant 15 minutes avant d'être supprimées. Par exemple, le dispositif ne conserve que le dernier ensemble de mesures et en collecte de nouvelles toutes les 15 minutes.	
		Heure	1 à 24	Les données sont conservées pendant X heures.	Les données sont conservées pendant X heures.	
		Jour	1 à 7	Les données sont conservées pendant X jours		
		Semaine	1 à 4	Les données sont conservées pendant X semaines.		
		Mois	1 à 12	Les données sont conservées pendant X mois.		

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Année	1 à 10	Les données sont conservées pendant X ans.		
		Conservation permanente		Les données sont conservées de façon permanente.	Les données sont stockées sans politique de conservation ou de suppression définie.	
7.0	Sharing – full (partage de l'intégralité des données)	Valeur nulle		Personnes avec lesquelles les données sont partagées.	Personnes extérieures à l'entreprise qui ont accès à l'intégralité des données en fonction de leur type.	
		Données non partagées	Oui/non	Les données ne sont pas partagées en dehors de l'entreprise.	Les données ne sont ni partagées en dehors de l'entreprise qui fournit le dispositif/service ni traitées par un tiers.	
		Groupe	Non/portée et motif	Les données ne sont partagées qu'avec des entreprises du même groupe.	Les données ne sont partagées qu'au sein d'autres entreprises du même groupe.	
		Fournisseur d'infrastruc-tures	Oui/non	Les données sont stockées sur l'infrastructure d'un tiers.	Les données sont stockées sur les serveurs d'une entreprise distincte, l'entreprise qui fournit le dispositif/service pouvant, par exemple, faire appel à un fournisseur de services en nuage pour le stockage ou le traitement des données.	Note – L'emplacement de stockage ou de traitement des données dépend du tiers ainsi que de l'entreprise qui offre le service/dispositif.
		Sous-traitant	Oui/non	Les données sont partagées avec des sous-traitants.	Les données sont partagées avec un ou plusieurs sous-traitants qui fournissent une partie du service.	Note – Les informations concernant entre autres l'emplacement dépendent de chaque sous-traitant.

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Fonctions auxiliaires des autres tiers sous contrat	Non/portée et motif	Les données sont partagées avec des tiers sous contrat.	Les données sont partagées avec des tiers sous contrat qui assurent des fonctions supplémentaires (non essentielles) pour utiliser un dispositif ou en tirer parti, comme l'envoi de bulletins d'information ou d'offres promotionnelles, ou encore l'indication de centres de réparation compris dans la garantie.	
		Partenaires	Non/portée et motif	Les données sont partagées avec des tiers privés.	Les données sont partagées avec d'autres tiers n'ayant aucun lien direct ou indirect avec le dispositif/service. Par exemple, les fournisseurs de dispositifs partagent des données avec leurs partenaires commerciaux dans le cadre de campagnes ciblées.	
		Organismes publics	Non/portée, motif et organismes en question	Les données sont partagées avec des organismes publics importants.	Les données sont partagées avec certains organismes sous certaines conditions. Par exemple, lorsqu'une alarme se déclenche, les données relatives aux dispositifs de sécurité sont partagées avec la police afin qu'elles puissent intervenir de façon appropriée. Ou, si une alarme médicale se déclenche, des informations sont transmises aux services d'ambulance ou à un hôpital afin qu'ils puissent intervenir.	
7.1	Sharing – anony-mized (partage des données anony-misées)	Valeur nulle		Personnes avec lesquelles les données sont partagées	Personnes extérieures à l'entreprise qui ont accès aux données anonymisées en fonction du type d'utilisateur.	
		Données non partagées	Oui/non	Les données ne sont pas partagées en dehors de l'entreprise.	Les données ne sont ni partagées en dehors de l'entreprise qui fournit le dispositif/service ni traitées par un tiers.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Groupe	Non/portée et motif	Les données ne sont partagées qu'avec des entreprises du même groupe.	Les données ne sont partagées qu'au sein d'autres entreprises du même groupe.	
		Fournisseur d'infras-tructures	Oui/non	Les données sont stockées sur l'infrastructure d'un tiers.	Les données sont stockées sur les serveurs d'une entreprise distincte. Par exemple, l'entreprise qui fournit le dispositif/service fait appel à un fournisseur de services en nuage pour le stockage ou le traitement des données.	
		Sous-traitant	Oui/non	Les données sont partagées avec des sous-traitants.	Les données sont partagées avec un ou plusieurs sous-traitants qui fournissent une partie du service.	
		Fonctions auxiliaires des autres tiers sous contrat	Non/portée et motif	Les données sont partagées avec des tiers sous contrat.	Les données sont partagées avec des tiers sous contrat qui assurent des fonctions supplémentaires (non essentielles) pour utiliser un dispositif ou en tirer parti, comme l'envoi de bulletins d'information ou d'offres promotionnelles, ou encore l'indication de centres de réparation compris dans la garantie.	
		Partenaires	Non/portée et motif	Les données sont partagées avec des tiers privés.	Les données sont partagées avec d'autres tiers n'ayant aucun lien direct ou indirect avec le dispositif/service. Par exemple, les fournisseurs de dispositifs partagent des données avec leurs partenaires commerciaux dans le cadre de campagnes ciblées.	
		Organismes publics	Non/portée, motif et organismes en question	Les données sont partagées avec des organismes publics importants.	Les données sont partagées avec certains organismes sous certaines conditions. Par exemple, lorsqu'une alarme se déclenche, les données relatives aux dispositifs de sécurité sont partagées avec la police afin qu'elles puissent intervenir de façon appropriée. Ou, si une alarme médicale se déclenche, des informations sont transmises aux services d'ambulance ou à un hôpital afin qu'ils puissent intervenir.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
7.2	Sharing – summary (partage du récapitulatif des données)	Valeur nulle		Personnes avec lesquelles les données sont partagées	Personnes extérieures à l'entreprise qui ont accès au récapitulatif des données en fonction du type d'utilisateur.	
		Données non partagées	Oui/non	Les données ne sont pas partagées en dehors de l'entreprise.	Les données ne sont ni partagées en dehors de l'entreprise qui fournit le dispositif/service ni traitées par un tiers.	
		Groupe	Non/portée et motif	Les données ne sont partagées qu'avec des entreprises du même groupe.	Les données ne sont partagées qu'au sein d'autres entreprises du même groupe.	
		Fournisseur d'infrastructures	Oui/non	Les données sont stockées sur l'infrastructure d'un tiers.	Les données sont stockées sur les serveurs d'une entreprise distincte. Par exemple, l'entreprise qui fournit le dispositif/service fait appel à un fournisseur de services en nuage pour le stockage ou le traitement des données.	
		Sous-traitant	Oui/non	Les données sont partagées avec des sous-traitants.	Les données sont partagées avec un ou plusieurs sous-traitants qui fournissent une partie du service.	
		Fonctions auxiliaires des autres tiers sous contrat	Non/portée et motif	Les données sont partagées avec des tiers sous contrat.	Les données sont partagées avec des tiers sous contrat qui assurent des fonctions supplémentaires (non essentielles) pour utiliser un dispositif ou en tirer parti, comme l'envoi de bulletins d'information ou d'offres promotionnelles, ou encore l'indication de centres de réparation compris dans la garantie.	
		Partenaires	Non/portée et motif	Les données sont partagées avec des tiers privés.	Les données sont partagées avec d'autres tiers n'ayant aucun lien direct ou indirect avec le dispositif/service. Par exemple, les fournisseurs de dispositifs partagent des données avec leurs partenaires commerciaux dans le cadre de campagnes ciblées.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Organismes publics	Non/portée, motif et organismes en question	Les données sont partagées avec des organismes publics importants.	Les données sont partagées avec certains organismes sous certaines conditions. Par exemple, conseils municipaux qui collectent les données relatives à la consommation moyenne d'eau en fonction de la localisation.	
8.0	informing (informa-tion)	Valeur nulle				
		Oui/non	Oui/non	Conditions générales envoyées à l'adresse électronique enregistrée par l'utilisateur final.	Les fournisseurs d'applications de vendeurs de dispositifs envoient leurs valeurs d'étiquette dans ce tableau à une adresse électronique enregistrée par l'utilisateur final.	
		URL	ACME.com/français/ type de dispositif/modèle/ conditions générales	Conditions générales sont disponibles à l'URL affichée	Les fournisseurs d'applications de vendeurs de dispositifs rendent les valeurs d'étiquette figurant dans ce tableau disponibles à une adresse URL.	Le portail de gestion des politiques de confidentialité (PPM) peut effectuer automatiquement le traitement.
		URL	ACME.com/français/ type de dispositif/modèle/ conditions générales	Les conditions générales sont disponibles à une URL stockée sur le dispositif	Les fournisseurs d'applications de vendeurs de dispositifs rendent les valeurs d'étiquette figurant dans ce tableau disponibles à une adresse URL stockée sur le dispositif.	Le portail PPM ou le dispositif peut effectuer automatiquement le traitement.
		Oui/non	Oui/non	Conditions générales affichées à l'écran (le cas échéant)	Les fournisseurs d'applications de vendeurs de dispositifs affichent les valeurs d'étiquette figurant dans ce tableau disponibles sur l'écran du dispositif (le cas échéant).	
		Oui/non	Oui/non	Affichage sur un écran distant associé à l'utilisateur	Les fournisseurs d'applications de vendeurs de dispositifs affichent les valeurs d'étiquette figurant dans ce tableau sur un écran distant associé à l'utilisateur.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Oui/non	Oui/non	Par courrier postal	Les fournisseurs d'applications de vendeurs de dispositifs envoient leurs valeurs d'étiquette dans ce tableau à une adresse postale enregistrée par l'utilisateur final.	
		Oui/non	Oui/non	Par SMS	Les fournisseurs d'applications de vendeurs de dispositifs envoient leurs valeurs d'étiquette dans ce tableau à un numéro de téléphone mobile enregistré par l'utilisateur final.	
9.0	Obtaining consent (obtention du consen-tement)	Valeur nulle				
			Oui/non	Consentement par défaut dans l'application	L'utilisateur doit donner son consentement par défaut en effectuant une simple action dans l'application.	
			Oui/non	Consentement donné à l'aide d'un document signé par l'utilisateur final	Fichier XML récapitulatif signé à l'aide de la clé privée de l'utilisateur final (par exemple, signature numérique).	
			Oui/non	Consentement à l'aide de la méthode recommandée pour l'architecture oneM2M	Méthode recommandée pour l'architecture oneM2M {à compléter}	Applicable si cela est défini ultérieurement
10.0	Protection	Valeur nulle			Il existe cinq niveaux de protection de la vie privée et de la sécurité de l'utilisateur final, le niveau 1 étant le plus faible et le niveau 5 le plus élevé. Chacun de ces niveaux implique des exigences qui lui sont propres.	Ces niveaux sont alignés sur ceux déjà proposés par le Groupe de travail 4 de l'organisation oneM2 M.

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Niveau de protection annoncé	1	Niveau de protection annoncé = 1	Niveau 1: niveau le plus bas en matière de protection de la vie privée et la sécurité de l'utilisateur final. Ce niveau est utilisé lorsque le risque associé à une violation de la sécurité et de la vie privée de l'utilisateur final est minimal.	
		Niveau de protection annoncé	2	Niveau de protection annoncé = 2	Niveau 2: permet d'assurer dans une certaine mesure que la vie privée et la sécurité de l'utilisateur final sont protégées. Les entités prouvent, au moyen d'un protocole d'authentification sécurisé, que l'entité a le contrôle des données sensibles ou des justificatifs d'identité. Des contrôles sont mis en place pour assurer la protection des données sensibles ou des justificatifs d'identité stockés contre toute attaque les visant.	
		Niveau de protection annoncé	3	Niveau de protection annoncé = 3	Niveau 3: permet d'assurer avec une confiance élevée que la vie privée et la sécurité de l'utilisateur final sont protégées. Ce niveau est nécessaire lorsqu'un risque important est associé à une violation de la sécurité et de la vie privée de l'utilisateur final. L'authentification multifactorielle est utilisée. Toute donnée ou information sensible échangée durant les protocoles d'authentification est protégée par un chiffrement, qu'elle soit en transit et au repos.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		Niveau de protection annoncé	4	Niveau de protection annoncé = 4	Niveau 4: permet d'assurer avec confiance extrêmement élevée que la vie privée et la sécurité de l'utilisateur final sont protégées. Ce niveau est utilisé lorsqu'un risque élevé est associé à une violation de la sécurité et de la vie privée de l'utilisateur final. Il s'agit du plus haut niveau de protection de la sécurité et de la vie privée de l'utilisateur final. À la différence du niveau 3, ce niveau nécessite l'utilisation de dispositifs matériels inaltérables pour le stockage de toutes les données sensibles telles que les clés cryptographiques.	
		Niveau de protection annoncé	5	Niveau de protection annoncé = 5	Identique au niveau 4, à ceci près que la disponibilité d'une accréditation ou d'une assurance externe est prouvée.	Cela peut dépendre de l'activité de l'architecture oneM2 M dans le cadre de la certification de dispositifs.
11.0	Age (âge)	Valeur nulle			Lorsqu'il est utile pour définir les paramètres de confidentialité permettant de déterminer l'âge des utilisateurs (par exemple, la fourchette de dates de naissance).	
		JJ/MM/AAAA		Date de naissance	Date de naissance de l'utilisateur final permettant de déterminer l'accès aux ressources.	
		Valeur ou plage de valeurs numériques		Âge minimal requis de l'utilisateur	Âge minimal requis de l'utilisateur permettant de déterminer si ce dernier peut accéder aux ressources.	

Tableau J.1 – Attributs de confidentialité

Identificateur de l'étiquette	Nom de l'étiquette	Valeur	Paramètre	Description condensée de l'étiquette	Description complète de l'étiquette	Notes
		JJ/MM/AAAA		Durée de vie maximale du dispositif	Durée de vie maximale du dispositif permettant de déterminer l'accès aux ressources	Destiné aux dispositifs munis de batteries intégrées présentant une durée de vie limitée ou à des produits chimiques présents dans des appareils médicaux, etc.

Appendice I

Tableau de correspondance de la terminologie de l'architecture GBA 3GPP

(Cet Appendice ne fait pas partie intégrante de la présente Recommandation.)

Le Tableau I.1 compare la terminologie et les abréviations utilisées dans le cadre de l'architecture GBA conformément à la spécification 3GPP [ETSI TS 133 220] avec la terminologie et les abréviations correspondantes propres à l'architecture oneM2M et utilisées dans la présente Recommandation.

Tableau I.1 – Correspondance de la terminologie et des abréviations utilisées dans le cadre l'architecture GBA

Entités, clés et processus de l'architecture GBA	**Entités, clés et processus d'amorçage de la sécurité de l'architecture oneM2M**
Équipement d'utilisateur (UE)	Entité inscrite
Fonction BSF	Fonction MEF
Fonction NAF	Fonction MAF
Procédure d'amorçage	**Prise de contact de sécurité d'amorçage et génération de clé d'inscription temporaire**
Ks	Ke
B-TID	KeID
Procédure d'utilisation d'amorçage	**Utilisation dans le cadre d'une prise de contact MAF**
Nom de domaine complet de la fonction NAF	Identificateur de la fonction MAF
Ks_(ext/int)_NAF	Justificatif d'identité principal (Km)

Appendice II

Mécanisme général d'authentification mutuelle

(Cet Appendice ne fait pas partie intégrante de la présente Recommandation.)

II.0 Introduction

Les systèmes d'authentification mutuelle oneM2M permettent aux entités oneM2M de prouver qu'elles connaissent les justificatifs d'identité connexes, tels que les justificatifs d'identité principaux, sans avoir à échanger ni les valeurs de ces derniers ni aucune donnée sensible comme les identités de sécurité et les identificateurs de sécurité. Pour empêcher la lecture et la copie des justificatifs d'identité, un environnement sécurisé intégré au cadre CSF de sécurité assure une protection contre l'altération de ces justificatifs d'identité et des informations traitées associées.

Un protocole d'authentification mutuelle général est appliqué aux schémas à clé symétrique et à clé asymétrique. Les messages et paramètres précis de protocole dépendent du système et des paramètres de sécurité choisis. Ce protocole comprend généralement les étapes suivantes, représentées sur la Figure II.1:

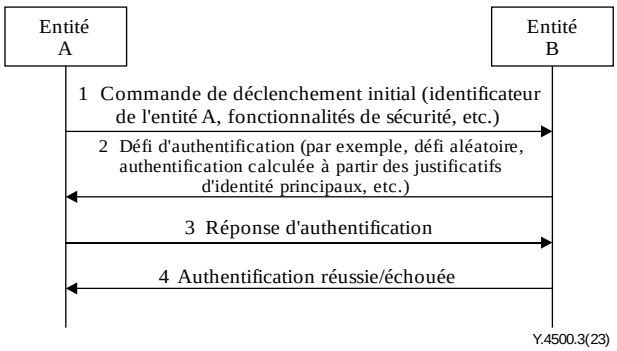


Figure II.1 – Authentification mutuelle

1 Durant l'étape initiale, une entité A est identifiée de façon sécurisée auprès d'une entité B avec laquelle un contact antérieur a été établi ou non. L'entité A s'identifie auprès d'une entité B protégée contre toute écoute clandestine, c'est-à-dire qu'aucune donnée de clé (justificatifs d'identité principaux) n'est échangée.

2 Dans la deuxième étape, l'entité B soumet l'entité A à un défi d'authentification constitué, entre autres, d'une épreuve et du jeton d'authentification (AUTN) de l'entité B calculé à partir du justificatif d'identité principal. Le défi d'authentification, qui peut être aléatoire ou non, dépend du schéma d'authentification et des paramètres de sécurité choisis pour les systèmes fondés sur des clés symétriques ou asymétriques.

3 L'entité A envoie une réponse d'authentification qui contient un jeton d'authentification (AUTN) calculé à partir de ses justificatifs d'identité principaux connus et du défi d'authentification reçu. Cette réponse d'authentification est envoyée si l'entité B a été authentifiée avec succès par l'entité A.

4 L'entité B vérifie ensuite la relation entre l'identité de l'entité A et la réponse reçue à l'étape 3. Si la vérification est concluante, l'entité B est assurée que la réponse a été créée par l'entité A au moyen d'un secret associé à l'identité de l'entité A fournie à l'étape 1.

II.1 Authentification de groupe

Les transactions oneM2M peuvent naturellement faire intervenir des groupes d'entités M2M plutôt que des entités individuelles. Un certain nombre d'entités sont classées dans un même groupe en

raison de leur proximité ou de leurs caractéristiques et de leur propriétaire identiques, ou bien pour d'autres raisons. Pour obtenir des services, toutes les entités d'un tel groupe devraient d'abord être authentifiées. Le mécanisme d'authentification traditionnel comporte deux solutions principales. La première consiste pour le fournisseur de services à authentifier les entités du groupe une à une, tandis que pour la seconde, chaque entité procède à une authentification mutuelle auprès d'un agent de groupe, qui effectue ensuite une authentification mutuelle auprès du fournisseur de services. Si le premier mécanisme d'authentification est utilisé, les délais d'authentification résultant du calcul et de la communication peuvent être trop élevés. Quant au deuxième mécanisme d'authentification, il présente les failles de sécurité suivantes:

1) Il peut être exposé à une attaque par intercepteur menée l'agent de groupe. L'agent de groupe serait placé dans un lieu non sécurisé ou appartenant à un fournisseur différent du fournisseur de service. Si l'agent de groupe est compromis ou ment au fournisseur de services, il peut agir comme un attaquant intermédiaire pour procéder à une fausse authentification auprès des entités et signaler les fausses identités au fournisseur de services étant donné que le fournisseur de services ne s'authentifie pas directement auprès de chaque entité M2M.

2) Préoccupations en matière de confidentialité: toutes les informations provenant des entités M2M sont transférées par l'intermédiaire de l'agent de groupe, qui connaît toutes les informations générées par chaque entité. Sur la base de considérations de sécurité, si l'agent de groupe n'appartient pas au même propriétaire que les entités et les fournisseurs de services, il ne doit pas pouvoir recevoir le message.

Les entités M2M (par exemple, ASN ou ADN) présentant la même fonctionnalité peuvent donc utiliser l'authentification de groupe auprès du fournisseur de services (par exemple, le nœud d'infrastructure) afin de fournir un tunnel sécurisé de bout en bout et de réduire les délais de communication.

Appendice III

Appendice laissé en blanc

Cet appendice est intentionnellement laissé en blanc.

Appendice IV

Appendice laissé en blanc

Cet appendice est intentionnellement laissé en blanc.

Appendice V

Précisions concernant le cadre des cartes UICC en matière de prise en charge des services M2M

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

V.0 Introduction

Cet Appendice fournit d'autres informations pratiques relatives au cadre des cartes UICC de l'architecture oneM2M décrit dans l'Annexe D.

V.1 Contenu suggéré des fichiers EF lors de la prépersonnalisation

Si les fichiers EF n'ont aucune valeur attribuée, il est possible que l'on ne sache pas exactement quelle doit être cette valeur en se basant sur le texte principal. Le présent Appendice indique dans le Tableau V.1 des valeurs pouvant servir dans de tels cas.

Tableau V.1 – Valeurs de fichiers EF prépersonnalisées

Identification de fichier	Description	Valeur
'6F02'	Identificateur d'abonnement à un service oneM2M	'8000FF…FF'
'6F03'	Identificateur du fournisseur de service oneM2M	'8000FF…FF'
'6F04'	Identificateur de nœud M2M	'8000FF…FF'
'6F05'	Identificateur d'entité de services communs (CSE) locale	'8000FF...FF'
'6F06'	Liste des identificateurs d'application M2M	'00FF...FF' pour chaque relevé
'6F07'	Identificateur de fonction MEF	'00FF...FF' pour chaque relevé
'6F08'	Liste des identificateurs d'entité de services communs de nœud d'infrastructure (IN-CSE)	'00FF...FF' pour chaque relevé
'6F09'	Nom de domaine complet de la fonction MAF	'8000FF...FF'
'6F0A'	Tableau des services oneM2M	Dépendant de l'opérateur/du fournisseur de services

V.2 Modifications du fichier EF par le biais du téléchargement de données ou d'applications CAT

Ce paragraphe définit s'il est recommandé de modifier le contenu d'un fichier EF à l'aide du protocole hertzien UICC ou d'une application CAT. La mise à jour par voie hertzienne ou par l'Internet de certains fichiers EF pourrait se traduire par un comportement imprévisible de l'équipement d'utilisateur. Ce genre de modification est mis en évidence par la mention "À effectuer avec précaution" dans le Tableau V.2. Certains fichiers EF sont accompagnés de la mention "Non", signifiant que leur modification par voie hertzienne ou par l'Internet ne devrait en aucun cas être envisagée.

Tableau V.2 – Comportement de mise à jour des fichiers EF

Identification de fichier	Description	Modification recommandée
'6F02'	Identificateur d'abonnement à un service oneM2M	Non
'6F03'	Identificateur du fournisseur de service oneM2M	Non
'6F04'	Identificateur de nœud M2M	À effectuer avec précaution
'6F05'	Identificateur d'entité de services communs (CSE) locale	À effectuer avec précaution
'6F06'	Liste des identificateurs d'application M2M	À effectuer avec précaution
'6F07'	Identificateur de fonction MEF	À effectuer avec précaution
'6F08'	Liste des identificateurs d'entité de services communs de nœud d'infrastructure (IN-CSE)	À effectuer avec précaution
'6F09'	Nom de domaine complet de la fonction MAF	À effectuer avec précaution
'6F0A'	Tableau des services oneM2M	À effectuer avec précaution

V.3 Liste des identificateurs de fichier courts (SFI) au niveau ADF_{M2MS} ou DF_{M2M}

Tableau V.3 – Valeur des identificateurs de fichier courts (SFI)

Identification de fichier	SFI	Description
'6F02'	'02'	Identificateur d'abonnement à un service M2M
'6F03'	'03'	Identificateur du fournisseur de service M2M
'6F04'	'04'	Identificateur de nœud M2M
'6F05'	'05'	Identificateur d'entité de services communs (CSE) locale
'6F06'	'06'	Liste des identificateurs d'application M2M
'6F0A'	'0A'	Tableau des services oneM2M

Toutes les autres valeurs de SFI sont réservées en vue d'une utilisation ultérieure.

V.4 Étiquettes associées aux cartes UICC définies dans l'Annexe J

Tableau V.4 – Étiquettes de cartes UICC

Étiquettes	Nom de l'élément de données	Usage
'80'	Objet de données TLV du nom de domaine complet de la fonction MAF	$EF_{MAFFQDN}$
'80'	Objet de données TLV de l'identificateur M2M-Node-ID	EF_{M2MNID}
'80'	Objet de données TLV de l'identificateur d'entité de services communs locale	EF_{CSEID}
'80'	Objet de données TLV de l'identificateur M2M-SP-ID	$EF_{1M2MSPID}$
'80'	Objet de données TLV d'identificateur d'abonnement M2M	$EF_{1M2MSID}$
NOTE – La valeur 'FF' n'est pas une valeur d'étiquette valide.		

Appendice VI

Demande de décision de contrôle d'accès

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

Une demande de décision de contrôle d'accès telle que présentée au § 6.2.2 est générée par un point d'application de politique (PEP) conformément à la demande d'accès de l'expéditeur et aux informations supplémentaires fournies par l'entité de services communs (CSE) hôte utilisant le format défini par le point de décision de politiques (PDP). Le PEP peut envoyer la demande de décision de contrôle d'accès à un PDP.

Le PDP demande au PRP de récupérer toutes les politiques de contrôle d'accès applicables conformément à la demande de décision de contrôle d'accès, puis utilise ladite demande pour évaluer les politiques de contrôle d'accès récupérées afin de rendre une décision de contrôle d'accès. Une demande de décision de contrôle d'accès envoyée par le PEP au PDP peut contenir les informations suivantes:

– Un expéditeur: identificateur de l'expéditeur qui envoie une demande d'accès à la ressource cible.
– Une ressource: URI de la ressource cible à laquelle l'expéditeur souhaite accéder.
– Une opération: opération que l'expéditeur souhaite effectuer sur la ressource cible.
– Un élément AccessTime: représente l'heure de l'accès.
– Un élément LocationRegion: représente l'emplacement de l'expéditeur.
– Un élément IPAddress: représente l'adresse IP de l'expéditeur.

L'URI de la ressource cible sert à localiser la ressource cible, puis à trouver les politiques de contrôle d'accès associées.

L'identificateur de l'expéditeur est utilisé pour comparer le sujet de composant de règle afin de vérifier si une règle est applicable à la demande de décision de contrôle d'accès.

L'opération sert à comparer les opérations de composant de règle afin de vérifier si l'opération est autorisée par la règle.

Les éléments AccessTime, LocationRegion, ou IPAddress sont utilisés pour vérifier les contextes de composant de règle afin de garantir que certaines conditions supplémentaires sont satisfaites pour utiliser de la règle et prendre une décision de contrôle d'accès.

Appendice VII

Conseils de mise en œuvre et index des solutions

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

L'utilisation de la présente Recommandation implique un processus d'évaluation des risques, propre au contexte, à partir duquel des solutions de sécurité pertinentes sont identifiées.

Le paragraphe 6 donne un aperçu des procédures de sécurité oneM2M. Le paragraphe 6.1.1 présente les interactions entre les couches, le § 6.1.2 présente la séquence des événements et le § 6.2 fournit des informations générales complémentaires, en particulier en matière d'autorisation (§ 6.2.2).

Le paragraphe 7 relatif à l'autorisation et au contrôle d'accès s'applique, quel que soit le type de justificatif d'identité utilisé. Le paragraphe 7.1 décrit le cadre général de gestion de la politique de contrôle d'accès oneM2M, qui peut être encore renforcé par la prise en charge de cadres de contrôle d'accès fondé sur les rôles (§ 7.4) ainsi que l'autorisation dynamique (§ 7.3). En outre, le paragraphe 11 étend les points susmentionnés pour définir une architecture de protection de la confidentialité qui facilite l'établissement et la gestion des profils de confidentialité des utilisateurs.

Le présent Appendice contient le Tableau VII.1, dont l'objectif est d'aider les responsables de la mise en œuvre à identifier les paragraphes de la présente Recommandation qui s'appliquent à un type donné de justificatif d'identité. Les paragraphes pertinents spécifiques à la prise en charge de la sécurité de bout en bout sont indiqués en italiques.

Tableau VII.1 – Index des paragraphes indiquant les procédures utilisées pour chaque type de justificatif d'identité

<table>
<tr><th>Procédure/ solution</th><th>Clé prépartagée (PSK)</th><th>Certificats</th><th>Fonction TEF (GBA, MEF, MAF)</th></tr>
<tr><td rowspan="2">Configuration à distance de la sécurité</td><td>8.3.2.1</td><td>8.3.2.2 et 8.7</td><td>8.3.2.3 (GBA)</td></tr>
<tr><td colspan="2">9.2.1.1</td><td>9.2.1.2, 9.2.2.3, 9.2.2.4 (GBA) et 9.2.3</td></tr>
<tr><td rowspan="3">Création d'association de sécurité</td><td colspan="3">8.2.1, 8.4 (ESPrim), 8.5.2.3 (Signature ESData) et 8.5.2.4 (Signature et chiffrement ESData)</td></tr>
<tr><td colspan="2">9.1.1.1 et 9.1.2.1</td><td>9.1.1.2 et 9.1.2.2 (MAF)</td></tr>
<tr><td>8.1.1, 8.2.2.1 et 8.5.2.2.2 (Chiffrement ESData)</td><td>8.1.2, 8.2.2.2 et 8.5.2.2.4 (Chiffrement ESData)</td><td>8.1.3, 8.2.2.3 (MAF), 8.8 et 8.5.2.2.3 (Chiffrement ESData)</td></tr>
<tr><td rowspan="2">Détails concernant les algorithmes</td><td colspan="3">10.2.1 et 10.3.6</td></tr>
<tr><td>10.2.2 et 10.3</td><td>10.1, 10.2.3 et 10.3</td><td></td></tr>
</table>

Appendice VIII

Appendice laissé en blanc

Cet appendice est intentionnellement laissé en blanc.

Appendice IX

Appendice laissé en blanc

Cet appendice est intentionnellement laissé en blanc.

Appendice X

Règles de mise en œuvre du langage de balisage de conditions

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

Des règles de mise en œuvre types sont indiquées ci-dessous et dans la Figure X.1 et sont répétées pour chaque rangée.

Note sur les conventions: {} identifie les différentes branches d'une condition précédente, [] identifie la trame de filtre et () contient des commentaires. La logique comprend des alinéas pour mieux montrer l'imbrication des instructions. La logique fonctionne en vérifiant les mêmes rangées des trames de filtres en cours de vérification.

La logique suivante est utilisée pour générer la valeur récapitulative pour chaque rangée:

La valeur de la [trame de filtre actuelle] est-elle différente, entre autres, de "no data collected" (aucune donnée collectée) ou "No data shared" (aucune donnée partagée) (= pas de préférence ou de limite)?

{Oui}: la valeur de la [trame de filtre actuelle] est-elle égale à la valeur de synthèse (valeur composée) de la [trame de filtre précédente]?

{Oui}: la valeur de synthèse de la [trame de filtre actuelle] est égale à la valeur définie.

{Non}: la valeur de la [trame de filtre actuelle] est-elle égale à "Oui"?

{Oui}: la valeur de synthèse de la [trame de filtre actuelle] est égale à la valeur de synthèse de la [trame de filtre précédente].

{Non}: la valeur de synthèse de la [trame de filtre actuelle] est égale à la valeur de synthèse de la [trame de filtre actuelle].

{Non}: la valeur de synthèse de la [trame de filtre actuelle] est égale à la valeur de synthèse de la [trame de filtre précédente].

La logique suivante est utilisée pour générer symbole d'acceptabilité des conditions générales:

La valeur de la [trame de filtre actuelle] est-elle égale à "Oui"?

{Oui}: le symbole "☺" est défini comme le symbole d'acceptabilité des conditions générales (une émoticône est utilisée de sorte que le résultat affiché à l'utilisateur n'appartienne à aucune langue et prenne peu de place à l'écran).

{Non}: la valeur de la [trame de filtre actuelle] est-elle égale à la valeur de la [trame de filtre précédente]?

{Oui}: le symbole "☺" est défini comme symbole d'acceptabilité des conditions générales de la [trame de filtre actuelle].

{Non}: le symbole "☹" est défini comme symbole d'acceptabilité des conditions générales de la [trame de filtre actuelle].

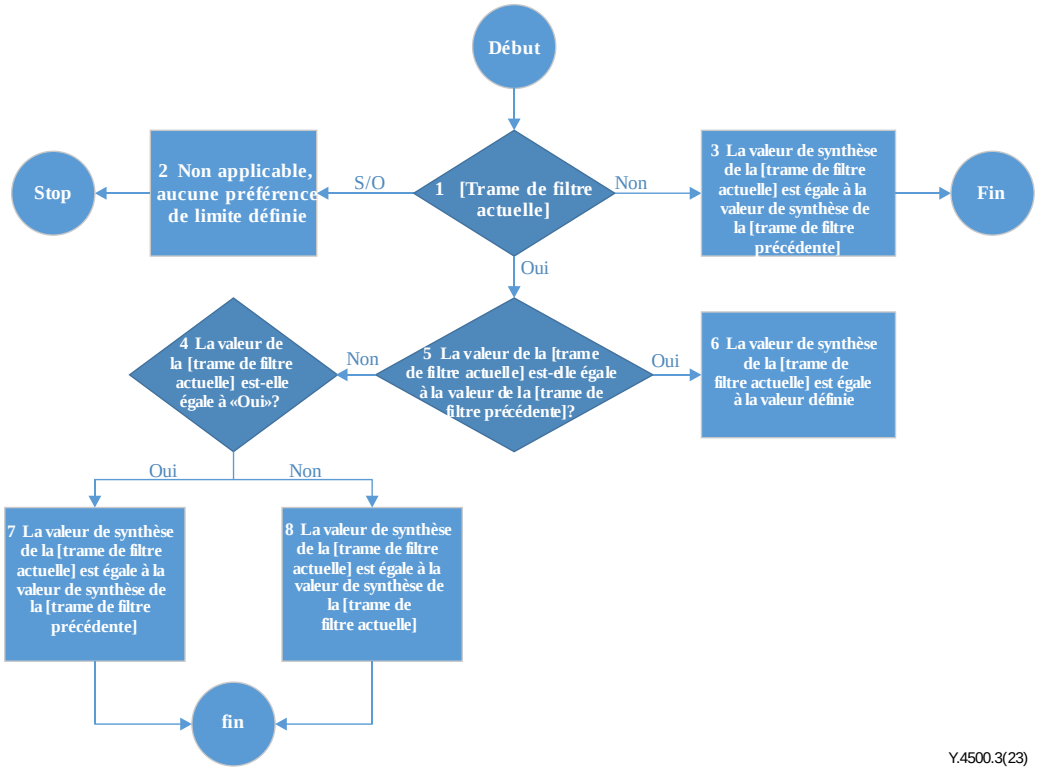


Figure X.1 – Règles de mise en œuvre du langage de balisage

Appendice XI

Exemple de mise en œuvre du protocole SCEP

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

XI.1 Introduction

Le présent Appendice décrit la mise en œuvre du protocole simple d'inscription de certificats (SCEP). Un script exécutant les procédures décrites ici est disponible dans le répertoire Gitlab oneM2M à l'adresse suivante: http://git.onem2m.org/.

XI.2 Procédures de configuration de certificats utilisant le protocole SCEP

La Figure XI.2-1 illustre dans les grandes lignes les procédures du protocole SCEP. La figure indique les modules suivants d'un service d'automatisation de certificats utilisant le protocole SCEP [IETF RFC 8824].

– L'entité de configuration de profils est l'élément principal faisant autorité dans tout système d'automatisation. En établissant des certificats de dispositif préautorisés, l'entité de configuration informe le *client d'automatisation du dispositif* et le service d'infrastructure de clé publique (PKI), qui émet les justificatifs d'identité, qu'un certain nombre de dispositifs uniques demanderont un ou plusieurs certificats de client uniques dédiés.

– La fonctionnalité de configuration transmet à la fois au dispositif distant et au service PKI leurs profils de configuration uniques à travers un canal authentifié et confidentiel. Les profils de configuration peuvent être révisés à tout moment, ce qui permet de forcer toute modification des justificatifs d'identité existants, si nécessaire. Les protocoles de configuration types, entre autres, sont les suivants: TR-069 et OMA-DM.

– Le *client automatisé du dispositif* ou le service intelligent d'application de certificat fournit une machine à états qui utilise les données de configuration, aussi dénommées "profils de configuration", pour générer des clés ainsi que pour demander et remplacer des certificats à des périodes prédéterminées, en formulant des demandes de client SCEP local. Généralement, l'intelligence est pilotée par le temps, garantissant un renouvellement en temps utile des clés et certificats existants, mais elle peut aussi être pilotée par les événements et recevoir des profils de configuration révisés du système de configuration. Le client SCEP est une application native installée sur des systèmes, serveurs ou dispositifs. Il communique avec des répondeurs SCEP au moyen d'un protocole défini dans [IETF RFC 8824]. Les répondeurs SCEP sont identifiés dans les divers profils de configuration.

– La Figure XI.2-1 identifie un certain nombre d'exemples de messages de réponse à une demande de message SCEP. Ces messages sont documentés dans [IETF RFC 8824]. À la réception d'une demande de certificat de chaîne, le répondeur SCEP répond en fournissant le certificat demandé. À la réception d'une demande de certificat de client, le répondeur SCEP valide d'abord l'identité du demandeur et s'assure qu'il détient un justificatif d'identité unique, avant de demander à l'autorité de certification émettrice d'émettre un nouveau certificat, le répondeur SCEP le transmettant ensuite au client SCEP.

– Le répondeur SCEP peut également rejeter la demande de certificat ou indiquer que l'émission est en attente d'une action de l'autorité de certification émettrice. À la réception d'une chaîne de certificats de remplacement, le *client d'automatisation du dispositif* valide la chaîne de certificats reçue, y compris les comparaisons aux réponses de la liste de révocation de certificats (CRL) ou au protocole de statut de certificat en ligne (OCSP). Si et seulement si la nouvelle chaîne de certificats est réputée correcte, la chaîne de certificats est écrite dans la mémoire de certificats de l'application, écrasant ainsi le certificat précédent. Lors du renouvellement, les ancres de confiance de l'homologue peuvent également être renouvelées.

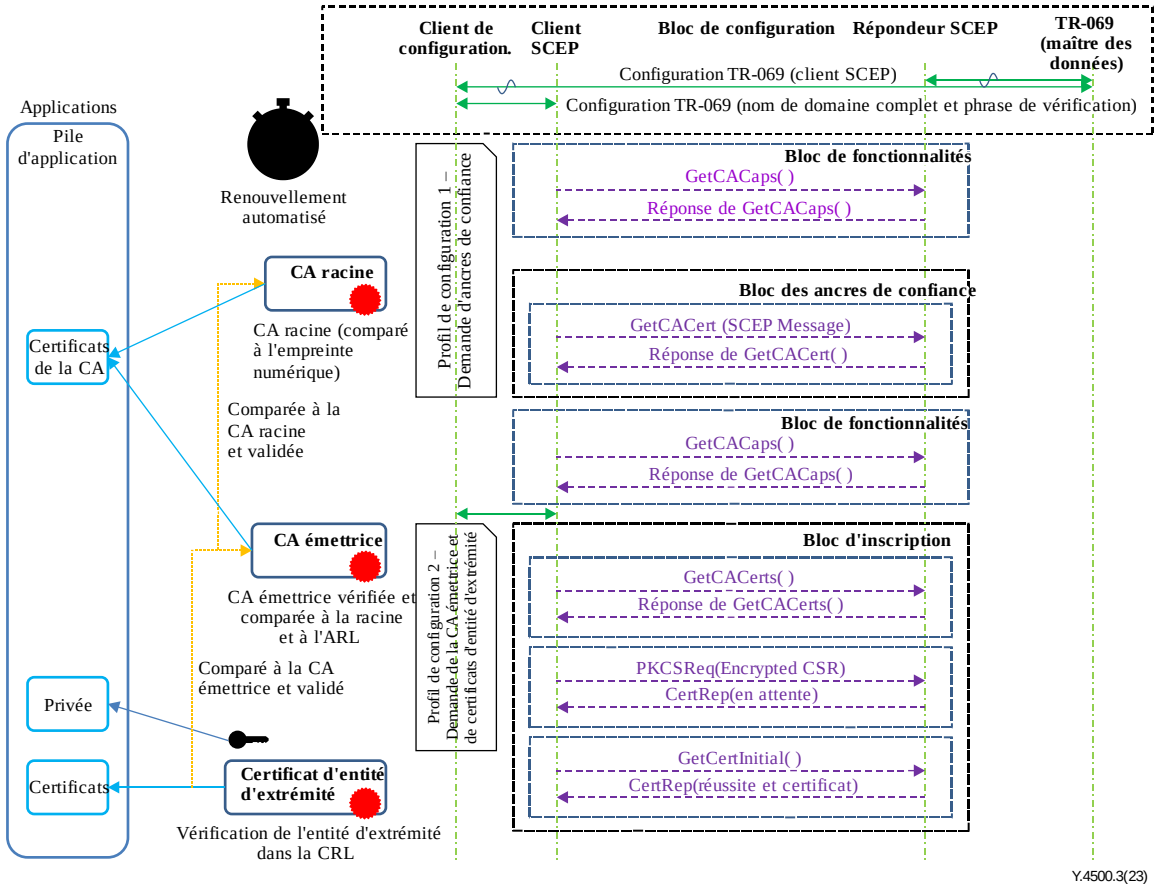


Figure XI.2-1 – Procédure de configuration de certificats SCEP

La solution d'automatisation des certificats SCEP comprend les cinq fonctions suivantes:

1 Configuration initiale du client SCEP à l'aide de profils de configuration

La configuration initiale du client SCEP répond à la nécessité d'établir des ensembles de profils de configuration propres au contexte dans un dispositif de point d'extrémité. Les deux options de fourniture de profils de configuration sont les suivantes:

1) La configuration manuelle de chaque dispositif; et

2) La configuration automatique à partir d'un gestionnaire de dispositifs ou d'un service de gestion d'éléments (par exemple, à l'aide des procédures exposées dans la Recommandation [ITU-T Y.4500.22].

Le nombre d'ensembles de profils de configuration correspond au nombre de piles de sécurité d'application requises.

Cette fonction télécharge un ensemble de profils de configuration à partir du gestionnaire de dispositifs ou du service de gestion d'éléments afin de permettre les actions suivantes:

– Établissement d'un justificatif cryptographique x509v3 unique lié par le biais d'une chaîne à une autorité de certification racine fiable, permettant ainsi au dispositif de point d'extrémité d'amorcer ultérieurement sa configuration [IETF RFC 5280].

– Établissement d'une paire de clés unique significative au niveau local.

– Génération d'une demande de signature de certificat associée.

– Validation hors bande d'une ancre de confiance par la vérification d'une empreinte numérique incluse dans le profil de configuration.

– Validation de chaque autorité de certification (CA) subordonnée récupérée en la comparant à son supérieur.
– Authentification de la demande de certificat de client émise par une entité préautorisée (le répondeur SCEP) et sécurisation de celle-ci au moyen d'un nom d'utilisateur et d'un mot de passe.
– Téléchargement et validation éventuels de l'ancre de confiance d'un homologue fiable. Ces ancres de confiance homologues peuvent être mises à jour sur la base d'un profil de configuration révisé.

2 Intelligence des dispositifs et machines à états

L'intelligence du dispositif et la machine à états sont au cœur de toute solution fondée sur le protocole SCEP, le protocole de gestion de certificats version 2 (CMPv2) ou le protocole EST. Logiquement, une bonne machine à états peut commander n'importe quel répondeur de message lorsque le protocole SCEP est pris en considération ici.

La machine à états est déclenchée par un ensemble complet et valide de profils de configuration.

Quoique conçue pour fonctionner de manière autonome dans le contexte de dispositifs IoT non surveillés sans interface utilisateur de navigateur Web, cette fonction a été écrite pour refléter le parcours d'un utilisateur sur un navigateur. L'objectif est de garantir la compatibilité avec tout processus manuel de test et de diagnostic nécessaire aux dispositifs IoT et avec les éléments du service dotés d'une interface utilisateur classique (par exemple, l'utilisation d'un téléphone intelligent dans le domaine domestique oneM2M).

Voici les étapes principales:

1) Le dispositif demande sa propre ancre de confiance (CA racine).
2) L'ancre de confiance (CA racine) du dispositif est validée en étant comparée à une empreinte numérique fournie par un profil de configuration.
3) L'équipement demande ses propres certificats intermédiaires un par un.
4) Les certificats intermédiaires du dispositif sont validés en étant comparés à l'émetteur supérieur afin d'assurer une protection contre toute attaque par intercepteur.
5) Le dispositif demande un premier certificat de client. Cela suppose qu'un dispositif ne possède pas de certificat de client, mais qu'il détient un ensemble valide de profils de configuration. Durant cette étape, il est toujours demandé à l'autorité de certification émettrice (ICA) d'assurer la confidentialité des demandes de certificat. Le client SCEP récupère la clé publique de l'ICA. La demande de certificat est chiffrée au moyen de la clé publique de façon que seule la clé privée de l'autorité de certification (CA) ou de l'autorité d'enregistrement (RA) puisse déchiffrer la demande.
6) S'il est envoyé par l'autorité de configuration, le dispositif demande un nouveau certificat ou demande immédiatement le renouvellement d'un certificat existant. Une demande de nouveau certificat pourrait concerner une infrastructure PKI différente.
7) Le renouvellement automatique d'un certificat existant, pouvant s'effectuer par exemple en fonction d'un pourcentage de la durée de vie des certificats actuels, est également pris en charge.
8) Tous les certificats sont analysés pour demander des listes CRL associées ou une réponse OCSP.
9) Le client demande l'ancre de confiance des homologues, si elle est différente de la sienne.
10) Il est demandé à la CA intermédiaire et émettrice d'une entité homologue d'autoriser une authentification mutuelle si nécessaire. Une fois qu'une chaîne de certificats nouvelle ou de remplacement a été établie, elle est validée, car elle sera probablement utilisée pour remplacer la bonne chaîne de certificats existante.

11) Les données de clé sont transférées dans les espaces de stockage sécurisés appropriés de l'application.

12) Si nécessaire, l'autorité de configuration du certificat actuel en est informée.

13) Les artefacts de certificats expirés sont supprimés.

Il convient de noter que la liste ci-dessus ne suppose en aucun cas un classement de machines à états dans un ordre donné et n'indique pas de solution. Toutefois, il est supposé que les solutions sophistiquées dépassent le cadre des états identifiés, et que des solutions plus simples peuvent choisir d'omettre les états non requis par la solution du dispositif.

3 Client SCEP

Un client SCEP est généralement un logiciel à code source ouvert conçu pour exécuter des actions de demande de certificat à un répondeur SCEP. Le client SCEP est régi par la machine à états décrite ci-dessus à l'aide des données configurées dans la procédure de configuration initiale.

Le client SCEP est conforme à [IETF RFC 8824] et peut être récupéré auprès de communautés de logiciels à code source ouvert si un client natif n'existe pas à l'heure actuelle. Par exemple, voir [b-SSCEP], qui repose sur les travaux de Martin Bartosch.

Ce client SCEP a été choisi parce que les auteurs ont modifié le comportement de leur client SCEP pour prendre en charge les infrastructures PKI présentant de longues chaînes (voir [b-SSCEP-I]).

Il existe une alternative, à savoir le client SCEP écrit en Java par Dave Grant et son équipe et décrit dans [b-JSCEP].

NOTE – Il a également été modifié pour prendre en charge les infrastructures PKI à longues chaînes et a été récemment scindé par Wes Bunton pour répondre aux exigences propres à Android.

4 Répondeur SCEP

Un répondeur SCEP est un composant supplémentaire des services d'infrastructure PKI tant d'entreprise que gérés. En substance, un répondeur SCEP peut être considéré comme un service supplémentaire d'une RA.

Sur demande, les répondeurs SCEP fournissent des ancres de confiance, des CA intermédiaires, des CA émettrices et des certificats de portée locale. Une paire de clés privée/publique doit être générée sur le dispositif.

La formulation de demandes d'émission de certificats s'opposerait au principe d'un nom d'utilisateur et d'un mot de passe uniques, conservés en toute sécurité dans les champs d'objet alternatif de la demande et de phrase de vérification de la demande de signature du certificat (voir [IETF RFC 8824]).

Généralement, ces mots de passe à usage unique arrivent à expiration lors de la délivrance du certificat et doivent être ensuite réinitialisés lorsque des services de renouvellement de certificat sont nécessaires.

La solution de configuration identifiée fait autorité, c'est-à-dire qu'elle assure le suivi des dispositifs et des éléments gérés, et préfournit au répondeur SCEP des paires valides comprenant un nom d'utilisateur et un mot de passe, avant que le client SCEP ne les utilise.

Les authentifications ayant échoué sont rejetées et les demandes CSR authentifiées avec succès sont transmises à l'infrastructure PKI pour exécution. Une fois l'authentification menée à bien, un certificat d'entité d'extrémité est renvoyé.

La solution de configuration peut même demander la révocation des certificats de dispositif qui ne sont plus fiables.

5 Infrastructure PKI et certificats de portée locale

Un service PKI fournit les connaissances, les compétences et le cadre de conformité préalables nécessaires à la délivrance de certificats SCEP. Les éléments constitutifs d'une solution SCEP sont les suivants: l'infrastructure PKI, la CA, le répondeur SCEP (RA), l'authentificateur de demandes et l'entité d'autorisation de demandes.

En général, dans l'espace des équipements locaux d'abonné (CPE) ou sur l'Internet des objets (IoT), une infrastructure PKI repose sur une bonne compréhension des volumes de certificats et de la séparation opérationnelle cryptographique requise à appliquer.

Autorité de certification

Dans le cadre du protocole SCEP; une autorité de certification (CA) signe des certificats de client. Le nom des autorités de certification est enregistré dans le champ "issuer" des certificats résultants. Avant d'exécuter une opération d'infrastructure PKI, le répondeur SCEP partage un certificat de CA émettrice conforme au profil décrit dans [IETF RFC 5280] avec le client SCEP et partage éventuellement des certificats RA dédiés. Il peut s'agir d'un certificat qui a été émis par une CA de niveau supérieur. Le client construit une chaîne de certificats entière à partir de l'ancre de confiance, en validant chaque certificat tour à tour.

Autorité d'enregistrement

En tant que répondeur SCEP, une autorité d'enregistrement (RA) du protocole SCEP vérifie la validation et l'autorisation du demandeur SCEP et transmet les demandes de certification à la CA. Le répondeur SCEP reçoit un certificat de la CA et le retransmet au client SCEP. Le nom des RA ne figure pas dans le champ "issuer" des certificats résultants.

Authentification du demandeur

Comme pour chaque protocole utilisant la cryptographie à clé publique, l'association entre les clés publiques utilisées dans le protocole et les identités auxquelles elles sont associées est authentifiée de manière sécurisée d'un point de vue cryptographique. Cette exigence est nécessaire pour éviter toute attaque par intercepteur, durant laquelle un attaquant peut manipuler les données au cours de leur transmission entre les participants du protocole et ainsi compromettre la sécurité du protocole. La communication entre le demandeur et l'autorité de certification est sécurisée au moyen d'objets de message sécurisés SCEP qui précisent comment le système PKCS#7 est utilisé pour chiffrer et signer les données de la demande CSR.

Autorisation de la demande

Les méthodes d'authentification SCEP suivantes peuvent être prises en charge pour l'autorisation de certificats:

– Utilisation de noms d'utilisateur et de mots de passe uniques.

– Utilisation d'un certificat d'entité d'extrémité unique et preuve de possession de la clé privée.

Bibliographie

[b-UIT-T X.510] Recommandation UIT-T X.510 (2020), *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: Spécifications de protocole pour un fonctionnement sécurisé*.

[b-UIT-T Y.4500.5] Recommandation UIT-T Y.4500.5 (2018), *oneM2M – Activation de la gestion (OMA)*.

[b-UIT-T Y.4500.6] Recommandation UIT-T Y.4500.6 (2018), *oneM2M – Activation de la gestion (BBF)*.

[b-ATIS.oneM2M.TS0003] oneM2M Security Solutions TS-0003 V2.4.1-2016 ATIS
https://www.onem2m.org/technical/partners-releases

[b-ETSI TS 103 645] ETSI TS 103 645 (2019), Cyber Security For Consumer Internet Of Things.
https://www.etsi.org/deliver/etsi_ts/103600_103699/103645/01.01.01_60/ts_103645v010101p.pdf

[b-ETSI TS 118 103] oneM2M Security solutions TS-0003 v2.4.1 ETSI
https://www.etsi.org/deliver/etsi_ts/118100_118199/118103/02.04.01_60/ts_118103v020401p.pdf

[b-IETF RFC 6455] IETF RFC 6455 (2011), *The Web Socket Protocol, December*.

[b-IETF RFC 7252] IETF RFC 7252 (2014), *The Constrained Application Protocol (CoAP)*.

[b-IETF RFC 7230] IETF RFC 7230 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*.

[b-ISO 3166-1] ISO 3166-1:2020, *Codes pour la représentation des noms de pays et de leurs subdivisions – Partie 1: Codes pays*.

[b-ISO/CEI 7816-5] ISO/CEI 7816-5:2004, *Cartes d'identification – Cartes à circuit(s) intégré(s) – Partie 5: Enregistrement des fournisseurs d'application*.

[b-GP DTSERAM] GlobalPlatform (2015), *Device Technology Secure Element Remote Application Management v1.0 GPD_SPE_008*
https://globalplatform.org/specs-library/secure-element-remote-application-management-v1-0-1/

[b-GP TEEAdmin] GlobalPlatform, *Device Technology TEE Administration framework (PROJET)*.

[b-GP TEESystem] GlobalPlatform (2017), *Device Technology TEE System Architecture, Version 1.0*
https://globalplatform.org/specs-library/tee-system-architecture/

[b-GSMA IoT-SGA] IoT Security Guidelines and Assessment
https://www.gsma.com/iot/iot-security/iot-security-guidelines/

[b-IANA JWT] IANA JSON Web Token (JWT) registry.
http://www.iana.org/assignments/jwt/jwt.xhtml

[b-IoTSF SD-BP] Secure Design: Best Practice Guide. Release 2. Novembre 2019
https://www.iotsecurityfoundation.org/wp-content/uploads/2019/12/Best-Practice-Guides-Release-2_Digitalv3.pdf

[b-JSCEP] A Java implementation of the Simple Certificate Enrolment Protocol.
https://github.com/jscep/jscep

[b-Menezes] *Handbook of Applied Cryptography*, A.J. Menezes, P. C. van Oorschot, S. A. Vanstone, CRC Press, 1996.

[b-NIST 800-162] Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162.
http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

[b-OASIS XACML] *eXtensible Access Control Markup Language (XACML)*, Version 3.0., 22 January 2013. OASIS Standard.

[b-OMA REST] OMA-TS-REST-NetAPI-TerminalLocation-V1-0-20130924-A: *RESTful Network API for Terminal Location*, Version 1.0

[b-oneM2M.XML] *oneM2M XML Schemas*.
http://www.onem2m.org/technical/developers-corner/tools/xml-schemas

[b-oneM2M TR0002] oneM2M TR-0002: *Use Case collection*.

[b-oneM2M TR0008] oneM2M TR-0008: *Security*.

[b-oneM2M TR0012] oneM2M TR-0012: *End to End security*.

[b-oneM2M TR0019] oneM2M TR-0019: *Dynamic annex Authorization*.

[b-SSCEP] A command line client for the SCEP protocol.
https://github.com/certnanny/sscep

[b-SSCEP-I] https://github.com/certnanny/sscep/issues/42

[b-TSDSI STD T1.oneM2M TS-0003-2.4.1 V1.0.0] Solutions de sécurité oneM2M – TS-0003 v2.4.1 TSDSI
https://www.onem2m.org/technical/partners-releases (version 2)

[b-TTAT.MM-TS.0003 v2.4.1] TTAT, Solutions de sécurité oneM2M – TS-0003 v2.4.1 TTAM
https://www.onem2m.org/technical/partners-releases (version 2)
https://www.tta.or.kr/tta/ttaSearchView.do?key=77&rep=1&searchStandardNo=TTAT.MM-TS.0003%20v2.4.1&searchCate=TTAS

[b-TTC-TS-M2M-0003v2.4.1] Solutions de sécurité oneM2M – TS-0003 v2.4.1 TTC Japon
https://www.ttc.or.jp/application/files/8415/5442/6047/TS-M2M-0003v2.4.1.pdf

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série D	Principes de tarification et de comptabilité et questions de politique générale et d'économie relatives aux télécommunications internationales/TIC
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Environnement et TIC, changement climatique, déchets d'équipements électriques et électroniques, efficacité énergétique; construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Gestion des télécommunications y compris le RGT et maintenance des réseaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation et mesures et tests associés
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données, communication entre systèmes ouverts et sécurité
Série Y	**Infrastructure mondiale de l'information, protocole Internet, réseaux de prochaine génération, Internet des objets et villes intelligentes**
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Publié en Suisse
Genève, 2023