Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.1365

(03/2020)

SÉRIE X: RÉSEAUX DE DONNÉES, COMMUNICATION ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

Applications et services sécurisés (2) – Sécurité de l'Internet des objets (IoT)

Méthode de sécurité applicable à l'utilisation de la cryptographie fondée sur l'identité à l'appui des services de l'Internet des objets fournis sur les réseaux de télécommunication

Recommandation UIT-T X.1365

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX DE DONNÉES, COMMUNICATION ENTRE SYSTÈMES OUVERTS ET SÉCURITÉ

RÉSEAUX PUBLICS DE DONNÉES	X.1–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	X.200–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	X.300–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	X.600–X.699
GESTION OSI	X.700–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	X.850–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999
SÉCURITÉ DE L'INFORMATION ET DES RÉSEAUX	
Aspects généraux de la sécurité	X.1000–X.1029
Sécurité des réseaux	X.1030–X.1049
Gestion de la sécurité	X.1050–X.1069
Télébiométrie	X.1080–X.1099
APPLICATIONS ET SERVICES SÉCURISÉS (1)	
Sécurité en multidiffusion	X.1100–X.1109
Sécurité des réseaux domestiques	X.1110–X.1119
Sécurité des télécommunications mobiles	X.1120–X.1139
Sécurité de la toile	X.1140–X.1149
Protocoles de sécurité (1)	X.1150–X.1159
Sécurité d'homologue à homologue	X.1160–X.1169
Sécurité des identificateurs en réseau	X.1170–X.1179
Sécurité de la télévision par réseau IP	X.1180–X.1199
SÉCURITÉ DU CYBERESPACE	
Cybersécurité	X.1200–X.1229
Lutte contre le spam	X.1230–X.1249
Gestion des identités	X.1250–X.1279
APPLICATIONS ET SERVICES SÉCURISÉS (2)	
Communications d'urgence	X.1300–X.1309
Sécurité des réseaux de capteurs ubiquitaires	X.1310–X.1319
Sécurité des réseaux électriques intelligents	X.1330–X.1339
Courrier certifié	X.1340–X.1349
Sécurité de l'Internet des objets (IoT)	**X.1360–X.1369**
Sécurité des systèmes de transport intelligents	X.1370–X.1389
Sécurité de la technologie des registres distribués	X.1400–X.1429
Protocoles de sécurité (2)	X.1450–X.1459
ECHANGE D'INFORMATIONS SUR LA CYBERSÉCURITÉ	
Aperçu général de la cybersécurité	X.1500–X.1519
Échange concernant les vulnérabilités/les états	X.1520–X.1539
Échange concernant les événements/les incidents/l'heuristique	X.1540–X.1549
Échange de politiques	X.1550–X.1559
Heuristique et demande d'informations	X.1560–X.1569
Identification et découverte	X.1570–X.1579
Échange garanti	X.1580–X.1589
SÉCURITÉ DE L'INFORMATIQUE EN NUAGE	
Aperçu de la sécurité de l'informatique en nuage	X.1600–X.1601
Conception de la sécurité de l'informatique en nuage	X.1602–X.1639
Bonnes pratiques et lignes directrices concernant la sécurité de l'informatique en nuage	X.1640–X.1659
Mise en œuvre de la sécurité de l'informatique en nuage	X.1660–X.1679
Sécurité de l'informatique en nuage (autres)	X.1680–X.1699
COMMUNICATIONS QUANTIQUES	X.1700–X.1729

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

Recommandation UIT-T X.1365

Méthode de sécurité applicable à l'utilisation de la cryptographie fondée sur l'identité à l'appui des services de l'Internet des objets fournis sur les réseaux de télécommunication

Résumé

La Recommandation UIT-T X.1365 décrit une méthode de sécurité applicable à l'utilisation de la technologie de clé publique par cryptographie fondée sur l'identité (IBC) à l'appui des services de l'Internet des objets (IoT) fournis sur les réseaux de télécommunication, y compris les mécanismes de gestion de l'identité, l'architecture de gestion des clés, les opérations de gestion des clés et l'authentification.

La méthode classique de sécurité par certificat suppose de recourir à de lourdes opérations en matière de gestion des clés, notamment en ce qui concerne l'émission des certificats, l'interrogation et la révocation. Il est particulièrement difficile pour des systèmes employant ce type de méthode de s'adapter au nombre grandissant de dispositifs connectés à l'IoT, tout en continuant de fonctionner de manière appropriée.

La technologie IBC constitue une autre méthode de sécurité utilisant l'identité d'une entité comme clé publique. L'une des caractéristiques fondamentales de l'IoT est que tous les objets sont dotés d'un identifiant unique. L'utilisation de ces identifiants comme clés publiques présente l'avantage qu'aucun certificat n'est requis. Par conséquent, une solution de sécurité fondée sur la technologie IBC fait appel à une gestion des clés simplifiée, permet aux autorités réparties de contrôler leurs propres dispositifs et s'adapte bien tant à un nombre élevé de points d'extrémité qu'à des dispositifs multiples et variés.

Historique

Edition	Recommandation	Approbation	Commission d'études	ID unique*
1.0	UIT-T X.1365	26-03-2020	17	11.1002/1000/14089

Mots clés

IoT, cryptographie fondée sur l'identité, méthode de sécurité, sécurité des données des utilisateurs.

* Pour accéder à la Recommandation, reporter cet URL http://handle.itu.int/ dans votre navigateur Web, suivi de l'identifiant unique, par exemple http://handle.itu.int/11.1002/1000/11830-en.

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 on 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 à acquérir 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 la base de données des brevets du TSB sous http://www.itu.int/ITU-T/ipr/.

TABLE DES MATIÈRES

Recommandation UIT-T X.1365

Méthode de sécurité applicable à l'utilisation de la cryptographie fondée sur l'identité à l'appui des services de l'Internet des objets fournis sur les réseaux de télécommunication

1 Domaine d'application

La présente Recommandation propose une méthode de sécurité applicable à l'utilisation de la technologie de cryptographie fondée sur l'identité (IBC) à l'appui des services de l'Internet des objets fournis sur les réseaux de télécommunication. Cette méthode de sécurité comprend des mécanismes relatifs aux protocoles d'identification des dispositifs, d'émission de clés privées, de vérification des paramètres publics et d'authentification.

NOTE – Cette méthode ne concerne pas exclusivement le service de l'Internet des objets: d'autres services peuvent également y avoir recours.

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 rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références énumérées ci-dessous. Une liste des Recommandations UIT-T en vigueur est publiée périodiquement. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document en tant que tel le statut de Recommandation.

[IETF RFC 4764] IETF RFC 4764 (2007), *The EAP-PSK protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method*.

[IETF RFC 5091] IETF RFC 5091 (2007), *Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems*.

[IETF RFC 5216] IETF RFC 5216 (2008), *The EAP-TLS Authentication Protocol*.

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

[IETF RFC 5408] IETF RFC 5408 (2009), *Identity-Based Encryption Architecture and Supporting Data Structures*.

[IETF RFC 5480] IETF RFC 5480 (2009), *Elliptic curve cryptography subject public key information*.

[IETF RFC 5958] IETF RFC 5958 (2010), *Asymmetric key packages*.

[IETF RFC 6507] IETF RFC 6507 (2012), *Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)*.

[IETF RFC 6508] IETF RFC 6508 (2012), *Sakai-Kasahara Key Encryption (SAKKE)*.

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

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

[IETF RFC 8446] IETF RFC 8446 (2018), *The Transport Layer Security (TLS) Protocol Version 1.3*.

[ISO/CEI 11770-3] ISO/CEI 11770-3:2015, *Technologies de l'information – Techniques de sécurité – Gestion de clés – Partie 3: mécanismes utilisant des techniques asymétriques*.

[ISO/CEI 14888-3] ISO/CEI 14888-3:2018, *Techniques de sécurité IT – Signatures numériques avec appendice – Partie 3: mécanismes basés sur un logarithme discret*.

[ISO/CEI 18033-5] ISO/CEI 18033-5:2015, *Technologies de l'information – Techniques de sécurité – Algorithmes de chiffrement – Partie 5: chiffrements identitaires*.

3 Définitions

3.1 Termes définis ailleurs

La présente Recommandation utilise les termes suivants définis ailleurs:

3.1.1 fournisseur d'identité [b-UIT-T Y.2720]: entité qui crée, maintient et gère des informations d'identité sécurisées pour d'autres entités (par exemple, utilisateurs/abonnés, organisations et dispositifs) et propose des services fondés sur l'identité basés sur une relation de confiance, commerciale ou d'une autre nature.

3.1.2 identificateur (ID) [b-UIT-T E.101]: série de chiffres, caractères et symboles utilisés pour identifier de manière univoque un abonné, un utilisateur, un élément de réseau, une fonction, une entité de réseau, un service ou une application. Les identificateurs peuvent être utilisés pour l'enregistrement ou l'autorisation. Ils peuvent être publics pour tous les réseaux ou privés pour un réseau particulier (les identificateurs privés ne sont en principe pas divulgués à des tiers).

3.1.3 clé publique principale (MPK) [ISO/CEI 18033-5]: valeur publique unique déterminée par la clé principale secrète correspondante.

3.1.4 clé secrète principale (MSK) [ISO/CEI 18033-5]: valeur secrète utilisée par le générateur de clé privé pour calculer des clés privées pour un algorithme de chiffrement IBE.

3.1.5 générateur de clés privées (PKG) [ISO/CEI 18033-5]: entité ou fonction qui produit un jeu de clés privées.

3.2 Termes définis dans la présente Recommandation

La présente Recommandation définit les termes suivants:

3.2.1 domaine d'identité: ensemble d'entités qui partagent les mêmes paramètres publics et règles de dénomination de l'identité.

3.2.2 paramètre public: un des paramètres qui entrent en jeu dans le calcul cryptographique, notamment le choix d'un dispositif ou d'une fonction de cryptographie spécifique au sein d'une famille de dispositifs ou de fonctions de cryptographie ou d'une famille d'espaces mathématiques, ainsi que la clé publique principale.

3.2.3 serveur de paramètres publics: entité qui attribue des paramètres publics sur demande.

3.2.4 module de sécurité (SecM): dispositif logiciel, matériel ou mixte qui met en œuvre de manière sûre des mécanismes de cryptographie et assure des services de sécurité.

4 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

4G quatrième génération

5G cinquième génération

AGW passerelle commune (*aggregate gateway*)

AK clé d'authentification (*authentication key*)

AKA concordance de clés authentifiées (*authenticated key agreement*)

AN nœud d'accès (*access node*)

AS système d'accès (*access system*)

ASN.1 syntaxe abstraite numéro un (*abstract syntax notation one*)

AU unité d'authentification (*authentication unit*)

AuC centre d'authentification (*authentication centre*)

BLS-12 Barreto-Lynn-Scott de degré de prolongement 12

BLS-24 Barreto-Lynn-Scott de degré de prolongement 24

BN Barreto-Naehrig

CRL liste de révocation de certificats (*certificate revocation list*)

DER règles de codage distinctives (*distinguished encoding rules*)

EAP protocole d'authentification extensible (*extensible authentication protocol*)

ECCSI signatures à courbe elliptique sans certificat pour la cryptographie fondée sur l'identité (*elliptic-curve based certificateless signatures for identity-based encryption*)

EID identificateur d'UICC intégrée (*eUICC-ID*)

EIS jeu d'informations de l'eUICC (*eUICC information set*)

eUICC carte à circuit intégré universelle intégrée (*embedded universal integrated circuit card*)

EUM fabricant d'UICC intégrées (*eUICC manufacturer*)

GW passerelle (*gateway*)

HSM module de sécurité matériel (*hardware security module*)

HTTP protocole de transfert hypertexte (*hypertext transfer protocol*)

IBAKA concordance de clés authentifiées fondée sur l'identité (*identity-based authenticated key agreement*)

IBC cryptographie fondée sur l'identité (*identity-based cryptography*)

IBE chiffrement fondé sur l'identité (*identity-based encryption*)

IBS signature fondée sur l'identité (*identity-based signature*)

ID identificateur

IdP fournisseur d'identité (*identity provider*)

IMSI identité internationale d'abonnement mobile (*international mobile subscription identity*)

IoT Internet des objets (*Internet of things*)

IRL liste de révocation d'identités (*identity revocation list*)

ISD domaine de sécurité de l'émetteur (*issuer security domain*)

ISP plate-forme de services de l'IoT (*IoT service platform*)

KDF fonction de dérivation de la clé (*key derivation function*)

KDK clé de dérivation de la clé (*key derivation key*)

KEK clé de chiffrement de la clé (*key encryption key*)

KEM mécanisme d'encapsulation de la clé (*key encapsulation mechanism*)

KMIP protocole d'interopérabilité de la gestion des clés (*key management interoperability protocol*)

KMS service de gestion de clés (*key management service*)

KPAK clé d'authentification publique du KMS (*KMS public authentication key*)

KSS-16 Kachisa-Schaefer-Scott de degré de prolongement 16

KSS-18 Kachisa-Schaefer-Scott de degré de prolongement 18

LTE évolution à long terme (*long-term evolution*)

LTE-M évolution à long terme, catégorie M1

MAC contrôle d'accès au support (*media access control*)

MNO opérateur de réseau mobile (*mobile network operator*)

MSK clé secrète principale (*master secret key*)

NB-IoT Internet des objets à bande étroite (*narrowband Internet of things*)

OCSP protocole de statut du certificat en ligne (*online certificate status protocol*)

OID identificateur d'objet (*object identifier*)

OISP protocole de statut de l'identité en ligne (*online identity status protocol*)

PKG générateur de clés privées (*private key generator*)

PKI infrastructure de clé publique (*public key infrastructure*)

PPS serveur de paramètres publics (*public parameter server*)

PVT jeton de vérification public (*public verification token*)

RSF fonction de serveur de révocation (*revocation server function*)

SecM module de sécurité (*security module*)

SK Sakai-Kasahara

SM-DP préparation de données du gestionnaire des abonnements (*subscription manager data preparation*)

SM-SR acheminement sécurisé du gestionnaire des abonnements (*subscription manager secure routing*)

SOK Sakai-Ohgishi-Kasahara

SSK clé de signature secrète (*secret signing key*)

TLS sécurité dans la couche transport (*transport layer security*)

TLV étiquette, longueur et vecteur (*tag, length and vector*)

TVP paramètre variable dans le temps (*time-variant parameter*)

UE équipement d'utilisateur (*user equipment*)

UICC carte à circuit intégré universelle (*universal integrated circuit card*)

5 Conventions

Aucune.

6 Présentation

L'Internet des objets (IoT) peut être considéré comme une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication (TIC) interopérables existantes ou en évolution, d'après le § 6.1 de la Recommandation [b-UIT-T Y.4000]. La sécurité de l'IoT est un enjeu majeur, en raison de l'omniprésence des appareils concernés et du caractère de plus en plus sensible des données des utilisateurs. La Recommandation [b-UIT-T Y.4100] décrit les exigences de haut niveau communes en matière de sécurité de l'IoT, notamment pour la sécurité des communications, la sécurité de la gestion des données, la sécurité de la prestation de services, ainsi que l'authentification mutuelle et l'autorisation. La Recommandation [b-UIT-T X.1361] analyse de manière plus détaillée les menaces et les défis liés à la sécurité dans un environnement lié à l'IoT, et présente différentes méthodes pour les traiter et les atténuer. Les capacités requises en matière de sécurité définies par la Recommandation [b-UIT-T X.1361] comprennent:

– une capacité de sécurité des communications pour prendre en charge les communications de manière sécurisée et sûre en protégeant la confidentialité;
– une capacité de sécurité de la gestion des clés pour prendre en charge les communications de manière sécurisée;
– une capacité de sécurité de la gestion des données pour assurer la gestion des données de manière sécurisée et sûre en protégeant la confidentialité;
– une capacité d'authentification pour authentifier les dispositifs;
– une capacité d'autorisation (c'est-à-dire de contrôle de l'accès) pour accorder des autorisations aux dispositifs;
– une capacité de mise en œuvre de protocoles sécurisés fondés sur des algorithmes cryptographiques légers.

Les dispositifs connectés à l'IoT sont caractérisés par leurs ressources limitées, notamment en matière de capacités de calcul et de communication. La nature de ces dispositifs est à l'origine de difficultés inédites pour satisfaire aux exigences de sécurité des systèmes connectés à l'IoT. Un déploiement particulièrement aisé, des opérations peu importantes de gestion et une répartition de l'autorité font partie des principales caractéristiques requises de toute solution de sécurité envisagée pour l'IoT.

Comme le souligne la Recommandation [b-UIT-T X.1361], l'authentification, le contrôle de l'accès, l'intégrité des données et la confidentialité comptent parmi les services indispensables à la sécurité de l'IoT. Pour proposer ces services, il est possible d'employer des mécanismes de cryptographie à clés symétriques ou à clés publiques.

Les solutions de sécurité fondées sur l'utilisation de clés symétriques sont assez simples à mettre en œuvre. Cependant, elles ne sont pas très adaptées aux scénarios faisant intervenir des entités homologues, par exemple les applications machine-à-machine de l'IoT, sans un service en ligne agissant comme intermédiaire pour établir la confiance ou sans partager au préalable un secret entre les deux dispositifs concernés. Il est également difficile d'assurer des communications sécurisées entre systèmes sans risque de divulgation des secrets de l'utilisateur à des entités homologues.

Les solutions traditionnelles de cryptographie à clés publiques, fondées sur l'utilisation de certificats, nécessitent d'importantes opérations de gestion de clés, notamment l'émission, la demande, la

distribution, la vérification et la révocation des certificats. Les systèmes de ce type rencontrent des difficultés significatives pour faire face à l'augmentation constante du nombre de dispositifs et de fonctionnalités de l'IoT tout en conservant un niveau de performance acceptable. La surcharge liée à l'échange de certificats dans le cadre des protocoles de sécurité est également source de problèmes, en particulier dans les réseaux d'Internet des objets à bande étroite (NB-IoT), dont l'unité de données en mode paquet est petite.

La cryptographie fondée sur l'identité (IBC) est un autre type de technologie, qui utilise l'identité d'une entité comme clé publique. L'une des caractéristiques fondamentales de l'IoT est que chaque élément est pourvu d'un identificateur unique (ID). Si ces identificateurs sont utilisés comme clés publiques, aucun certificat n'est nécessaire. Par conséquent, une solution de sécurité IBC simplifie la gestion de clés, permet à différentes autorités de contrôler leurs propres dispositifs et peut être employée en même temps pour un grand nombre de terminaux et des dispositifs très variés. Étant donné qu'aucun certificat n'est transmis, les protocoles de sécurité peuvent être menés avec une plus grande efficacité.

Au sein d'un système IBC, une partie de confiance, appelée service de gestion de clés (KMS), est chargée de créer la clé privée de chaque entité. Avant de fournir un service de création de clé, le KMS déclenche un processus d'initialisation du système en lançant la fonction **IBSetup** (paramétrage fondé sur l'identité) qui, à partir d'un paramètre de sécurité donné, détermine un ensemble de paramètres pour le système et produit une clé principale secrète (MSK) et une clé principale publique (MPK). Soulignons que le KMS remplit la même fonction qu'un générateur de clés privées (PKG). Ainsi, par souci de simplicité, les termes "KMS" et "PKG" sont utilisés indifféremment dans la présente Recommandation et l'expression "paramètres publics" désigne l'ensemble constitué par les paramètres du système et la MPK. Le KMS assure la confidentialité de la MSK et le libre accès aux paramètres publics. Si nécessaire, les paramètres publics peuvent être publiés par un serveur de paramètres publics pour les services dédié à cet usage.

Un système de sécurité fondé sur la technologie IBC classique dispose d'une gamme de mécanismes d'IBC, notamment le chiffrement fondé sur l'identité (IBE), la signature fondée sur l'identité (IBS) et la concordance de clés authentifiées fondée sur l'identité (IBAKA) pour assurer différents services liés à la sécurité, tels que la protection de la confidentialité des données, l'authentification des entités et la mise en place de canaux sécurisés. Tous ces algorithmes d'IBC peuvent être considérés comme la combinaison de deux jeux de fonctions. Le premier regroupe les fonctions de création de clés, qui produisent des paires de clés publiques et privées fondées sur l'identité. La fonction de création de clés privées (**IBExtract**) produit une clé privée à partir d'un identificateur, de la MSK et des paramètres publics. La fonction de dérivation de la clé publique fondée sur l'identité (**IBDerivate**) calcule une clé publique à partir d'un identificateur et des paramètres publics. L'autre jeu de fonctions, qui comprend notamment le chiffrement et le déchiffrement (**IBEnc/IBDec**), la signature et la vérification (**IBSign/IBVerify**) ainsi que le protocole d'établissement de clé de session authentifiée, utilise les paires de clés ainsi créées pour mener à bien les opérations de cryptographie correspondantes.

La technologie d'IBC fait l'objet de normes établies par différentes organisations de normalisation, notamment l'Organisation internationale de normalisation (ISO), la Commission électrotechnique internationale (CEI), le Groupe d'étude sur l'ingénierie Internet (IETF), l'Institut d'ingénierie électrique et électronique (IEEE), l'Institut européen des normes de télécommunications (ETSI) et l'Administration des normes de Chine (SAC). Une liste de normes pertinentes élaborées par ces organismes figure dans la bibliographie. L'initiative OneM2M envisage également l'utilisation des technologies d'IBC dans les réseaux d'IoT dans sa quatrième publication, dont l'analyse relative à la sécurité est exposée dans [b-ETSI TR 118 508].

La présente Recommandation décrit un cadre de sécurité pour l'utilisation de la technologie IBC en vue d'assurer les fonctions de sécurité des services liés à l'IoT passant par les réseaux de télécommunication. Ce cadre porte notamment sur la gestion de l'identité, l'architecture de la gestion

des clés, les opérations de gestion des clés et l'authentification, ainsi que sur les protocoles de concordance de clés pour l'utilisation de la technologie IBC.

7 Architecture système de référence pour les services liés à l'IoT passant par les réseaux de télécommunication

Ce paragraphe présente une architecture système générique de référence pour les services liés à l'IoT passant par les réseaux de télécommunication. La Figure 1 montre une architecture système conceptuelle de référence pour les services liés à l'IoT. Le système est composé de trois domaines: le dispositif connecté à l'IoT, le système d'accès (AS) et la plate-forme de services de l'IoT (ISP).

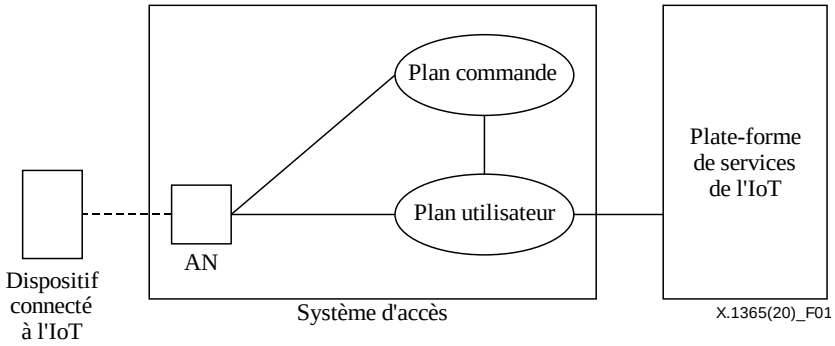

Figure 1 – Architecture système conceptuelle pour les services liés à l'Internet des objets

Les dispositifs connectés à l'IoT sont chargés de collecter des données ou de réaliser des actions. La plupart d'entre eux peuvent se connecter à un système de télécommunication et communiquer avec une ISP. De nos jours, la majorité des dispositifs se connectent à une ISP par le biais d'une liaison sans fil avec un réseau de télécommunication. Dans la présente Recommandation, le terme "système d'accès" (AS) renvoie aux réseaux de télécommunications. Un système d'accès est généralement constitué de deux parties: le réseau d'accès (AN) et le réseau central. Ce dernier peut être à son tour composé de deux parties: le plan commande, responsable du signalement des commandes, et le plan utilisateur, chargé de la transmission des données.

Cela fait déjà plusieurs générations que nous avons recours aux réseaux de télécommunication, sous la forme d'une connexion sans fil traditionnelle. À l'origine, les réseaux de télécommunication sont conçus de sorte à permettre les communications mobiles entre êtres humains sans discontinuité lors de l'itinérance. Au cours des dernières années, la prise en charge des dispositifs connectés à l'IoT a été intégrée à leurs fonctions, à partir de la quatrième génération (4G) de réseaux à évolution à long terme (LTE). Par exemple, dans la quatrième génération de l'évolution à long terme (4G-LTE), la catégorie M1 (LTE-M) et les technologies NB-IoT sont conçues pour prendre en charge les dispositifs connectés à l'IoT.

La majorité des systèmes de télécommunications actuels sont constitués de trois composantes: des terminaux, ou équipements d'utilisateurs (UE), un AN et des réseaux centraux. Dans le cas présent, on suppose que l'AN et les réseaux centraux font partie de l'AS, comme le montre la Figure 1. Les services liés à l'IoT sont généralement extérieurs aux réseaux de télécommunications, auxquels ils sont reliés par des interfaces pour la transmission de données et la gestion des services. Afin de mieux prendre en charge les services liés à l'IoT, les réseaux de télécommunication comportent de plus en plus de caractéristiques conçues pour l'IoT dans leurs spécifications. L'intégration entre les réseaux de télécommunication et les services liés à l'IoT s'est accrue au cours des dernières années.

Les spécifications des systèmes conçus pour les réseaux de la cinquième génération (5G) assurent la prise en charge des technologies de clés publiques pour les services liés à l'IoT, notamment l'authentification en vue de l'accès au réseau. Comme indiqué dans le paragraphe 6, la technologie IBC est plus simple à gérer et plus efficace en matière de transmission que les autres technologies de

clés publiques. Par conséquent, l'utilisation de la technologie IBC dans le cadre des services liés à l'IoT passant par les réseaux de télécommunication doit faire l'objet d'une spécification en tant que norme complémentaire des spécifications existantes.

8 Cadre pour l'utilisation de la cryptographie fondée sur l'identité pour les services liés à l'IoT passant par les réseaux de télécommunication

Ce paragraphe propose un cadre pour l'utilisation des technologies de clés publiques à IBC pour les services liés à l'IoT passant par les réseaux de télécommunication. Ce cadre comporte une architecture système formée des composants du réseau nécessaires à l'utilisation de la technologie IBC. Il spécifie également un cadre de gestion de clés pour la technologie IBC, indispensable à tout système ayant recours à cette technologie. Enfin, il aborde d'autres points essentiels, tels que les protocoles de gestion des clés, de dénomination de l'identité et d'authentification.

8.1 Architecture système pour l'IoT, utilisant la cryptographie fondée sur l'identité

Dans le cadre des services liés à l'IoT qui passent par les réseaux de télécommunication, la technologie IBC peut servir à l'authentification en vue de l'accès au réseau, aux services, ou aux deux. L'authentification en vue de l'accès au réseau vise à déterminer si un dispositif a l'autorisation d'accéder au réseau, tandis que l'authentification en vue de l'accès aux services vérifie si un dispositif peut se connecter à une ISP.

Les dispositifs connectés à l'IoT peuvent accéder au réseau de télécommunication directement ou indirectement. Il existe donc deux modèles d'accès:

- modèle de connexion directe: les dispositifs connectés à l'IoT se connectent directement à l'AS;
- modèle de connexion indirecte: les dispositifs connectés à l'IoT se connectent à l'AS par le biais d'une passerelle commune (AGW).

La Figure 2 présente une architecture système de référence pour l'IoT, dans laquelle la technologie IBC est employée pour assurer la sécurité de l'AS et de l'ISP. En matière de sécurité, l'AS et l'ISP peuvent avoir leurs propres exigences relatives aux services liés à l'IoT. Les justificatifs de sécurité peuvent être fournis par l'AS ou par l'ISP. Les trois scénarios suivants sont donc envisageables pour l'utilisation de l'IBC dans les réseaux liés à l'IoT.

- Scénario d'utilisation de la technologie IBC pour la protection de la sécurité de l'AS

  Dans ce scénario, c'est l'AS qui fournit et qui gère les justificatifs de sécurité pour l'accès au réseau stockés dans les dispositifs connectés à l'IoT. Ceux-ci sont authentifiés par l'AS lorsqu'ils se connectent à lui. Par exemple, un dispositif connecté à l'IoT calcule sa signature IBS à partir de la clé privée communiquée par l'AS, et lui envoie cette signature. L'AS peut alors authentifier le dispositif à partir de la signature IBS figurant dans les messages d'authentification. Si la vérification est un succès, l'AS envoie les données du dispositif au serveur de l'IoT.

- Scénario d'utilisation de la technologie IBC pour la protection de la sécurité de l'ISP

  Les justificatifs à IBC stockés dans les dispositifs connectés à l'IoT sont fournis et gérés par l'ISP et servent à l'accès aux services. L'ISP authentifie les dispositifs grâce à leur signature produite à l'aide de ces justificatifs à IBC.

- Scénario d'utilisation de la technologie IBC pour la protection de la sécurité de l'AS et de l'ISP

  Les justificatifs de sécurité à IBC stockés dans les dispositifs connectés à l'IoT sont fournis et gérés par l'AS, par l'ISP, ou de manière conjointe par l'AS et l'ISP. L'AS et l'ISP peuvent authentifier un dispositif connecté à l'IoT à l'aide du même jeu de justificatifs.

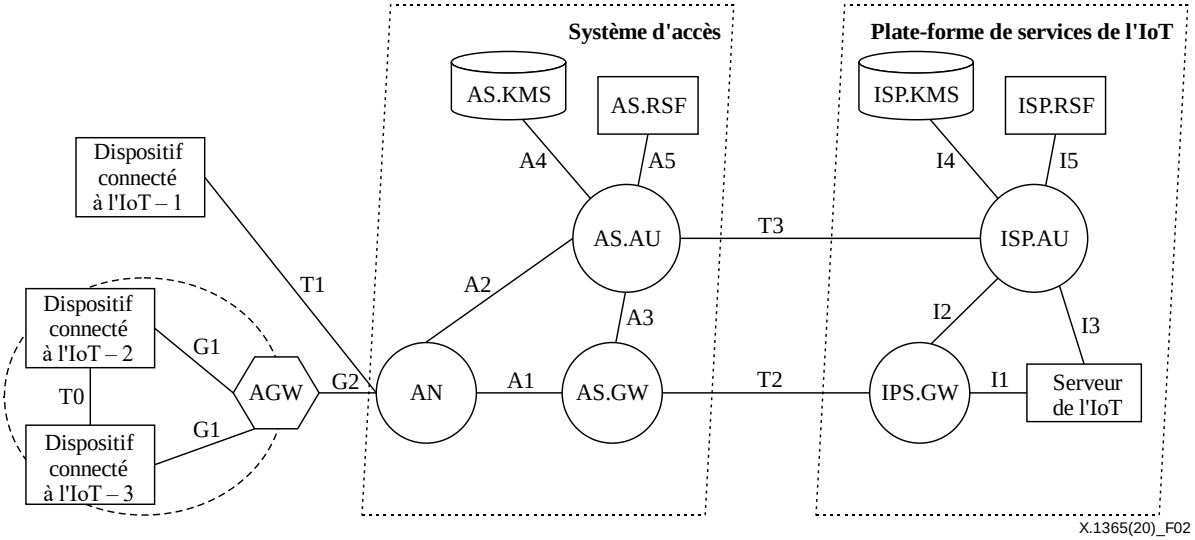

Figure 2 – Architecture système pour l'IoT fondée sur les scénarios de protection de la sécurité du système d'accès et de la plate-forme de services de l'IoT à l'aide de la cryptographie fondée sur l'identité

Les trois scénarios décrits ci-dessus couvrent la plupart des cas d'utilisation de la technologie IBC pour l'accès au réseau et aux services. Cependant, d'autres scénarios, qui n'entrent pas dans le domaine d'application de la présente Recommandation, pourraient être envisagés.

L'architecture système pour l'IoT fondée sur la technologie IBC comprend les fonctions de réseau (NF) et les dispositifs suivants:

- Système d'accès (AS): système d'accès pour les dispositifs connectés à l'IoT ou l'AGW, qui comprend un nœud d'accès (AS.AN), une fonction de système de gestion des clés (AS.KMS), une unité d'authentification (AS.AU), une fonction de serveur de révocation (AS.RSF) et une passerelle (AS.GW).
- Plate-forme de services de l'IoT: plate-forme pour la gestion des services liés à l'IoT, qui comporte une fonction de système de gestion des clés (ISP.KMS), une unité d'authentification (ISP.AU), une fonction de serveur de révocation (ISP.RSF), une passerelle (ISP.GW) et un serveur de l'IoT. Elle prend notamment en charge la gestion et la distribution des clés, l'authentification de l'identité, le chiffrement et le déchiffrement, l'utilisation et la vérification des signatures.
- Passerelle commune (AGW): nœud de regroupement chargé de la connexion des dispositifs connectés à l'IoT, qui regroupe et envoie l'ensemble des données de ces dispositifs à l'AS. Elle joue le rôle de mandataire pour la transmission des données entre eux et l'AN.
- Nœud d'accès (AN): nœud d'accès pour les dispositifs connectés à l'IoT ou pour l'AGW; il peut s'agir d'un point d'accès au réseau fixe ou hertzien.
- Fonction de système de gestion des clés (KMS): système de gestion chargé de la création, de la distribution et de la mise à jour des clés à IBC ainsi que des paramètres des dispositifs connectés à l'IoT et des fonctions du réseau.
- Unité d'authentification (AU): elle authentifie les dispositifs connectés à l'IoT à l'aide d'un système fondé sur la technologie IBC.
- Fonction de serveur de révocation (RSF): chaque serveur tient une liste de révocation d'identités (IRL), qui répertorie les clés publiques et identités dont l'utilisation est exclue.

  NOTE – L'AS et l'ISP peuvent disposer chacun de leurs propres KMS, AU et RSF.
- Passerelle d'accès au système (AS.GW): élément de réseau connecté à une passerelle de l'IoT, chargé de la transmission des données des utilisateurs de l'IoT.

– Passerelle de l'IoT (IoT GW): passerelle chargée de transférer ou de regrouper des données, de transmettre des données à un serveur de l'IoT, ou de transférer des données et des informations de signalisation d'un serveur de l'IoT à des dispositifs connectés à l'IoT.
– Serveur de l'IoT: serveur dépendant du fournisseur de services de l'IoT, qui collecte les données relatives à l'IoT transmises par la passerelle de l'IoT.
– Dispositif connecté à l'IoT: dispositif final utilisé pour la collecte de données et l'établissement de connexions avec un AN et un serveur de l'IoT. Il prend en charge la protection des données, notamment la négociation, le chiffrement et le déchiffrement des clés, ainsi que l'usage et la vérification des signatures.

Les fonctions des points de référence indiqués sur la Figure 2 sont les suivantes:

– G1: point de référence entre un dispositif connecté à l'IoT et l'AGW, utilisé pour l'authentification et les communications relatives à la sécurité.
– G2: point de référence entre une AGW et un AN, qui sert pour la signalisation et la communication de données entre ces deux éléments.
– T0: point de référence entre les dispositifs connectés à l'IoT, employé pour la signalisation et l'échange de données.
– T1: point de référence entre les dispositifs connectés à l'IoT et un AN, utilisé pour l'authentification et les communications relatives à la sécurité.
– T2: points de référence entre l'AS.GW et l'ISP.GW, qui constituent un tunnel de données sur le plan utilisateur entre l'AS.GW et l'ISP.
– T3: points de référence entre l'AS.AU et l'ISP.AU, qui servent pour les échanges relatifs à la signalisation, notamment les échanges d'identités et la fourniture de clés.
– A1: points de référence entre l'AN et l'AS.GW, employés dans le cadre du tunnel de données sur le plan utilisateur.
– A2: points de référence entre l'AS.AU et l'AN, utilisés pour la signalisation sur le plan commande.
– A3: points de référence entre l'AS.AU et l'AS.GW, employés dans le cadre du protocole d'allocation et de gestion de passerelle dans l'AS.
– A4: points de référence entre l'AS.AU et l'AS.KMS, qui servent au protocole de fourniture de clé dans l'AS.
– A5: points de référence entre l'AS.AS.AU et l'AS.RSF, utilisés pour le protocole de révocation de clé ou d'identité dans l'AS.
– I1: points de référence entre le serveur de l'IoT et l'ISP.GW, employés dans le cadre du tunnel de données sur le plan utilisateur.
– I2: points de référence entre l'ISP.AU et l'ISP.GW, qui servent pour le protocole d'allocation et de gestion de passerelle dans l'ISP.
– I3: points de référence entre l'ISP.AU et le serveur de l'IoT, utilisés pour l'échange d'informations, par exemple d'informations d'abonnement liées au service transférées du serveur à l'ISP.AU, ou de messages de notification d'authentification envoyés par l'ISP.AU au serveur.
– I4: points de référence entre l'ISP.AU et l'ISP.KMS, qui servent au protocole de fourniture de clé dans l'ISP.
– I5: points de référence entre l'ISP.AU et l'ISP.RSF, employés dans le cadre du protocole de révocation de clé ou d'identité dans l'AS.

8.2 Architecture de gestion des clés

Ce paragraphe décrit l'architecture fonctionnelle nécessaire pour la prise en charge de la gestion des clés en cas d'utilisation de mécanismes de la technologie IBC dans l'IoT. En fonction de la présence ou de l'absence de carte à circuit intégré universelle intégrée (eUICC) [b-GSMA SGP.02] dans un dispositif connecté à l'IoT, deux types d'architecture de gestion des clés sont envisageables: 1) l'utilisation de la technologie IBC dans les dispositifs connectés à l'IoT munis d'une eUICC; 2) l'utilisation de la technologie IBC dans les dispositifs connectés à l'IoT non munis d'une eUICC.

En cas d'utilisation de l'IBC dans des dispositifs connectés à l'IoT munis d'une eUICC, l'architecture reprend celle généralement employée pour la prestation de services à distance par le biais d'une eUICC, définie dans [b-GSMA SGP.02], en y ajoutant deux entités fonctionnelles supplémentaires: le KMS et le PPS. En fonction de l'emplacement du KMS, ce scénario comporte les deux modalités possibles suivantes:

1) Le KMS est géré par l'entité responsable d'un opérateur de réseau mobile (MNO): voir la Figure 3.
2) Le KMS est géré par l'entité responsable de la préparation des données du gestionnaire des abonnements (SM-DP): voir la Figure 4.

Dans les deux cas, les clés, y compris la clé privée et les paramètres publics, sont créées lorsqu'un MNO effectue une demande de profil. Elles sont alors transmises à distance aux dispositifs munis d'une eUICC, de la même manière que les clés installées conformément à la spécification actuelle en matière de fourniture de clés à distance [b-GSMA SGP.02]. Cette spécification détaille les rôles, les fonctions associées et les interfaces de la prestation de services à distance par le biais d'une eUICC. La spécification du profil, du format de stockage et de l'utilisation de ces clés dans les eUICC ne fait pas partie du champ de la présente Recommandation.

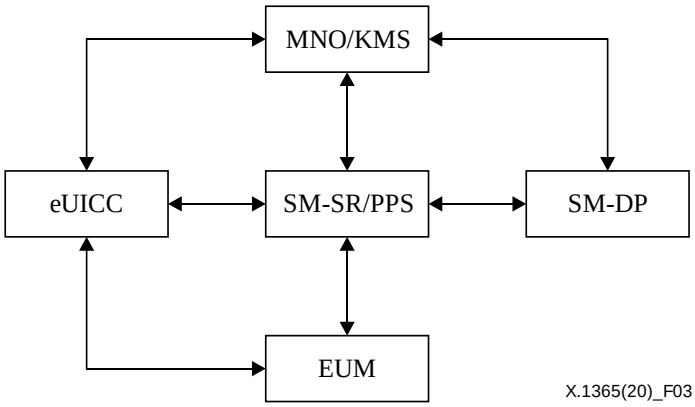

Figure 3 – Architecture A de gestion des clés utilisant la cryptographie fondée sur l'identité pour les dispositifs connectés à l'IoT munis d'une carte à circuit intégré universelle intégrée

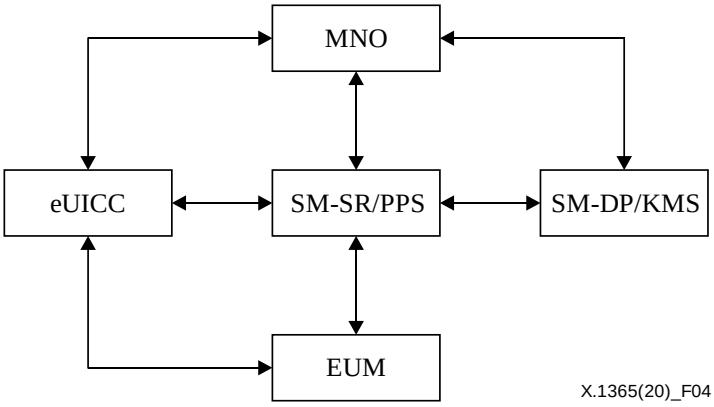

Figure 4 – Architecture B de gestion des clés utilisant la cryptographie fondée sur l'identité pour les dispositifs connectés à l'IoT munis d'une carte à circuit intégré universelle intégrée

La Figure 5 présente une architecture générique en cas d'utilisation de la technologie IBC dans des dispositifs connectés à l'IoT non munis d'une eUICC. Ses composantes de base sont les suivantes:

– SecM: un module de sécurité (SecM) est un élément capable de stocker de manière sécurisée les clés et de les employer pour exécuter des mécanismes de sécurité afin de mener à bien des opérations de sécurité. Un dispositif connecté à l'IoT doit être doté d'un SecM.

– IdP: un fournisseur d'identité (IdP) est une entité qui crée, entretient et gère les informations relatives à l'identité.

– AuC: un centre d'authentification (AuC) offre un service d'authentification des entités.

L'IdP dépend du service d'authentification fourni par l'AuC pour authentifier les dispositifs connectés à l'IoT. Après la procédure d'authentification initiale, l'IdP assure un service de fourniture d'identité au SecM, qui couvre notamment la création, la désignation, le remplacement et la révocation des identités. Une fois qu'une nouvelle identité est créée et attribuée à un dispositif connecté à l'IoT, l'IdP utilise le service de création de clés privées fourni par le KMS pour produire la clé privée correspondant à cette identité et pour transmettre les clés au SecM de manière sécurisée. L'IdP obtient également du KMS des paramètres publics et les envoie au PPS, qui publie les paramètres publics pour les entités extérieures. L'IdP peut également fournir un service d'authentification à d'autres entités en exécutant conjointement avec le SecM des protocoles d'authentification spécifiques, notamment ceux définis dans la présente Recommandation.

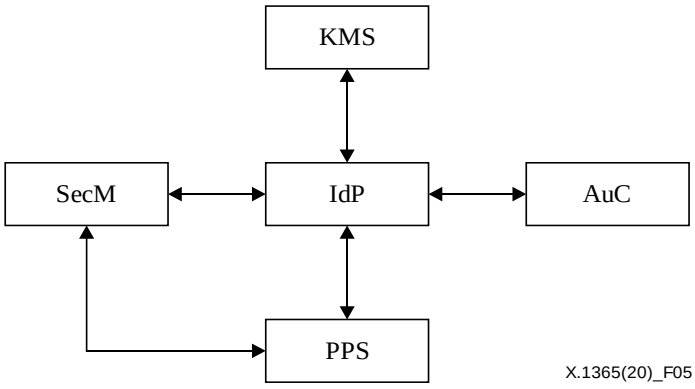

Figure 5 – Architecture de gestion des clés utilisant la cryptographie fondée sur l'identité pour les dispositifs connectés à l'IoT non munis d'une carte à circuit intégré universelle intégrée

8.3 Dénomination de l'identité

Lorsque la technologie IBC est employée dans le cadre des services liés à l'IoT passant par un réseau de télécommunication, la dénomination de l'identité peut fournir des informations utiles pour aider les opérateurs à gérer le réseau. Différentes informations peuvent être intégrées dans une identité, par exemple le type de service, l'emplacement, l'identificateur du dispositif et l'heure valable. Certaines de ces informations, par exemple l'heure valable, sont indispensables pour utiliser la technologie IBC. Les informations relatives à l'identité permettent à l'opérateur d'optimiser la gestion du réseau, par exemple en attribuant une connexion à des tranches de réseau spécifiques en fonction du type de service qu'elle propose. Les informations relatives à l'emplacement simplifient également la localisation des dispositifs. Un exemple de définition de l'identité figure dans l'Appendice 1.

8.4 Gestion de clé

Outre l'identité, un système fondé sur la technologie IBC s'appuie sur trois types de valeurs de clés cryptographiques: la clé secrète principale, les paramètres publics et la clé privée. La définition de ces structures de clés en syntaxe abstraite numéro un (ASN.1) figure dans l'Annexe B.

Le système fondé sur la technologie IBC a recours à cinq opérations différentes pour la gestion de ces clés:

1) l'initialisation du système;
2) l'initialisation du dispositif;
3) la vérification des paramètres publics;
4) la fourniture de clés et d'identités;
5) la révocation de clés et d'identités.

L'entité chargée de la gestion et le KMS peuvent échanger des messages au moyen du protocole d'interopérabilité de la gestion des clés (KMIP) [b-OASIS KMIP]. Il faut cependant définir l'extension du protocole d'interopérabilité de la gestion des clés (KMIP) requise pour répondre aux nouveaux besoins des fonctions **IBSetup** et **IBExtract**. Pour les dispositifs connectés à l'IoT munis d'une eUICC, on utilise les procédures standard pour la fourniture de clés à distance [b-GSMA SGP.02]. Pour les dispositifs connectés à l'IoT non munis d'une eUICC, les protocoles régissant les interactions entre les SecM et les entités responsables de la gestion sont définis à partir du protocole de transfert hypertexte (HTTP). Les spécifications de ces opérations figurent dans l'Annexe C.

L'opération d'initialisation du système consiste à initialiser un système fondé sur la technologie IBC en créant la MSK et les paramètres publics. Elle est normalement prise en charge par une entité responsable de la gestion, par exemple l'IdP, le SM-DP ou le MNO. Celle-ci met en place un canal sécurisé vers une entité de gestion des clés (KMS), qui exécute la fonction **IBSetup**. Les deux parties exécutent le KMIP dans le cadre de l'opération de création de la paire de clés. L'entité chargée de la gestion fournit au KMS les informations nécessaires pour lancer la fonction **IBCSetup** et produire la MSK ainsi que les paramètres publics. Le KMIP est étendu afin de prendre en charge les fonctions de paramétrage, notamment celles de plusieurs algorithmes d'IBC normalisés. L'opération est décrite en détail dans le paragraphe C.1.

L'opération d'initialisation du dispositif consiste à préparer un dispositif connecté à l'IoT à recevoir une identité et une clé. Il existe deux scénarios possibles: l'initialisation des dispositifs connectés à l'IoT munis d'une eUICC et l'initialisation des dispositifs connectés à l'IoT non munis d'une eUICC. En ce qui concerne les dispositifs pourvus d'une eUICC, celle-ci doit se charger de l'inscription auprès du service d'acheminement sécurisé du gestionnaire des abonnements (SM-SR) afin d'être prête à télécharger le profil [b-GSMA SGP.02]. Aucune autre opération n'est nécessaire pour les dispositifs à eUICC classiques. Le SecM des dispositifs connectés à l'IoT non dotés d'une eUICC doit d'abord contacter l'AuC pour obtenir un identificateur provisoire (PROV.ID) et un justificatif d'identité provisoire (PROV.CRED). La paire PROV.ID/PROV.CRED permet l'authentification des entités lors

de la fourniture d'identités et de clés. Si les dispositifs connectés à l'IoT ne parviennent pas à établir un canal sécurisé avec l'IdP au moyen du protocole de sécurité dans la couche transport (TLS), il faut en outre installer une identité de clé IdP.ID et la clé publique associée IdP.PUK dépendant de l'IdP ou des paramètres publics dans le SecM lors de l'initialisation du dispositif. L'opération est décrite en détail dans le § C.2.

L'opération de vérification des paramètres publics a pour but de récupérer les paramètres publics de l'IBC. Un dispositif connecté à l'IoT emploie la procédure de fourniture d'identité et de clé pour obtenir les paramètres publics du système à IBC auquel il appartient. Il peut suivre la spécification du § 4 de [IETF RFC 5408] pour obtenir les paramètres publics d'un autre système à IBC à partir du PPS connu. L'opération est décrite en détail dans le § C.3.

L'opération de fourniture d'identité et de clé comprend l'attribution d'une identité, l'extraction de la clé privée et la procédure de distribution de clé. Après le processus d'initialisation, les dispositifs connectés à l'IoT n'ont qu'une identité provisoire. C'est l'IdP, le SM-DP ou le MNO qui déterminent quelle identité doit être attribuée au dispositif demandeur, qui communiquent ensuite avec le KMS pour produire la clé privée correspondante et qui transmettent pour finir l'identité, la clé privée et les paramètres publics au dispositif de manière sécurisée. L'opération est décrite en détail dans le § C.4.

L'opération de révocation d'identité et de clé est utilisée lorsqu'une politique de sécurité stricte exige la révocation des identités en temps opportun. Si une identité est révoquée, son statut doit être changé en "révoquée". Si une entrée demande le statut d'une identité révoquée, l'IdP, le SM-DP ou le MNO doivent renvoyer la valeur correcte telle qu'elle est définie dans le protocole de statut de l'identité en ligne (OISP). Pour vérifier le statut d'un ensemble d'identités plus facilement, une entité peut récupérer régulièrement la liste de révocation d'identités (IRL) de l'IdP, du SM-DP ou du MNO, la conserver au niveau local, et vérifier à l'aide de l'IRL la plus récente si une identité a été révoquée sans avoir à demander le statut de chaque identité en ligne. L'opération est décrite en détail dans le § C.5.

8.5 Authentification

L'authentification est une procédure qui vise à déterminer si une entité (dispositif ou utilisateur) a le droit d'accéder à certaines ressources. Dans les réseaux de télécommunication, il existe deux sortes d'authentification qui concernent les dispositifs connectés à l'IoT: l'authentification pour l'accès au réseau et l'authentification pour les services. La première vise à déterminer si un dispositif est autorisé à accéder au réseau, tandis que la seconde vérifie si un dispositif a le droit de se connecter à une ISP.

Les protocoles d'authentification fondés sur les technologies de l'IBC conviennent pour l'authentification de dispositifs connectés à l'IoT au sein des réseaux de télécommunication. En effet, la technologie IBC allège considérablement le poids de la gestion des identités et des clés d'un grand nombre de dispositifs connectés à l'IoT. De plus, la technologie IBC rend possible l'authentification répartie, qui, d'une part, réduit fortement les délais d'authentification, et, d'autre part, permet de nouvelles applications, par exemple l'authentification entre dispositifs ou entre véhicules. Dans les réseaux de télécommunication actuels, tels que les réseaux 4G-LTE, la technologie IBC peut être employée en vue de l'authentification entre les dispositifs connectés à l'IoT et les ISP. Dans les réseaux cellulaires de cinquième génération, la technologie IBC peut être employée aussi bien pour l'authentification en vue de l'accès au réseau que pour l'authentification en vue de l'accès aux services. La spécification actuelle en matière de sécurité des réseaux 5G [b-ETSI TS 133.501] définit un cadre d'authentification unifié qui prend en charge les méthodes du protocole d'authentification extensible (EAP). L'Annexe de [b-ETSI TS 133.501] explique plus en détail comment utiliser l'EAP-TLS dans les réseaux 5G pour l'IoT.

Le cadre de l'EAP est ouvert et prend en charge un grand nombre de protocoles d'authentification, y compris l'EAP-TLS. Les méthodes d'authentification de l'EAP permettent d'utiliser aussi bien des clés symétriques que des clés asymétriques.

Étant donné qu'il s'agit d'une technologie relativement récente, rares sont les protocoles d'authentification existants qui prennent en charge la technologie IBC. L'Annexe D présente donc quatre protocoles existants amendés de sorte à prendre en charge la technologie IBC pour l'authentification:

1) paragraphe D.1: protocole de transport secret à passe unique [ISO/IEC 11770-3];
2) paragraphe D.2: sécurité de la couche transport avec la clé publique brute [IETF RFC 8446];
3) paragraphe D.3: sécurité de la couche transport grâce au protocole d'authentification extensible (EAP-TLS) [IETF RFC 5216];
4) paragraphe D.4: protocole d'authentification extensible avec partage préalable des clés (EAP-PSK) [IETF RFC 4764].

9 Exigences en matière de sécurité

La présente Recommandation porte uniquement sur les exigences en matière de sécurité relatives à l'utilisation de la technologie IBC dans le cadre de l'IoT. Les menaces et exigences générales en matière de sécurité pour l'IoT sont spécifiées dans [b-UIT-T X.1361]. Les principales préoccupations en matière de sécurité de tout système de cryptographie sont l'intégrité et l'authenticité des clés publiques ainsi que la confidentialité des clés secrètes éphémères et à long terme utilisées. Un système fondé sur la technologie IBC comprend les composantes suivantes: la MSK, les paramètres publics, les identificateurs, les clés privées et les secrets éphémères employés dans le cadre des opérations cryptographiques.

9.1 Exigences en matière de sécurité de la clé secrète principale

Toutes les clés privées sont produites à partir de la clé secrète principale (MSK). Ainsi, si sa sécurité est compromise, l'adversaire peut recréer la clé privée de n'importe quelle entité et, par conséquent, déchiffrer tous les messages protégés par la clé publique correspondante ou se faire passer pour cette entité. Tout accès illégal à la MSK constitue une menace pour la sécurité du système fondé sur la technologie IBC. La MSK doit donc être conservée dans un environnement renforcé, par exemple un module de sécurité matériel (HSM). Une authentification par des mécanismes de sécurité efficaces est obligatoire pour tout accès à la clé.

9.2 Exigence en matière de sécurité des paramètres publics

La clé publique est calculée à partir des paramètres publics et d'un identificateur à l'aide de l'opération **IBDerivate**. Par conséquent, l'utilisation d'un faux jeu de paramètres publics, produits par un adversaire, pour chiffrer un message ou vérifier une signature compromet la sécurité du message chiffré ou entraîne une conclusion erronée concernant l'identité de l'émetteur de la signature. Les paramètres publics doivent donc être transmis par un canal sécurisé ou accompagnés d'une signature valable. De plus, chaque entité doit procéder à une vérification de l'entité homologue à l'autre bout du canal sécurisé ou vérifier la validité de la signature par rapport à une clé publique fiable avant d'accepter les paramètres publics.

9.3 Exigence en matière de sécurité de l'identificateur

Dans l'IoT, chaque entité est dotée d'un identificateur. Si le même identificateur est attribué à plusieurs entités et que la clé privée qui lui correspond est fournie à chacune de ces entités, il existe un risque de divulgation d'informations sensibles ou d'attaques par usurpation d'identité. Il faut donc attribuer un identifiant unique à chaque dispositif.

9.4 Exigence en matière de sécurité de la clé privée

La clé privée risque d'être divulguée si l'environnement de sécurité d'un dispositif connecté à l'IoT est compromis. Par conséquent, la clé privée doit être distribuée par un canal sécurisé et conservée dans un environnement sûr.

9.5 Exigence en matière de sécurité des secrets éphémères

Les secrets éphémères, par exemple le secret aléatoire employé dans les procédures de chiffrement et de signature, risquent d'être divulgués si l'environnement de sécurité d'un dispositif connecté à l'IoT est compromis. Le caractère aléatoire de ces secrets éphémères doit donc être garanti.

Annexe A

Formulation générique et algorithmes relatifs à la cryptographie fondée sur l'identité

(Cette Annexe fait partie intégrante de la présente Recommandation.)

Cette Annexe contient une formulation générique de la cryptographie fondée sur l'identité (IBC) et offre une liste des algorithmes fondés sur la technologie IBC qui sont admis dans la présente Recommandation. En outre, les algorithmes qui suivent cette formulation générique, mais qui n'ont pas été intégrés à la liste ci-après pourront aisément y être ajoutés par la suite à titre d'extensions du cadre décrit. La formulation spécifique énoncée ici fournit également des indications quant aux descriptions des structures de données de clés, des opérations de gestion de clé, ainsi que de l'authentification et des protocoles d'établissement de clé, décrits dans les Annexes B à D.

Un système de cryptographie fondé sur la technologie IBC comprend les types de données de clés énumérés ci-dessous, ces clés étant classées selon la norme [ISO/CEI 18033-5]:

– *ib.msk*: la MSK est la valeur secrète utilisée par le KMS afin de calculer une clé privée fondée sur l'identité. *ib.msk* est générée lors du processus d'initialisation du système et n'est connue que du KMS;

– *ib.mpk*: il s'agit de la MPK qui est uniquement déterminée par la MSK correspondante. *ib.mpk* est calculée par le KMS lors du processus d'initialisation du système;

– *ib.sysparam*: il s'agit des paramètres qui entrent en jeu dans le calcul cryptographique, notamment le choix d'un dispositif ou d'une fonction de cryptographie spécifique au sein d'une famille de dispositifs ou de fonctions de cryptographie ou d'une famille d'espaces mathématiques. *ib.sysparam* est choisie par le KMS lors du processus d'initialisation du système;

– *ib.pubparam*: les paramètres publics constituent l'association des paramètres du système *ib.sysparam* et de la MPK *ib.mpk*. Ce type de clé est conçu pour offrir une vue unifiée concernant les normes internationales telles que la norme [ISO/CEI 18033-5] et les RFC liées à l'IBC, notamment le Document [IETF RFC 5091];

– *ib.prk*: il s'agit de la clé privée fondée sur l'identité, laquelle est générée par le KMS à l'aide d'*ib.msk* et d'*ib.pubparam*, correspondant à un identificateur *ID*;

– *ib.pub*: il s'agit de la clé publique fondée sur l'identité, laquelle est calculée à partir d'un identificateur *ID* et d'*ib.pubparam* au moyen d'une fonction définie par un schéma de cryptographie fondée sur l'identité.

Un système de cryptographie fondé sur la technologie IBC peut comprendre les fonctions suivantes, qui sont indiquées avec des paramètres d'entrée et de sortie.

IBSetup

Entrée: paramètre de sécurité

Sortie: *ib.pubparam*, *ib.msk*

IBExtract

Entrée: *ib.pubparam, ib.msk, ID*

Sortie: *ib.prk*

IBDerivate

Entrée: *ib.pubparam*, *ID*

Sortie: *ib.puk*

IBEnc

Entrée: *ib.pubparam*, *ID*, message *M*

Sortie: texte chiffré C

IBDec

Entrée: *ib.pubparam*, *ID*, *ib.prk*, texte chiffré *C*

Sortie: texte clair *M* ou erreur

IBSign

Entrée: *ib.pubparam*, *ID*, *ib.prk*, message *M*

Sortie: signature *S*

IBVerify

Entrée: *ib.pubparam*, *ID*, message *M*, signature *S*

Sortie: valable ou non valable

La présente Recommandation soutient l'utilisation des algorithmes fondés sur l'identité suivants:

– BB1-KEM (mécanisme d'encapsulation de la clé (KEM)) [IETF RFC 5091]
– BF-IBE [IETF RFC 5091]
– SK-KEM [IETF RFC 6508]
– SM9-IBE [b-GM/T 0044.2]
– Cha-Cheon-IBS (IBS2) [ISO/CEI 14888-3]
– ECCSI (elliptic curve-based certificateless signatures for identity-based encryption) [IETF RFC 6507]
– Hess-IBS (IBS1) [ISO/CEI 14888-3]
– SM9-IBS (norme chinoise relative à l'IBS) [ISO/CEI 14888-3]
– Fujioka-Suzuki-Ustaoglu-AKA (concordance de clés authentifiées (AKA)) [ISO/CEI 11770-3]
– Smart-Chen-Cheng-AKA [ISO/CEI 11770-3]
– SM9-AKA [b-GM/T 0044.2]
– Wang-AKA [b-IEEE P1363.3]

Tous ces algorithmes sont fondés sur l'hypothèse du logarithme discret et sont généralement utilisés sur le groupe de points d'une courbe elliptique. Un grand nombre de ces algorithmes ont également recours à un couplage cryptographique sur une courbe elliptique [b-Galbraith]. Un couplage cryptographique *e* est une application bilinéaire *e* efficacement calculable: $G1 \times G2 \rightarrow G3$, satisfaisant l'équation:

$$e([a]P1, [b]P2) = e(P1, P2)^{a*b}$$

où P1 et P2 constituent le générateur du groupe cyclique *G*1 et *G*2, respectivement. [*a*]P1 désigne le nombre *a* d'opérations du groupe avec P1. De la même façon, [*b*]P2 désigne l'opération du groupe avec P2.

Un couplage cryptographique peut être instancié par le couplage de Weil, le couplage de Tate, le couplage optimal Ate, etc., sur des courbes elliptiques adaptées aux couplages [b-Freeman]. Les courbes elliptiques adaptées aux couplages généralement utilisées comprennent les courbes elliptiques supersingulières, les courbes Barreto-Naehrig (BN), les courbes Barreto-Lynn-Scott de degré de prolongement 12 (BLS-12), les courbes Kachisa-Schaefer-Scott de degré de prolongement 16 (KSS-16), les courbes Kachisa-Schaefer-Scott de degré de prolongement 18 (KSS-18) et les courbes Barreto-Lynn-Scott de degré de prolongement 24 (BLS-24) [b-Freeman]. Toutes ces courbes *E* sont basées sur un champ premier, un champ fini de caractéristique p, F_p, où p est un nombre premier. *G*1 désigne le sous-groupe de points sur la courbe *E*. *G*2 est soit identique à *G*1, en cas d'utilisation de courbes supersingulières, soit un sous-groupe de points sur la courbe tordue *E'*. *E'* est obtenue à partir d'une extension du champ de base F_p. *G*3 désigne l'extension F_p^k du champ de base F_p, où k représente le degré de prolongement.

Des algorithmes fondés sur la technologie IBC sont élaborés au moyen d'autres mécanismes mathématiques, tels que les réseaux euclidiens (par exemple, [b-Ducas]). Ce type d'algorithmes est relativement efficace au calcul, tout en présentant une taille de clé et une taille en sortie plus importantes que les algorithmes fondés sur le logarithme discret sur courbes elliptiques. On pense souvent que les algorithmes sont résistants aux attaques d'ordinateurs quantiques. Cependant, les algorithmes de cette catégorie sont encore en développement. S'il semble ainsi prématuré de songer à la normalisation de ces algorithmes fondés sur la technologie IBC basés sur les réseaux euclidiens, leur intégration future peut toutefois être envisagée.

Annexe B

Spécification de données de clés relatives à la cryptographie fondée sur l'identité

(Cette Annexe fait partie intégrante de la présente Recommandation.)

À l'aide de la méthode standard décrite dans ASN.1, [IETF RFC 5408] a défini une structure générique des paramètres système, notamment *ib.pubparam* et d'autres informations complémentaires, et [IETF RFC 5091] a défini deux ensembles de structures de données de clés, dont *ib.msk* et *ib.prk* pour deux algorithmes fondés sur la technologie IBE, à savoir BF-IBE et BB1-IBE. La présente Recommandation, tout en maintenant la compatibilité avec les définitions existantes, élargit la définition des paramètres système et fixe de nouvelles structures de données de clés en vue de la prise en charge d'un plus grand nombre d'algorithmes et de diverses mises en œuvre efficaces avec des courbes et couplages différents.

Une structure générique des paramètres système est définie comme suit:

```
IBSysParams::= SEQUENCE {
        version                 INTEGER { v3(3) },
        domainName              IA5String,
        domainSerial            INTEGER,
        validity                ValidityPeriod,
        ibPublicParameters      IBPublicParameters,
        ibIdentityType          OBJECT IDENTIFIER,
        ibParamExtensions       [0] IMPLICIT IBParamExtensions OPTIONAL,
        signatureAlgorithm      [1] IMPLICIT AlgorithmIdentifier OPTIONAL,
        signature               [2] IMPLICIT BIT STRING OPTIONAL
}
```

IBSysParams correspond à la définition d'IBESysParams présentée dans [IETF RFC 5408]. Toutefois, cette version a été remplacée par la version v3 (3) et deux champs supplémentaires ont été ajoutés. districtName et districtSerial ont respectivement été renommés domainName et domainSerial. La définition d'IBPublicParameter a été modifiée, passant du type chaîne d'octets (OCTET STRING) au type nouvellement défini IBParameterData, qui constitue un choix (CHOICE) déterminé par la valeur de pkgAlgorithm. Cette définition élimine le double codage inutile occasionné par la définition précédente, à savoir le codage de publicParameterData comme une séquence (SEQUENCE) de BFPublicParameters, par exemple, puis le codage du résultat comme une chaîne d'octets (OCTET STRING). Hormis les deux nouveaux champs créés, la signification des autres champs reste la même, conformément à [IETF RFC 5408]. Les significations des deux nouveaux champs sont les suivantes:

– Signature Algorithm désigne l'algorithme de signature utilisé pour générer la valeur de signature. Ce champ est facultatif, le champ signature n'étant pas obligatoire.

– Le champ signature comprend la signature numérique calculée sur la base du résultat associé aux règles de codage distinctives (DER) de l'ASN.1, de la version du champ à *ibParamExtensions*. Ce champ, codé en tant que chaîne binaire (BIT STRING), est facultatif.

S'il est présent, le champ signature sert à aider une entité à vérifier l'authenticité des paramètres publics du système sans avoir recours à d'autres méthodes. Par exemple, si un dispositif connecté à l'IoT n'est pas en mesure, conformément aux exigences du Document [IETF RFC 5408], d'établir un canal sécurisé basé sur le protocole TLS pour obtenir les paramètres publics d'un autre système à IBC, il peut présenter une requête HTTP à son serveur de paramètres publics (PPS). Dans ce cas, le PPS sollicité doit signer les paramètres publics demandés à l'aide de sa clé de signature privée. Le dispositif connecté à l'IoT peut ensuite vérifier la signature présentée pour s'assurer de l'authenticité de la réponse. Si un PPS publie les paramètres publics d'un autre système à IBC sur ses entités serveuses, il est recommandé de considérer le message de signature comme une identité et d'utiliser l'algorithme **IBExtract** à titre d'algorithme de signature afin de générer la clé privée, qui constitue la valeur de signature correspondante. De cette manière, les dispositifs connectés à l'IoT vérifient que la signature constitue une clé privée valide correspondant au résultat associé aux DER de l'ASN.1, de la version du champ à *ibParamExtensions*, sans qu'une clé publique de vérification supplémentaire soit nécessaire pour vérifier la signature.

```
ValidityPeriod ::=  SEQUENCE {
        notBefore       GeneralizedTime,
        notAfter        GeneralizedTime
}
IBPublicParameters::= SEQUENCE SIZE (1..MAX) OF IBPublicParameter
IBPublicParameter::= SEQUENCE {
        pkgAlgorithm            OBJECT IDENTIFIER,
        publicParameterData IBParameterData
}
```

La valeur de publicParameterData est définie par pkgAlgorithm. Il peut s'agir de l'une des possibilités ci-dessous.

```
IBParameterData::= CHOICE{
        bb1ParameterData    [0] IMPLICIT BB1PublicParameters,
        bfParameterData     [1] IMPLICIT BFPublicParameters,
        eccsiParameterData  [2] IMPLICIT ECCSIPublicParameters,
        skParameterData     [3] IMPLICIT SKPublicParameters,
        sm9ParameterData    [4] IMPLICIT SM9PublicParameters
}
IBParamExtensions::= SEQUENCE OF IBParamExtension
IBParamExtension::= SEQUENCE {
        ibParamExtensionOID         OBJECT IDENTIFIER,
        ibParamExtensionValue       OCTET STRING
}
AlgorithmIdentifier ::= SEQUENCE {
        algorithm           OBJECT IDENTIFIER,
        parameters           ANY DEFINED BY algorithm OPTIONAL
}
```

Dans [IETF RFC 5091], deux ensembles de MSK, de paramètres publics et de bloc de clé privée, à savoir:

– BB1MasterSecret, BB1PublicParameters, BB1PrivateKeyBlock; et

– BFMasterSecret, BFPublicParameters, BFPrivateKeyBlock sont définis pour la fonction de génération de clés privées des systèmes de chiffrement BF et BB1. Ces appellations ne peuvent désigner que les mises en œuvre des fonctions présentant des couplages symétriques sur des courbes elliptiques supersingulières définies sur des champs premiers. La présente Recommandation offre une spécification de nouvelles structures dont la version a été remplacée par la version v3 en vue de prendre en charge les implémentations de ces algorithmes avec des couplages asymétriques. En ce qui concerne les couplages symétriques sur des courbes elliptiques supersingulières, le champ correspondant dans les structures de données de clés BB1 et BF reste le même, conformément à [IETF RFC 5091]. Trois autres ensembles de structures de données de clés sont définis pour ECCSI, SM9 et SK-KEM, respectivement.

```
BB1MasterSecret::= SEQUENCE {
        version     INTEGER { v3(3) },
        alpha       INTEGER,
        beta        INTEGER,
        gamma       INTEGER
}
```

– Pour les implémentations avec couplages asymétriques, alpha correspond à s1, beta à s2 et gamma à s3 dans le § 9.3 de la norme [ISO CEI 18033-5].

```
BB1PublicParameters::= SEQUENCE {
        version     INTEGER { v3(3) },
        curve       OBJECT IDENTIFIER,
        hashfcn     OBJECT IDENTIFIER,
        pairing     PAIRING OPTIONAL,
        p           INTEGER OPTIONAL,
        q           [0] IMPLICIT INTEGER OPTIONAL,
        pointP      FpPoint,
        pointQ      [1] EXPLICIT FpxPoint OPTIONAL
        pointP1     FpPoint,
        pointP2     [2] EXPLICIT FpxPoint OPTIONAL,
        pointP3     FpPoint,
        v           FpxElement
}
```

– Le couplage indique le type d'application bilinéaire qui doit être utilisé avec les paramètres générés. Les trois types de couplage suivants sont pris en charge: le couplage de Weil, le couplage de Tate et le couplage optimal Ate.

– p et q deviennent facultatifs. Pour certains types de courbes, par exemple, BN et BLS-12, p et q sont prédéterminés par des identificateurs d'objet (OID) de courbes et il est, par conséquent, inutile de les spécifier à nouveau.

– Dans les mises en œuvre avec des couplages asymétriques, pointP et pointQ correspondent à Q1 dans *G*1 et à Q2 dans *G*2 dans le § 9.3 de la norme [ISO/CEI 18033-5]. Dans le cas des couplages symétriques, pointP équivaut à pointQ. Par conséquent, pointQ est FACULTATIF.

– Dans les mises en œuvre avec des couplages asymétriques, pointP1 et pointP3 correspondent à R et à T dans le § 9.3 de la norme [ISO/CEI 18033-5].

– Dans les mises en œuvre avec des couplages asymétriques, tels que le couplage optimal Ate sur des courbes BN, la valeur de pointP2 dépend d'une extension du champ F_p. Si v est indiqué, pointP2 est facultatif car il n'est pas nécessaire à l'exécution de l'algorithme BB1-KEM.

– v désigne le résultat du couplage, qui est un élément du champ d'extension de F_p. Dans le cas de l'implémentation avec des couplages asymétriques, tels que le couplage optimal Ate sur des courbes BN, F_p^k désigne le champ d'extension, où k est le degré de prolongement. Dans cette situation, v correspond à J dans le § 9.3 de la norme [ISO/CEI 18033-5].

– La signification des autres champs reste la même, conformément à [IETF RFC 5091].

```
PAIRING::= ENUMERATED{
        weil        (1),    --couplage de Weil
        tate        (2),    --couplage de Tate
        optimalAte  (3)     --couplage optimal Ate
}
FpPoint::= SEQUENCE{
        x   INTEGER,
        y   INTEGER
}
```

FpPoint désigne un point sur une courbe elliptique définie sur un champ premier. Tout point possède deux coordonnées, qui sont désignées sous les noms de coordonnée x et coordonnée y. Ces deux coordonnées ont pour valeur des grands entiers.

```
FpxPoint::= CHOICE{
        fpPoint     [1] EXPLICIT FpPoint,
        fp2Point    [2] EXPLICIT Fp2Point,
        fp3Point    [3] EXPLICIT Fp3Point,
        fp4Point    [4] EXPLICIT Fp4Point
}
```

– Fp2Point désigne un point sur une courbe elliptique définie sur un champ F_p^2. Chacune des coordonnées d'un point a pour valeur un élément de F_p^2.

– Fp3Point désigne un point sur une courbe elliptique définie sur un champ F_p^3. Chacune des coordonnées d'un point a pour valeur un élément de F_p^3.

– Fp4Point désigne un point sur une courbe elliptique définie sur un champ F_p^4. Chacune des coordonnées d'un point a pour valeur un élément de F_p^4.

Fp2Point::= SEQUENCE{

x Fp2Element,

y Fp2Element

}

– Fp2Point désigne un point sur une courbe elliptique définie sur un champ F_p^2. Tout point possède deux coordonnées, qui sont désignées sous les noms de coordonnée x et coordonnée y. Ces deux coordonnées ont pour valeurs des éléments de F_p^2.

Fp3Point::= SEQUENCE{

x Fp3Element,

y Fp3Element

}

– Fp3Point désigne un point sur une courbe elliptique définie sur un champ F_p^3. Chacune des coordonnées d'un point a pour valeur un élément de F_p^3.

Fp4Point::= SEQUENCE{

x Fp4Element,

y Fp4Element

}

– Fp4Point désigne un point sur une courbe elliptique définie sur un champ F_p^4. Chacune des coordonnées d'un point a pour valeur un élément de F_p^4.

Fp2Element::= SEQUENCE{

a INTEGER,

b INTEGER

}

– Fp2Element désigne un élément d'un champ F_p^2 représenté par a+bα, où α est une racine non quadratique dans F_p.

Fp3Element::= SEQUENCE{

a INTEGER,

b INTEGER,

c INTEGER

}

– Fp3Element désigne un élément d'un champ F_p^3 représenté par a+bβ+cβ^2, où β est une racine non cubique dans F_p.

Fp4Element::= SEQUENCE{

a Fp2Element,

b Fp2Element

}

– Fp4Element désigne un élément d'un champ F_p^4 représenté par une tour composée de deux éléments de F_p^2.

FpxElement::= CHOICE{

fp2Elemt [1] EXPLICIT Fp2Element,

--pour la mise en œuvre de courbes elliptiques supersingulières

fp12Elemt [2] EXPLICIT Fp12Element,

--au moyen de la représentation sous forme de tour F_p->F_p^2->F_p^6->F_p^12

fp16Elemt [3] EXPLICIT Fp16Element,

--au moyen de la représentation sous forme de tour F_p->F_p^2->F_p^4->F_p^8->F_p^16

fp18Elemt [4] EXPLICIT Fp18Element,

--au moyen de la représentation sous forme de tour F_p->F_p^3->F_p^6->F_p^18

fp24Elemt [5] EXPLICIT Fp24Element

--au moyen de la représentation sous forme de tour F_p->F_p^2->F_p^6->F_p^12->F_p^24

}

– FpxElement désigne la représentation sous forme de tour d'un élément de *G*3. Le couplage e met en correspondance deux données d'entrée de *G*1 et *G*2, respectivement, à un élément de *G*3. Pour les courbes adaptées aux couplages couramment utilisées, les éléments de *G*3 sont généralement représentés à l'aide de la méthode de la tour. Il peut exister différentes représentations en tour pour les divers degrés de prolongement. La présente Recommandation définit une représentation en tour d'éléments couramment utilisée dans les champs associés aux degrés de prolongement 12, 16, 18 et 24.

Fp12Element::= SEQUENCE{

a Fp6Element,

b Fp6Element

}

– Fp12Element définit un élément de F_p^12 associé à une représentation en tour de type 2x3x2 et doit être utilisé dans les mises en œuvre avec des courbes BN ou des courbes BLS-12 ou BLS-24.

Fp6Element::= SEQUENCE{

a Fp2Element,

b Fp2Element,

c Fp2Element

}

– Fp6Element définit un élément de F_p^6 associé à une représentation en tour de type 3x2 et doit être utilisé dans les mises en œuvre avec des courbes BN ou des courbes BLS-12 ou BLS-24.

Fp16Element::= SEQUENCE{

a Fp8Element,

b Fp8Element

}

– Fp16Element définit un élément de F_p^16 associé à une représentation en tour de type 2x2x2x2 et doit être utilisé dans les mises en œuvre avec des courbes KSS-16.

Fp8Element::= SEQUENCE{

a Fp4Element,

b Fp4Element

}

– Fp8Element définit un élément de F_p^8 associé à une représentation en tour de type 2x2x2 et doit être utilisé dans les mises en œuvre avec des courbes KSS-16.

Fp18Element::= SEQUENCE{

a Fp6bElement,

b Fp6bElement,

c Fp6bElement

}

– Fp18Element définit un élément de F_p^18 associé à une représentation en tour de type 3x2x3 et doit être utilisé dans les mises en œuvre avec des courbes KSS-18.

Fp6bElement::= SEQUENCE{

a Fp3Element,

b Fp3Element

}

– Fp6bElement définit un élément de F_p^6 associé à une représentation en tour de type 2x3 et doit être utilisé dans les mises en œuvre avec des courbes KSS-18.

Fp24Element::= SEQUENCE{

a Fp12Element,

b Fp12Element

}

– Fp24Element définit un élément de F_p^24 associé à une représentation en tour de type 2x2x3x2 et doit être utilisé dans les mises en œuvre avec des courbes BLS-24.

BB1PrivateKeyBlock::= SEQUENCE {

version INTEGER { v3(3) },

pointD0FpxPoint,

pointD1FpxPoint

}

– La signification de pointD0 et de pointD1 reste la même, conformément au Document [IETF RFC 5091], mais elle découle de G2 lorsque BB1-KEM est mis en œuvre avec des couplages asymétriques. Dans ce cas, pointD0 et pointD1 correspondent, respectivement, à dID0 et dID1 dans le § 9.3 de la norme [ISO/CEI 18033-5].

```
BFMasterSecret::= SEQUENCE {
        version         INTEGER {v3(3) },
        masterSecret    INTEGER
}
```

– La signification de chacun des champs reste la même, conformément à [IETF RFC 5091].

```
BFPublicParameters::= SEQUENCE {
        version         INTEGER { v3(3) },
        curve           OBJECT IDENTIFIER,
        hashfcn         OBJECT IDENTIFIER,
        pairing         PAIRING OPTIONAL,
        p               INTEGER OPTIONAL,
        q               [0] IMPLICIT INTEGER OPTIONAL,
        pointP          FpxPoint,
        pointPpub       FpxPoint
}
```

– La signification de chacun des champs reste la même, conformément au Document [IETF RFC 5091], mais pointP et pointPpub découlent de *G*2 lorsque BF-IBE est mis en œuvre avec des couplages asymétriques. Dans ce cas, pointP et pointPpub correspondent, respectivement, à Q et à R dans le § 8.2 de la norme [ISO/CEI 18033-5].

```
BFPrivateKeyBlock::= SEQUENCE {
        version         INTEGER { v3(3) },
        privateKey      FpPoint
}
```

– La signification de chacun des champs reste la même, conformément à [IETF RFC 5091]. Dans les mises en œuvre avec des couplages asymétriques, privateKey correspond à skID dans le § 8.2 de la norme [ISO/CEI 18033-5].

```
ECCSIMasterSecret::= SEQUENCE {
        version         INTEGER {v3(3) },
        masterSecret    INTEGER
}
```

– masterSecret correspond à KSAK dans le Document [IETF RFC 6507].

```
ECCSIPublicParameters::= SEQUENCE {
        version         INTEGER { v2(2) },
        curve           OBJECT IDENTIFIER,
        hashfcn         OBJECT IDENTIFIER,
        pointP          FpPoint,
        pointPpub       FpPoint
}
```

– pointP correspond à G dans le Document [IETF RFC 6507].

– pointPpub correspond à la clé d'authentification publique du KMS (KPAK) dans le Document [IETF RFC 6507].

```
ECCSIPrivateKeyBlock::= SEQUENCE {
        version     INTEGER { v2(2) },
        ssk         INTEGER ,
        pvt         OCTET STRING
}
```

– ssk et pvt sont, respectivement, une clé de signature secrète (SSK) et un jeton de vérification public (PVT) dans le Document [IETF RFC 6507].

```
SKMasterSecret::= SEQUENCE {
        version         INTEGER {v3(3) },
        masterSecret    INTEGER
}
```

– masterSecret correspond à z_T dans le Document [IETF RFC 6508] et à s dans le § 9.2 de la norme [ISO/CEI 18033-5].

```
SKPublicParameters::= SEQUENCE {
        version     INTEGER { v3(3) },
        curve       OBJECT IDENTIFIER,
        hashfcn     OBJECT IDENTIFIER,
        pairing     PAIRING OPTIONAL,
        p           INTEGER OPTIONAL,
        q           [0] IMPLICIT INTEGER OPTIONAL,
        pointP1     FpPoint,
        pointP1pub  [1] EXPLICIT FpPoint OPTIONAL,
        pointP2     [2] EXPLICIT FpxPoint OPTIONAL,
        pointP2pub  [3] EXPLICIT FpxPoint OPTIONAL,
        v           [4] EXPLICIT FpxElement
}
```

– Dans les mises en œuvre avec des couplages symétriques sur des courbes supersingulières, p et q sont définis dans le Document [IETF RFC 5091]. Dans les mises en œuvre avec des couplages asymétriques, p et q sont prédéterminés par la courbe utilisée et deviennent facultatifs.

– pointP1 correspond à P dans le Document [IETF RFC 6508] et Q1 correspond à *G*1 dans le § 9.2 de la norme [ISO/CEI 18033-5].

– pointP1pub correspond à Z_T dans le Document [IETF RFC 6508] et à R dans le § 9.2 de la norme [ISO/IEC 18033-5]. Il est possible que pointP1pub ne soit pas nécessaire pour d'autres algorithmes, tels que les algorithmes de signature, fondés sur la fonction de génération Sakai-Kasahara (SK) ; il est donc facultatif.

– pointP2 correspond à Q2 dans *G*2 dans le § 9.2 de la norme [ISO/IEC 18033-5] lorsque le SK-KEM est mis en œuvre avec des couplages asymétriques. pointP2 n'est pas indispensable au fonctionnement de SK-KEM, il est donc facultatif.

– pointP2pub correspond à [*ib.msk*]Q2, qui n'est pas nécessaire au fonctionnement de SK-KEM, mais peut être utile à d'autres algorithmes, tels que les algorithmes de signature, fondés sur la fonction de génération de clés SK. Il est donc facultatif.

```
SKPrivateKeyBlock::= SEQUENCE {
        version      INTEGER { v3(3) },
        privateKey   FpxPoint
}
```

– privateKey correspond à RSK dans le Document [IETF RFC 6508] et à skID dans le § 9.2 de la norme [ISO/CEI 18033-5].

```
SM9MasterSecret::= SEQUENCE {
        version      INTEGER {v3(3) },
        masterSecret INTEGER
}
```

– masterSecret correspond à *ib.msk*, qui est défini par U dans le § 7.4 de la norme [b-ISO/CEI 14888-3a].

```
SM9PublicParameters::= SEQUENCE {
        version      INTEGER { v3(3) },
        curve        OBJECT IDENTIFIER,
        hashfcn      OBJECT IDENTIFIER,
        pairing      PAIRING OPTIONAL,
        p            INTEGER OPTIONAL,
        q            [0] IMPLICIT INTEGER OPTIONAL,
        pointP1      FpPoint,
        pointP1pub   [1] EXPLICIT FpPoint OPTIONAL,
        pointP2      [2] EXPLICIT FpxPoint OPTIONAL,
        pointP2pub   [3] EXPLICIT FpxPoint OPTIONAL,
        v            [4] EXPLICIT FpxElement
}
```

– Dans les mises en œuvre avec des couplages symétriques sur des courbes supersingulières, p et q sont tels que définis dans le Document [IETF RFC 5091]. Dans les mises en œuvre avec des couplages asymétriques, p et q sont prédéterminés par la courbe utilisée.

– pointP1 correspond à P dans le § 7.4 de la norme [ISO/CEI 14888-3].

– pointP1pub n'est pas nécessaire pour SM9-IBS mais il l'est pour SM9-IBE. Dans ce cas, pointP2pub correspond à [*ib.msk*]P.

– pointP2 correspond à Q dans le § 7.4 de la norme [ISO/CEI 14888-3]. pointP2 n'est pas indispensable au fonctionnement de **SM9-IBE**, il est donc facultatif.

– pointP2pub correspond à V dans le § 7.4 de la norme [ISO/CEI 14888-3a]. pointP2pub n'est pas indispensable au fonctionnement de **SM9-IBE**, il est donc facultatif.

```
SM9PrivateKeyBlock::= SEQUENCE {
        version      INTEGER { v3(3) },
```

privateKey FpxPoint

}

– privateKey correspond à X dans le § 7.4 de la norme [ISO/CEI 14888-3a] en ce qui concerne la signature et correspond au *ib.prvk* dans *G*1 pour SM9-IBE et SM9-AKA.

Il convient de se référer à la définition de BFMasterSecret, de BFPublicParameters et de BFPrivateKeyBlock pour les algorithmes fondés sur la génération de clés Sakai-Ohgishi-Kasahara (SOK), tels que BF-IBE, Cha-Cheon-IBS, Hess-IBS, Fujioka-Suzuki-Ustaoglu-AKA, Smart-Chen-Cheng-AKA et Wang-AKA. Il convient de se référer à la définition de BB1MasterSecret, de BB1PublicParameters et de BB1PrivateKeyBlock pour les algorithmes fondés sur la génération de clés BB1, tels que BB1-KEM. SKMasterSecret, SKPublicParameters et SKPrivateKeyBlock doivent être utilisés pour SK-KEM et, éventuellement, pour d'autres algorithmes fondés sur la fonction de génération de clés SK. SM9MasterSecret, SM9PublicParameters et SM9PrivateKeyBlock doivent être utilisés pour les algorithmes SM9, dont SM9-IBE, SM9-IBS et SM9-AKA. ECCSIMasterSecret, ECCSIPublicParameters et ECCSIPrivateKeyBlock doivent être utilisés pour le schéma de signature ECCSI.

Si la clé privée doit être protégée, la structure EncryptedPrivateKeyInfo, telle que définie dans le Document [IETF RFC 5958] doit être utilisée.

EncryptedPrivateKeyInfo::= SEQUENCE {

encryptionAlgorithm EncryptionAlgorithmIdentifier,

encryptedData EncryptedData

}

EncryptionAlgorithmIdentifier::= AlgorithmIdentifier

EncryptedData::= OCTET STRING

AlgorithmIdentifier::= SEQUENCE {

algorithm OBJECT IDENTIFIER,

parameters ANY DEFINED BY algorithm OPTIONAL

}

Annexe C

Opérations de gestion de clés

(Cette Annexe fait partie intégrante de la présente Recommandation.)

Dans un fondé sur la technologie IBC, les opérations de gestion de clés comprennent l'initialisation du système, la fourniture d'identités ou de clés privées, la révocation de clés et d'identités et la publication des paramètres du système. L'initialisation du système comprend une étape consistant à lancer la fonction **IBSetup** et la fourniture de clés privées comprend une étape consistant à lancer la fonction **IBExtract**. Ces opérations requièrent des interactions entre une entité de gestion et le KMS. La présente Recommandation fait appel au KMIP aux fins de l'échange de messages entre ces deux parties. L'extension nécessaire à la satisfaction des nouvelles exigences des algorithmes **IBSetup** et **IBExtract** pris en charge est indiquée dans l'Appendice II. Les protocoles régissant les interactions entre les SecM et les entités responsables de la gestion sont définis à partir du protocole HTTP pour les dispositifs connectés à l'IoT non munis d'une eUICC. Pour les eUICC, les normes [b-GSMA SGP.02] sont appliquées et élargies au besoin.

C.1 Initialisation du système

Dans chaque système à IBC, un processus d'initialisation du système doit être mené à bien avant de fournir des KMS aux utilisateurs. Au cours de ce processus, le KMS exécute une ou plusieurs fonctions **IBSetup** afin de générer un ou plusieurs ensembles de paires de clés *ib.msk* et *ib.pubparam*. La méthode permettant de renforcer la sécurité du KMS n'entre pas dans le cadre de la présente Recommandation. La génération et le stockage d'*ib.msk* dans un HSM constituent de bonnes pratiques. Lorsque cela est possible, on procèdera au déploiement d'un mécanisme de génération de clés décentralisée utilisant un système de partage secret afin de fractionner *ib.msk* et de répartir les parties secrètes et la fonction de génération de clés privées dans divers KMS. Dans cette situation, la génération en bonne et due forme d'une clé privée correspondant à un identificateur ne pourra avoir lieu que si le nombre de KMS fonctionnant convenablement dépasse un seuil donné.

Voir la Figure C.1.

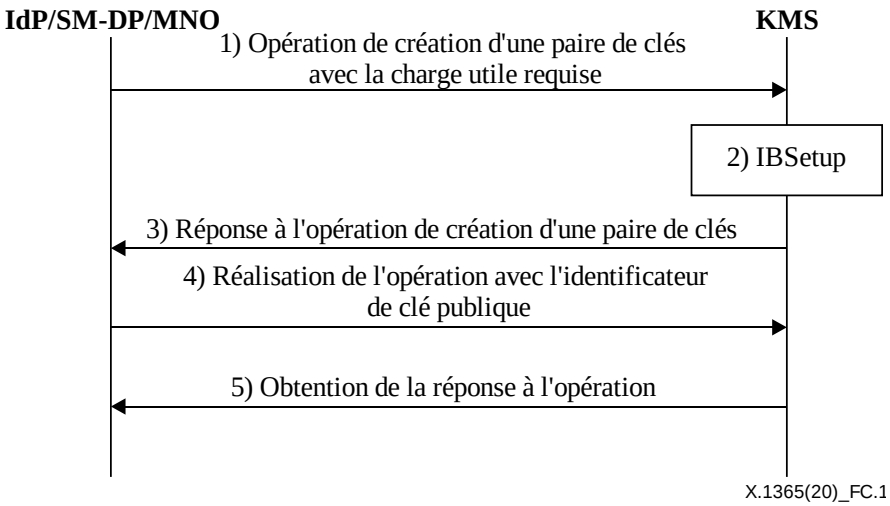

Figure C.1 – Initialisation du système à l'aide du protocole d'interopérabilité de la gestion des clés

Conditions de démarrage:

On suppose que l'IdP/SM-DP/MNO joue le rôle d'initiateur du système et est responsable du processus d'initialisation de ce dernier. Avant que l'IdP/SM-DP/MNO puisse lancer la fonction **IBSetup** dans un KMS, les conditions suivantes doivent être remplies:

a) Un canal sécurisé a été établi entre l'IdP/SM-DP/MNO et le KMS.

b) L'IdP/SM-DP/MNO a suivi un processus d'authentification auprès du KMS et l'IdP/SM-DP/MNO authentifié a reçu l'autorisation de procéder à la demande **IBSetup**.

Procédure:

1) L'IdP/SM-DP/MNO prépare la charge utile des demandes et lance l'opération de création de paires de clés afin d'envoyer un message de demande codé au KMS.

2) Le KMS vérifie la validité de la demande et s'assure que l'IdP/SM-DP/MNO est autorisé à lancer cette opération. Si l'une de ces conditions n'est pas satisfaite, le KMS envoie une réponse indiquant un échec. À défaut, le KMS exécute **IBSetup** avec les paramètres indiqués dans la demande.

3) Le KMS renvoie la réponse liée à l'exécution à l'IdP/SM-DP/MNO. Lorsque l'opération a été réalisée avec succès, le KMS transmet, au minimum, un identificateur unique de clé privée à *ib.msk* et un identificateur unique de clé publique à *ib.pubparam*.

4) À titre facultatif, une fois l'opération de création de paires de clés menée à bien, SM-DP/MNO peut lancer l'opération d'obtention à l'aide de l'identificateur unique de clé publique obtenu dans la dernière réponse afin de récupérer les paramètres publics *ib.pubparam*.

5) Le KMS envoie la valeur de la clé des nouveaux paramètres publics générés.

L'extension du KMIP visant à prendre en charge cette opération est décrite dans l'Appendice II.

Condition d'achèvement: Le KMS est initialisé avec succès et l'IdP/SM-DP/MNO dispose de l'identificateur unique de clé privée et de l'identificateur unique de clé publique afin d'accéder, respectivement, à la MSK *ib.msk* et aux paramètres publics *ib.pubparam*. L'IdP/SM-DP/MNO utilise l'identificateur unique de clé privée pour appeler l'opération de signature en vue de générer des clés privées d'identité et utilise l'identificateur unique de clé publique pour appeler l'opération d'obtention en vue de récupérer les paramètres publics.

C.2 Initialisation du dispositif

L'opération d'initialisation du dispositif consiste à préparer le dispositif à recevoir une identité et une clé. Selon que l'on a affaire à des dispositifs à eUICC ou à d'autres dispositifs non munis d'une eUICC, des procédures d'initialisation distinctes doivent être suivies.

C.2.1 Cas 1: Initialisation des dispositifs à eUICC

Dans le cas des dispositifs à eUICC, l'identité ainsi que la clé privée *ib.prk* et les paramètres publics *ib.sysparam* correspondants sont téléchargés sur le profil du domaine de sécurité de l'émetteur (ISD). Par conséquent, une fois le processus d'initialisation du dispositif achevé, l'eUICC devrait être prête pour la création du profil de l'ISD. Conformément à la norme [b-GSMA SGP.02], l'opération d'enregistrement est menée à bien. Ce qui suit reprend le § 3.5.1 de la norme [b-GSMA SGP.02].

- **Enregistrement de l'eUICC auprès du service d'acheminement sécurisé du gestionnaire des abonnements (SM-SR)**

Conditions de démarrage:

a) Les eUICC sont produites et un profil d'approvisionnement est chargé et activé au sein du réseau de l'opérateur d'approvisionnement. Après avoir été soumises à des tests, elles sont prêtes à être expédiées. Chaque eUICC est associée à un jeu d'informations de l'eUICC (EIS).

Procédure:

1) Le fabricant d'eUICC (EUM) envoie une demande d'enregistrement d'eUICC contenant l'EIS au SM-SR sélectionné.
2) Le SM-SR stocke l'EIS dans sa base de données, l'eUICC-ID (EID) constituant le paramètre de clé.
3) Le SM-SR confirme l'enregistrement à l'EUM. Le message de confirmation contient l'EID.

Condition d'achèvement: l'eUICC est enregistrée auprès du SM-SR et prête pour le téléchargement du profil. Elle peut désormais être expédiée au fabricant du dispositif machine à machine.

C.2.2 Cas 2: Initialisation des dispositifs connectés à l'IoT non munis d'une eUICC

Pour les dispositifs connectés à l'IoT non munis d'une eUICC, l'opération d'enregistrement suivante doit être réalisée.

- **Enregistrement du module de sécurité (SecM) auprès du centre d'authentification (AuC)**

Conditions de démarrage:

a) Le SecM a été produit et le dispositif connecté à l'IoT peut communiquer avec le fournisseur d'identité (IdP) sur le réseau de l'opérateur.

Procédure:

1) Le SecM envoie une demande d'acquisition de données d'approvisionnement pour SecM à l'AuC.
2) L'AuC génère un identificateur d'approvisionnement (PROV.ID) et le justificatif d'authentification y étant associé (PROV.CRED) pour le SecM ayant soumis la demande.
3) L'AuC envoie le PROV.ID et le PROV.CRED au SecM. Dans le même message, l'AuC envoie également au SecM soit une identité de clé IdP.ID et la clé publique y étant associée IdP.PUK, soit *ib.sysparam* si le SecM n'est pas en mesure de mettre en œuvre le protocole TLS.
4) Le SecM stocke le PROV.ID et le PROV.CRED de façon sécurisée et stocke également l'IdP.ID et l'IdP.PUK ou *ib.sysparam*, s'ils lui ont été fournis. Le SecM protège l'IdP.ID et l'IdP.PUK ou *ib.sysparam* de toute modification autorisée.

Condition d'achèvement: Le SecM est enregistré auprès de l'AuC et prêt à recevoir une identité et une clé.

C.3 Vérification des paramètres publics

Une entité doit avoir recours à la procédure d'approvisionnement d'identité ou de clé pour obtenir les paramètres publics correspondant au système à IBC avec lequel elle s'est enregistrée. Toute entité, qui peut être un dispositif connecté à l'IoT ou une entité de gestion d'un système à IBC, doit suivre la spécification du paragraphe 4 du Document [IETF RFC 5408] pour obtenir les paramètres publics d'un autre système à IBC à partir du PPS connu. Les IBESysParams fournis dans la réponse décrits dans le Document [IETF RFC 5408] sont remplacés par les IBSysParams définis dans la présente Recommandation. Le Document [IETF RFC 5408] présume que le dispositif connecté à l'IoT présentant la requête est en mesure d'établir un canal sécurisé basé sur le protocole TLS avec le PPS demandé. Si cette exigence ne peut être satisfaite, le signatureAlgorithm et le champ signature des IBSysParams existent et sont valides. Une fois les IBSysParams obtenus, un processus de vérification des signatures en bonne et due forme doit être suivi. Les paramètres publics obtenus ne seront acceptés que si la signature dans IBSysParams est valide et que la clé publique de vérification des signatures est authentique et valide.

C.4 Fourniture de clés et d'identités

L'opération de fourniture d'identité et de clé comprend l'attribution d'une identité, l'extraction de la clé privée et la procédure de distribution de clé. Après le processus d'initialisation, les dispositifs connectés à l'IoT n'ont qu'une identité provisoire. C'est l'IdP, le SM-DP ou le MNO qui déterminent quelle identité doit être attribuée au dispositif demandeur, qui communiquent ensuite avec le KMS pour produire la clé privée correspondante et qui transmettent pour finir l'identité, la clé privée et les paramètres publics au dispositif de manière sécurisée.

Voir la Figure C.2.

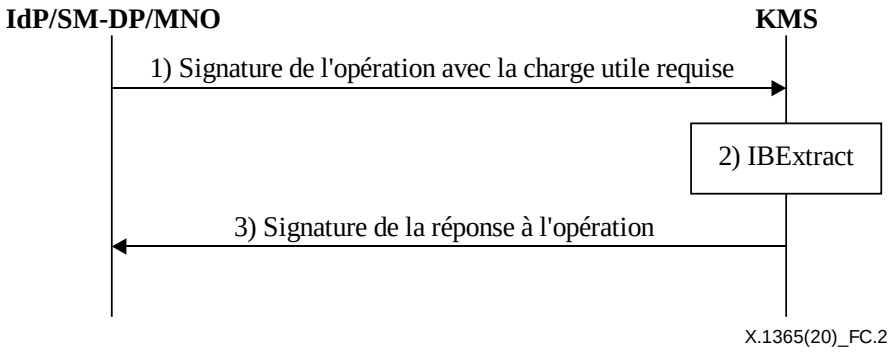

Figure C.2 – Génération d'une clé privée à l'aide du protocole d'interopérabilité de la gestion des clés

- **Génération d'une clé privée à l'aide du KMIP**

Conditions de démarrage:

Présumer que l'IdP/SM-DP/MNO joue le rôle de générateur de la clé privée *ib.prk*. Avant que l'IdP/SM-DP/MNO puisse lancer la fonction IBExtract dans un KMS, les conditions suivantes doivent être remplies:

a) Un canal sécurisé a été établi entre l'IdP/SM-DP/MNO et le KMS.

b) L'IdP/SM-DP/MNO a suivi un processus d'authentification auprès du KMS et l'IdP/SM-DP/MNO authentifié a reçu l'autorisation de procéder à la demande IBExtract.

Procédure:

1) L'IdP/SM-DP/MNO prépare la charge utile des demandes et lance l'opération de signature afin d'envoyer le message de demande codé au KMS.

2) Le KMS vérifie la validité de la demande et s'assure que l'IdP/SM-DP/MNO est autorisé à lancer cette opération. Si l'une de ces conditions n'est pas satisfaite, le KMS envoie une réponse indiquant un échec. À défaut, le KMS exécute **IBExtract** avec *ib.msk*, *ib.pubparam* et les paramètres indiqués dans la demande.

3) Le KMS renvoie la réponse liée à l'exécution à l'IdP/SM-DP/MNO. Lorsque l'opération a été réalisée avec succès, le KMS transmet la clé privée générée à *ib.prk* sous la forme d'un IBPrivateKeyBlock, lequel constitue un choix (CHOICE) défini dans le protocole ASN.1 comme suit:

```
IBPrivateKeyBlock::= CHOICE{
        bb1PrivateKeyBlock      BB1PrivateKeyBlock,
        bfPrivateKeyBlock       BFPrivateKeyBlock,
        eccsiPrivateKeyBlock    ECCSIPrivateKeyBlock,
        skPrivateKeyBlock       SKPrivateKeyBlock,
```

sm9PrivateKeyBlock SM9PrivateKeyBlock

}

L'extension du KMIP visant à prendre en charge cette opération est décrite dans l'Appendice II.

Condition d'achèvement: L'IdP/SM-DP/MNO a obtenu la clé privée correspondant à l'identité associée à la demande.

- **Fourniture de clés et d'identités pour les eUICC**

Conditions de démarrage:

a) L'eUICC est enregistrée auprès du SM-SR et prête pour le téléchargement du profil.

b) Un profil non personnalisé a été créé par le SM-DP sur la base de la description de profil fournie par le MNO.

c) Le MNO présente une demande pour un certain nombre de profils d'eUICC.

d) Le profil non personnalisé a été validé sur le type d'eUICC cible à l'aide de la procédure de vérification des profils non personnalisés.

Procédure:

1) Le MNO fournit la commande de profils à un SM-DP sélectionné. Les détails du processus de commande des profils sont présentés dans le § 3.5.3 de la norme [b-GSMA SGP.02].

2) Un profil personnalisé est créé par le SM-DP sur la base des données transmises par le MNO. Le SM-DP doit notamment utiliser l'identité internationale d'abonnement mobile (IMSI) sélectionnée pour effectuer l'opération de signature avec le KMS, comme indiqué dans le paragraphe "Génération d'une clé privée à l'aide du KMIP", afin de générer une clé privée pour l'IMSI sélectionnée. La clé privée et les ibPublicParameters générés dans IBSysParams doivent être intégrés sur le profil en tant que clés.

3) Le profil cible est fourni sur l'eUICC par le MNO. Les détails du processus de téléchargement et d'installation du profil sont fournis dans le § 3.5.4 de la norme [b-GSMA SGP.02].

4) Le profil cible de l'eUICC est activé par l'intermédiaire du SM-SR ou par l'intermédiaire du SM-DP et du SM-SR. Les étapes de l'activation du profil sont détaillées dans les § 3.5.6 ou 3.5.7 de la norme [b-GSMA SGP.02].

Condition d'achèvement: Le profil cible est activé sur l'eUICC. Le profil précédemment activé a été désactivé. L'EIS est à jour.

- **Fourniture de clés et d'identités pour les dispositifs connectés à l'IoT non munis d'une eUICC**

Cas 1: Le SecM est en mesure d'établir une session TLS avec l'IdP.

Conditions de démarrage:

a) Le SecM a été enregistré auprès de l'AuC.

Procédure:

1) Le SecM établit une session TLS avec l'IdP et doit obtenir confirmation de la validité du certificat TLS de l'IdP.

2) Le SecM suit une procédure d'authentification Web avec l'IdP à l'aide du PROV.ID et du PROV.CRED.

3) L'IdP sélectionne une identité attribuée au dispositif ayant soumis la demande et réalise l'opération de signature avec le KMS, comme indiqué dans le paragraphe "Génération d'une clé privée à l'aide du KMIP", afin de générer une clé privée pour l'identité sélectionnée.

4) L'IdP envoie l'identité attribuée, la clé privée générée et les paramètres publics au SecM au moyen de la session TLS.

5) Le SecM stocke la clé privée de façon sécurisée et les paramètres publics sont protégés contre toute modification non autorisée.

Condition d'achèvement: La clé cible est fournie sur le SecM.

Le SecM et l'IdP suivent le protocole défini dans le paragraphe 5 et le Document [IETF RFC 5408] pour mener à bien la procédure de fourniture d'identités et de clés. Dans la réponse, la structure IBEPrivateKeyReply définie dans le Document [IETF RFC 5408] est remplacée par IBPrivateKeyReply.

```
IBPrivateKeyReply::= SEQUENCE SIZE (1..MAX) OF IBPrivateKey

IBPrivateKey::= SEQUENCE {
        pkgIdentity         IBIdentityInfo OPTIONAL,
        pkgAlgorithm        OBJECT IDENTIFIER,
        pkgKeyData          IBPrivateKeyBlock,   --défini par pkgAlgorithm
        pkgOptions          SEQUENCE SIZE (1..MAX) OF PKGOption,
        ibSysParams         IBSysParams OPTIONAL
}

PKGOption::=  SEQUENCE  {
        optionID        OBJECT IDENTIFIER,
        optionValue     OCTET STRING
}
```

Cas 2: Le SecM ne peut pas mettre en œuvre le protocole TLS.

Conditions de démarrage:

a) Le SecM a été enregistré auprès de l'AuC.

Procédure:

1) Le SecM génère la clé de chiffrement de la clé (KEK) et code la demande de fourniture de clé (IBKeyProvRequest). La demande comprend le KEK, l'identificateur d'approvisionnement (PROV.ID) et le justificatif d'authentification (PROV.CRED) qui sont chiffrés à l'aide de la clé publique de l'IdP identifiée par l'IdP.ID. Le résultat du chiffrement est codé en tant qu'EncryptedMsg. Il envoie la demande chiffrée à l'IdP dans le corps d'une requête HTTP POST.

2) L'idP décode le texte chiffré à l'aide de la clé de confidentialité indiquée par l'idP.ID dans la requête et vérifie l'actualisation de l'horodateur ou l'exactitude du compteur, ou les deux. Si la requête ne passe pas ces contrôles avec succès, l'idP envoie une réponse indiquant un échec. L'idP vérifie ensuite l'exactitude du PROV.ID et du PROV.CRED auprès de l'AuC. Si cette vérification échoue, l'IdP envoie une réponse indiquant un échec. L'IdP sélectionne une identité attribuée au dispositif ayant soumis la demande et réalise l'opération de signature avec le KMS, comme indiqué dans le paragraphe "Génération d'une clé privée à l'aide du KMIP", afin de générer une clé privée pour l'identité sélectionnée.

3) L'IdP chiffre la clé privée générée et, au besoin, l'identité ainsi que les paramètres publics, qui sont codés en tant qu'IBKeyProvisionData, la clé de chiffrement de la clé (KEK) utilisant l'algorithme (keyProtAlg) transmis dans la requête. Le texte chiffré est codé en tant

qu'EncryptedMsg. L'idP envoie la réponse chiffrée au SecM dans le corps de la réponse HTTP.

4) Le SecM déchiffre la réponse et obtient l'identité attribuée, la clé privée et les paramètres publics. Il stocke la clé privée de façon sécurisée et les paramètres publics sont protégés contre toute modification non autorisée.

Condition d'achèvement: La clé cible est fournie sur le SecM.

```
IBKeyProvisionRequest::= SEQUENCE{
        version     INTEGER { v1(1) },
        timer       Time OPTIONAL,
        counter     INTEGER       OPTIONAL,
        identity    OCTET STRING,
        credential  OCTET STRING,
        keyProtAlg  OBJECT IDENTIFIER,
        kek         OCTET STRING
}
Time::= CHOICE {
    utcTime         UTCTime,
    generalTime     GeneralizedTime
}
IBKeyProvisionResponse::= SEQUENCE SIZE(1..MAX) OF IBKeyProvisionData
IBKeyProvisionData::= SEQUENCE{
        identity        OCTET STRING OPTIONAL,
        ibSysParams     IBSysParams OPTIONAL,
        ibPrivateKey    IBPrivateKeyBlock
}
EncryptedMsg::= SEQUENCE {
        encryptionAlgorithm         EncryptionAlgorithmIdentifier,
        encryptedData               EncryptedData
}
EncryptionAlgorithmIdentifier::= AlgorithmIdentifier
EncryptedData::= OCTET STRING
```

C.5 Révocation de clés et d'identités

Si une identité doit être rejetée dans le système fondé sur la technologie IBC pour diverses raisons, par exemple, car le propriétaire de l'identité s'est désabonné du service ou que la clé privée correspondante a été compromise, l'identité doit être révoquée et il peut être nécessaire de détruire la clé privée correspondante pour des raisons de sécurité. Si une identité est révoquée, son statut doit être changé en "revoked". Si une entrée demande le statut d'une identité révoquée, l'IdP/SM-DP/MNO doit renvoyer la valeur correcte telle qu'elle est définie dans l'OISP. Pour vérifier le statut d'identités plus facilement, une entité peut récupérer régulièrement l'IRL de l'IdP/SM-DP/MNO, la conserver au niveau local, et vérifier à l'aide de l'IRL la plus récente si une identité a été révoquée sans avoir à

demander le statut de chaque identité en ligne. Pour l'eUICC, le processus de destruction de clé privée peut être mené à bien en désactivant tout d'abord, puis en supprimant le profil de l'eUICC.

- **Révocation de clés et d'identités pour les dispositifs à eUICC**

Conditions de démarrage:

a) Le profil cible est activé sur l'eUICC.

Procédure:

1) Le MNO commence à désactiver le profil par l'intermédiaire du processus du SM-DP. Les détails du processus de désactivation des profils sont présentés dans le § 3.5.8 de la norme [b-GSMA SGP.02]. Le SM-DP doit attribuer à l'identité le statut "revoked".

2) Le MNO démarre le processus de suppression du profil. Les étapes de la suppression de l'ISD-P sont détaillées dans le § 3.5.10 de la norme [b-GSMA SGP.02]. Le SM-DP attribue à l'identité le statut "revoked" et, une fois le processus de suppression de l'ISD-P mené à bien, le statut "supprimée" est également attribué à l'identité. Lorsqu'une entité demande le statut d'une identité, le SM-DP répond de façon appropriée conformément à l'enregistrement de statut. Le SM-DP publie périodiquement une liste de statut des identités indiquant les identités ayant été révoquées au cours de la période concernée.

Condition d'achèvement: Le profil cible est désactivé et supprimé de l'eUICC.

- **Révocation de clés et d'identités pour les dispositifs connectés à l'IoT non munis d'une eUICC**

Si une identité est révoquée, l'IdP lui attribue le statut "revoked". Lorsqu'une entité demande le statut d'une identité, l'IdP répond de façon appropriée conformément à l'enregistrement de statut. L'IdP publie périodiquement une liste de statut des identités indiquant les identités ayant été révoquées au cours de la période concernée.

Le processus de déclenchement des révocations et la maintenance du statut des identités ne relèvent pas du champ d'application de la présente Recommandation.

- **Protocole de statut de l'identité en ligne**

Au vu du nombre important de dispositifs connectés à l'IoT se connectant à un opérateur de télécommunications, il peut être nécessaire pour un SM-DP, un IdP ou un dispositif connecté à l'IoT d'obtenir des informations opportunes au sujet du statut de révocation de l'identité d'un dispositif connecté à l'IoT. Un OISP est indiqué dans la présente Recommandation afin d'activer le SM-DP, l'IdP ou un dispositif connecté à l'IoT, et ce, en vue de déterminer le statut actuel d'une identité au moyen de requêtes en ligne. Un client OISP émet une demande de statut auprès d'un répondeur OISP et suspend l'acceptation de l'identité dont il est question jusqu'à l'obtention de la réponse du répondeur. L'OISP présente des similitudes avec le protocole de statut du certificat en ligne (OCSP) [IETF RFC 6960].

Une requête OISP contient les données suivantes:

```
OISPRequest::= SEQUENCE {
        version         INTEGER { v1(1) },
        identity        IBIdentityInfoSet
}
```

– "version" désigne la version du protocole, qui, dans le cas du présent document, est v1(1).

– "identity" désigne la requête OISP.

```
IBIdentityInfoSet::= SEQUENCE SIZE(1..MAX) OF IBIdentityInfo

IBIdentityInfo::= SEQUENCE {
```

```
    domainName          IA5String OPTIONAL,
    domainSerial        INTEGER OPTIONAL,
    identityType        OBJECT IDENTIFIER OPTIONAL,
    identityData        OCTET STRING
}
```

– domainName est FACULTATIF et IA5String représente l'URI [b-URI] ou l'IRI [b-IRI].

– domainSerial est FACULTATIF et comprend un ENTIER qui définit un jeu unique de paramètres publics d'IBC dans l'éventualité où plusieurs jeux de paramètres seraient utilisés par un même domaine.

– identityType est FACULTATIF et contient un IDENTIFICATEUR D'OBJET définissant le format dans lequel le champ identityData est codé. En cas d'absence de ce champ, un type d'identité par défaut est utilisé.

– identityData désigne les données de l'identité cible.

Dès qu'il reçoit une demande, le répondeur OISP vérifie si le message est bien formé et si la requête contient l'information requise par le répondeur. Si cette vérification échoue, le répondeur OISP génère un message d'erreur. Dans le cas contraire, il envoie une réponse définitive conformément au statut des identités requises dans la demande.

```
OISPResponse::= SEQUENCE {
    responseStatus      OISPResponseStatus,
    responseData        OISPResponseData OPTIONAL
}
```

– responseStatus désigne le statut du traitement de la demande antérieure.

– responseData est FACULTATIF et comprend les données de réponse associées à la demande. Si la valeur de responseStatus fait partie des conditions d'erreur, le champ responseData n'est pas défini.

```
OISPResponseStatus::= ENUMERATED {
    successful          (0),  -- La réponse présente des confirmations valides.
    malformedRequest    (1),  -- Demande de confirmation incorrecte
    internalError       (2),  -- Erreur interne chez l'émetteur
    tryLater            (3),  -- Veuillez réessayer plus tard.
                        --      (4) N'est pas utilisé.
    unauthorized        (5)   -- Demande non autorisée
}
OISPResponseData::= SEQUENCE {
    version             INTEGER { v1(1) },
    producedAt          GeneralizedTime,
    hashAlgorithm       AlgorithmIdentifier OPTIONAL,
    tbsIdStatus         SEQUENCE OF SingleIdStatus,
    signatureAlgorithm  AlgorithmIdentifier OPTIONAL,
    signature           BIT STRING OPTIONAL,
    certs               [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL
```

}

– La version DOIT être v1(1) pour cette version de la syntaxe de réponse de base.

– producedAt désigne la date à laquelle le répondeur OISP a signé cette réponse.

– hashAlgorithm désigne un algorithme de hachage visant à générer idHash dans tbsIdStatus lorsque ce champ existe. Ce champ est facultatif et la valeur par défaut est OBJECT IDENTIFIER (identificateur d'objet) pour SHA256 sans paramètres.

– tbsldStatus désigne les réponses associées à chacune des identités d'une demande.

– signatureAlgorithm est FACULTATIF et comprend l'algorithme ayant été utilisé pour signer la réponse.

– La signature est calculée en fonction du résultat associé aux DER de l'ASN.1 du champ producedAt à tbsIdStatus à l'aide de l'algorithme de signature indiqué. Ce champ est FACULTATIF et peut ne pas être défini si le client OISP dispose d'autres moyens pour garantir l'authenticité de la réponse. Par exemple, la réponse est transmise par l'intermédiaire d'un canal sécurisé TLS établi entre le client et le répondeur.

– certs est FACULTATIF et désigne le certificat qui permet au client OISP de vérifier la signature du répondeur. La structure du certificat est décrite dans le Document [IETF RFC 5280].

```
SingleIdStatus::= SEQUENCE {
    idHash          OCTET STRING OPTIONAL,
    identityID      IBIdentityInfo OPTIONAL,
    identityStatus  IdentityStatus,
}
```

– idHash est FACULTATIF et comprend le hachage de l'identité associée à la demande. Si identityID est trop long, idHash peut être utilisé pour représenter l'identité requise. identityID est FACULTATIF et inclut le champ IBIdentityInfo de l'identité cible dans la demande.

– identityStatus désigne le statut de l'identité dans la demande précédente.

```
IdentityStatus::= CHOICE {
    good                [0]   IMPLICIT NULL,
    revoked             [1]   IMPLICIT RevokedInfo,
    unknown             [2]   IMPLICIT UnknownInfo,
    updated             [3]   IMPLICIT IBIdentityInfo,
    revokedAndDeleted   [4]   IMPLICIT RevokedInfo
}
UnknownInfo::= NULL
```

– L'état "good" indique une réponse positive à la demande d'information sur le statut.

– L'état "revoked" indique que l'identité a été révoquée, que ce soit de façon temporaire ou permanente, et la valeur consiste en l'information liée à la révocation.

– L'état "unknown" indique que le répondeur ne connaît pas le certificat demandé.

– L'état "updated" indique que l'identité a été mise à jour et que la valeur est une identité nouvellement attribuée à l'identité objet de la requête.

– L'état "revokedAndDeleted" indique que l'identité a été révoquée et que la clé privée a été supprimée du dispositif distant.

```
RevokedInfo::= SEQUENCE {
```

```
    revocationTime          GeneralizedTime,
    revocationReason        [0] EXPLICIT IRLReason OPTIONAL
}
IRLReason::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        pkgCompromise           (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        identityHold            (6),
                                -- La valeur 7 n'est pas utilisée.
        removeFromIRL           (8),
        privilegeWithdrawn      (9)
}
```

- **Liste de révocation d'identités**

En plus d'avoir recours à l'OISP pour répondre aux demandes relatives au statut des identités, une entité telle que l'IdP ou le SM-DP peut publier une IRL, à savoir une liste complète des identités révoquées sur une période régulière. Pour accélérer le processus de vérification du statut de l'identité, une entité de vérification des statuts possédant une importante capacité de stockage peut demander à obtenir l'IRL, puis la stocker localement. L'entité de vérification peut établir si une identité est acceptable pour des opérations données, telles que l'autorisation d'accès réseau, sur la base de l'IRL. Si cette identité ne figure pas dans l'IRL, il est présumé que l'identité est valide. Pour renforcer l'efficacité du système, l'IdP/SM-DP/MNO peut simplement publier les identités nouvellement révoquées, depuis une date précise. C'est ce qu'on appelle une IRL delta. Une IRL delta contient des informations relatives aux identités révoquées depuis la publication d'une IRL complète. La consultation des IRL delta peut réduire de façon considérable la communication générale et le temps de traitement des IRL. Une IRL est comparable à une liste de révocation de certificats (CRL) [IETF RFC 5280].

L'IRL est définie comme suit:

```
IdentityRevocationList::= SEQUENCE {
    tbsIdentityList         TBSIdentityRevocationList  ,
    signatureAlgorithm      AlgorithmIdentifier OPTIONAL,
    signatureValue          BIT STRING OPTIONAL
}
```

– tbsIdentityList désigne la liste des identités révoquées accompagnée de renseignements complémentaires, tels que la date de la révocation.

– signatureAlgorithm désigne l'algorithme que l'émetteur de l'IRL a utilisé pour signer la liste. Ce champ est facultatif et est absent en cas d'inexistence de signatureValue.

– signatureValue désigne la valeur de la signature générée par l'émetteur sur tbsIdentityList. Ce champ est facultatif et est absent si le client invoqué dispose d'autres moyens pour garantir l'authenticité de la liste extraite.

```
TBSIdentityRevocationList ::=  SEQUENCE  {
      version                 INTEGER { v1(1) },
      issuer                  Name,
      irlNumber               INTEGER OPTIONAL,
      deltaList               BOOLEAN OPTIONAL,
      thisUpdate              Time,
      nextUpdate              Time OPTIONAL,
      domainName              IA5String OPTIONAL,
      domainSerial            INTEGER OPTIONAL,
      revokedIdentities       SEQUENCE OF SEQUENCE  {
          identity            IBIdentityInfo,
          revocationDate      Time,
          irlEntryExtensions  Extensions OPTIONAL
      } OPTIONAL,
      irlExtensions           [0] EXPLICIT Extensions OPTIONAL
}
Name::= CHOICE {--imported from [IETF RFC 5280]
      rdnSequence    RDNSequence
}
RDNSequence::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName::= SET SIZE (1..MAX) OF AttributeTypeAndValue
AttributeTypeAndValue::= SEQUENCE {
      type    AttributeType,
      value   AttributeValue
}
AttributeType::= OBJECT IDENTIFIER
AttributeValue::= ANY -- DEFINED BY AttributeType
Extensions ::=  SEQUENCE SIZE (1..MAX) OF Extension
Extension::=  SEQUENCE  {--imported from [IETF RFC 5280]
      extnID      OBJECT IDENTIFIER,
      critical    BOOLEAN DEFAULT FALSE,
      extnValue   OCTET STRING
                  -- comprend le codage DER d'une valeur d'un type défini par ASN.1
                  -- correspondant au type d'extension identifié
                  -- par extnID
}
```

– "version" désigne la version de la structure de l'IRL.

– "issuer" désigne le nom de l'entité émettrice de l'IRL.
– "irlNumber" désigne le numéro de l'émetteur de l'IRL actuelle. Il commence à 0. Pour chaque publication d'IRL complète, ce nombre augmente de 1. Il est facultatif.
– "deltaList" indique si l'IRL actuelle est une IRL delta. La liste ne contient que des informations relatives aux identités révoquées depuis la publication d'une IRL complète indexée par irlNumber.
– "thisUpdate" indique la date à laquelle cette IRL a été générée.
– "nextUpdate" désigne la date à laquelle la prochaine IRL devra être générée. Il est facultatif.
– "domainName" désigne le domaine d'identité de l'IBC.
– "domainSerial" désigne le numéro du domaine d'identité de l'IBC.
– "revokedIdentities" désigne le jeu d'identités révoquées.
– "identity" désigne les données de l'identité révoquée.
– "revocationDate" désigne la date à laquelle l'identité a été révoquée.
– "irlEntryExtensions" désigne les extensions possibles de revokedIdentity. Actuellement, aucune extension n'est définie.
– "irlExtensions" désigne les possibles extensions de l'IRL. Actuellement, aucune extension n'est définie.

Annexe D

Authentification

(Cette Annexe fait partie intégrante de la présente Recommandation.)

Dans la présente Annexe, quatre protocoles d'authentification font l'objet d'une extension aux fins de la prise en charge de l'IBC.

D.1 Protocole de transport secret à passe unique

Ce protocole correspond au mécanisme 2 de transport de clés secrètes décrit dans la norme [ISO/CEI 11770-3]. Il permet de transférer une clé secrète, générée, chiffrée et signée par une entité A, de l'entité A à une entité B au moyen d'une authentification de clé explicite de l'entité A vers l'entité B et d'une authentification de clé implicite de l'entité B vers l'entité A. L'authentification de clé explicite de l'entité A vers l'entité B est réalisée au moyen de la signature par l'entité A du secret chiffré et d'un paramètre variable dans le temps (TVP). L'authentification de clé implicite de l'entité B vers l'entité A est réalisée grâce au chiffrement du secret à l'aide de l'identificateur de B, qui implique que seule l'entité B peut récupérer le secret. Voir la Figure D.1.

Les exigences ci-dessous doivent être satisfaites aux fins de la mise en œuvre du protocole.

– L'entité A possède une clé privée de signature *A.ib.prk* correspondant à son identificateur et aux paramètres publics connexes *A.ib.pubparam*.

– L'entité B possède une clé privée de déchiffrement *B.ib.prk* correspondant à son identificateur et aux paramètres publics connexes *B.ib.pubparam*.

– L'entité A accès à une copie authentifiée des paramètres publics de chiffrement de l'entité B *B.ib.pubparam*, ainsi qu'à l'identificateur de B.

– L'entité B a accès à une copie authentifiée des paramètres publics de signature de l'entité A *A.ib.pubparam*, ainsi qu'à l'identificateur de A.

– Le TVP facultatif est soit un horodateur, soit un numéro de séquence. Dans le cas où des horodateurs sont utilisés, les entités A et B doivent veiller au maintien de la synchronisation de leurs horloges ou faire appel à un horodateur tiers de confiance.

– A et B peuvent partager les mêmes paramètres publics, c'est-à-dire *A.ib.pubparam* = *B.ib.pubparam*.

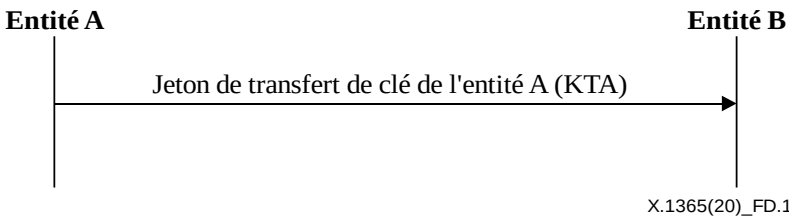

Figure D.1 – Protocole de transport secret à passe unique

1) L'entité A génère un secret aléatoire *K* de la longueur requise.

2) L'entité A génère *BE*=**IBEnc**(*B.ib.pubparam, ID_B,* [*ID_A*]||*K*||*Text1*). Text1 peut être vide et *ID_A* est facultatif si l'entité B dispose d'autres moyens d'obtenir l'identificateur de l'entité A.

3) L'entité A génère *S*=**IBSign**(*A.ib.pubparam, ID_A, A.ib.prk,* [*ID_B*]||TVP||BE||*Text2*). Text2 peut être vide et *ID_B* est facultatif si l'entité B connaît l'identificateur utilisé *ID_B* pour le chiffrement.

4) L'entité A génère le jeton KTA=[*ID_B*]||TVP||BE|| *Text2*|| S||*Text3*.

5) Lorsque le TVP est un horodateur, l'entité B vérifie si le TVP respecte la différence de temps autorisée. Si tel n'est pas le cas, l'entité B rejette le jeton.

6) Si l'entité B peut obtenir *ID_A* par d'autres moyens et que le TVP est un numéro de séquence, l'entité B vérifie si le numéro de séquence est plus grand que celui conservé par l'entité B. Si tel n'est pas le cas, l'entité B rejette le jeton.

7) Si l'entité B peut obtenir *ID_A* par d'autres moyens, l'entité B vérifie la signature *S* dans KTA au moyen d'**IBVerify**(*A.ib.pubparam, ID_A,* [*ID_B*]||TVP||BE||*Text2,* S). Si la signature n'est pas valide, l'entité B rejette le jeton.

8) L'entité B déchiffre *BE* au moyen de *[ID_A]||K||Text1*=**IBDec**(*B.ib.pubparam, ID_B, B.ib.prk,* BE).

9) Si l'entité B ne peut obtenir *ID_A* qu'à la suite de l'étape 8, l'entité B vérifie l'actualisation du TVP, dans le cas où le TVP est un numéro de séquence. Si le TVP n'a pas été actualisé, l'entité B rejette le jeton. L'entité B vérifie ensuite la signature *S*. Si la signature n'est pas valide, l'entité B rejette le jeton.

10) Lorsque tous les contrôles et vérifications ont été subis avec succès, l'entité A et l'entité B utilisent *K* pour protéger les messages suivants. Les deux entités peuvent utiliser une fonction de dérivation de la clé (KDF) [b-IEEE 1363] afin de générer des clés pour le chiffrement et l'authentification des messages.

NOTE 1 – Le protocole peut être converti en un protocole d'authentification des entités unilatérale en supprimant BE du message signé par l'entité A et KTA. Cette modification devient le modèle d'authentification d'entités en une seule passe décrit dans la norme [b-ISO/CEI 9798-3].

NOTE 2 – Le protocole peut être converti en un protocole d'authentification d'entités bilatéral en exigeant que l'entité B renvoie *K* à l'entité A. L'entité B est authentifiée par l'entité A en démontrant qu'elle peut récupérer *K*, ce qui implique qu'elle détienne la clé privée *B.ib.prk*.

NOTE 3 – Les algorithmes de signature et de chiffrement (signcryption) fondés sur l'identité, tels que l'algorithme BLMQ [b-Barreto], et l'algorithme Chen-Malone-Lee [b-Chen], peuvent être utilisés pour une plus grande efficacité.

D.2 TLS-IBS

Le paragraphe ci-après décrit un autre protocole d'authentification connu sous le nom de protocole TLS-IBS. On suppose que tant le serveur que le dispositif connecté à l'IoT sont dotés de justificatifs d'identité, lesquels sont constitués d'une identité, d'une clé privée à utiliser aux fins de signature et de paramètres publics associés au KMS (par exemple, la KPAK, telle que définie dans le Document [IETF RFC 6507] en tant que paramètre de calcul). Les définitions de la structure des paramètres publics du KMS pour les algorithmes pris en charge peuvent être consultées à l'Annexe B.

Le TLS-IBS est développé sur la base du Document [IETF RFC 7250]. Traditionnellement, le client et le serveur TLS échangent des clés publiques validées par des certificats pour infrastructures à clé publique (PKI). Cette pratique associée à l'utilisation de certificats PKI est considérée complexe et susceptible de représenter un danger pour la sécurité. Afin de simplifier l'échange de certificats, le recours à une clé publique brute dans TLS est spécifié dans le Document [IETF RFC 7250]. En d'autres termes, en lieu et place de la transmission d'un certificat complet dans les messages TLS, seules les clés publiques sont échangées entre le client et le serveur. Toutefois, on suppose l'existence d'un mécanisme hors bande aux fins de la liaison entre la clé publique et l'identité. Pour les réseaux IoT, le TLS associé à une clé publique brute est particulièrement attractif. Cependant, la liaison entre les identités et les clés publiques peut, dans ce cas de figure, se révéler difficile. La maintenance d'une importante table de mise en correspondance des identités et des clés publiques au niveau du serveur génère des frais supplémentaires. À titre d'exemple, les dispositifs doivent se préenregistrer auprès du serveur. En vue de simplifier la liaison entre la clé publique et l'entité présentant la clé publique, il pourrait être préférable de recourir à l'IBC, par exemple, à la clé

publique ECCSI spécifiée dans le Document [IETF RFC 6507], aux fins d'authentification. À la différence des certificats et clés publiques brutes décrits dans la Recommandation UIT-T X.509, une clé publique utilisée dans un système à IBC prend la forme de l'identité de l'entité. Cela contribue à éliminer la nécessité de la liaison entre une clé publique et l'entité présentant la clé publique.

Lorsque l'IBS est utilisé à titre de clé publique brute pour TLS, la signature et les algorithmes de hachage sont négociés lors de l'établissement de la connexion. L'établissement de la connexion entre le client et le serveur TLS suit les procédures définies dans le Document [IETF RFC 7250] et TLS 1.3 [IETF RFC 8446] mais avec la prise en charge des algorithmes de l'IBS en tant que schémas de signature numérique.

Le protocole TLS-IBS, développé sur la base du Document [IETF RFC 7250] et TLS 1.3 et ayant pour algorithmes de signature ECCSI [IETF RFC 6507], IBS1 (Hess-IBS), IBS1 (Cha-Cheon-IBS) et SM9-IBS [ISO/CEI 14888-3], est décrit comme suit.

1) Le dispositif connecté à l'IoT envoie au serveur un message ClientHello contenant les extensions key_share, signature_algorithms, server_certificate_type et client_certificate_type, et indiquant qu'il prend en charge la clé publique brute et les algorithmes de l'IBS.

2) Le serveur envoie au dispositif connecté à l'IoT un message ServerHello contenant les extensions key_share, server_certificate_type, client_certificate_type, Certificate, CertificateRequest, CertificateVerify et Finished, et indiquant que la clé publique brute est prise en charge, et inclut son identité (ServerID) et les paramètres KMS (OID, KMS parameters) dans la partie dédiée au certificat. Les structures de données relatives aux paramètres KMS sont définies dans le paragraphe D.2.3. Le message CertificateVerify comprend une signature générée avec la clé privée détenue par le serveur.

3) Après avoir vérifié l'identité et la signature du serveur, le dispositif connecté à l'IoT envoie sa clé publique brute au serveur, à travers les messages Certificate, CertificateVerify et Finished. Le dispositif connecté à l'IoT indique son identité (ClientID) et les paramètres KMS (OID, paramètres KMS) dans la zone de certificat, ce qui correspond à la clé publique brute du client. Les structures de données relatives aux paramètres KMS sont définies dans le paragraphe D.2.3. Une signature générée à l'aide de la clé privée du client est incluse.

4) Les étapes restantes sont les mêmes que celles exposées dans TLS 1.3 dans le Document [IETF RFC 8446].

Voir la Figure D.2.

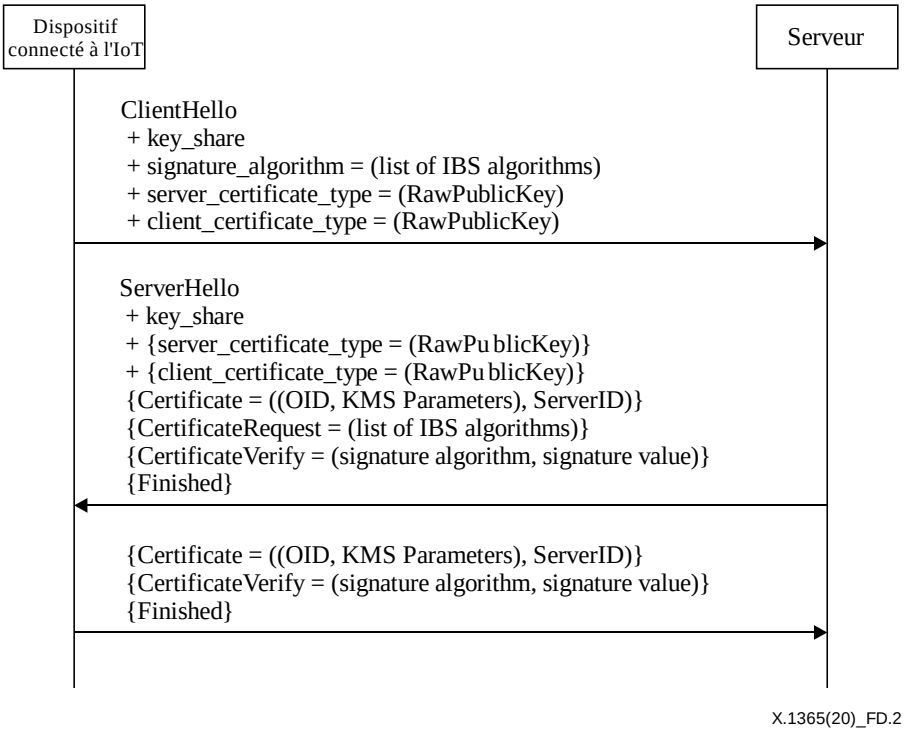

Figure D.2 – TLS-IBS

D.2.1 ClientHello

Le format du message ClientHello est identique à celui décrit dans TLS 1.3 [IETF RFC 8446]. Toutefois, les valeurs de l'algorithme de signature doivent être étendues pour l'IBS.

Le message ClientHello informe le serveur des types de certificats ou clés publiques brutes pris en charge par le client et des types de certificats que le client s'attend à recevoir du serveur. Le message ClientHello contient les algorithmes de l'IBS souhaités sur la base du classement par ordre de préférence fourni par le client. TLS 1.3 offre une définition d'une structure de données nommée SignatureScheme pour les algorithmes de signature. Aux fins de sa prise en charge, l'algorithme de l'IBS doit être étendu comme suit:

```
enum {
...
/* IBS signature algorithm */
eccsi_sha256 (0x0704),
ibs1_sha256(0x0705)
ibs2_sha256(0x0706)
sm9_ibs_sm3(0x0707)
/* Reserved Code Points */
private_use (0xFE00..0xFFFF),
(0xFFFF)
} SignatureScheme;
```

Des informations détaillées concernant les points de code pour les algorithmes de signature étendus sont fournies dans le registre [b-IANA TLS REG] pour TLS.

D.2.2 ServerHello

Le format du message ServerHello est identique à celui décrit dans TLS 1.3 [IETF RFC 8446]. La structure SignatureScheme est étendue de la même façon que dans le message Client_Hello.

D.2.3 Certificat de serveur

En ce qui concerne le certificat de serveur, une structure de certificat est définie en tant que RawPublicKey dans le Document [b-IETF RFC 7250]. Une structure de données nommée subjectPublicKeyInfo est utilisée en vue de spécifier la clé publique brute et son algorithme de chiffrement, tel que dans le Document [IETF RFC 7250]. Deux champs, à savoir les champs algorithme et paramètres, sont définis dans la structure subjectPublicKeyInfo. Le champ algorithme indique l'algorithme de chiffrement utilisé avec une clé publique brute, lequel est représenté par des OID. Le champ paramètres fournit les paramètres requis en lien avec l'algorithme. L'identité du serveur doit figurer dans la partie subjectPublicKey.

NOTE – L'identité doit suivre le format défini dans l'Appendice I.

```
subjectPublicKeyInfo::= SEQUENCE {
        algorithm           AlgorithmIdentifier,
        subjectPublicKey    BIT STRING
}
AlgorithmIdentifier::= SEQUENCE {
        algorithm           OBJECT IDENTIFIER,
        parameters          ANY DEFINED BY algorithm OPTIONAL
}
```

Lors de l'utilisation d'un algorithme de l'IBS, une identité est utilisée à titre de clé publique brute, qui peut être convertie en chaîne d'octets (OCTET STRING). Ainsi, le certificat et la structure subjectPublicKey peuvent être réutilisés sans modifications.

Le champ algorithme dans la structure AlgorithmIdentifier est l'identificateur d'objet de l'algorithme d'IBS utilisé. Outre cela, il est nécessaire de communiquer à l'entité homologue le jeu de paramètres publics utilisés par le signataire. Ces informations peuvent être transportées avec la charge utile du champ paramètres dans AlgorithmIdentifier. Les structures de paramètres publics correspondant aux algorithmes cités ci-dessus sont, respectivement, ECCSIPublicParameters, BFPublicParameters, BFPublicParameters et SM9PublicParameters, telles que définies à l'Annexe B.

Il est nécessaire de définir une structure de données pour la valeur de signature aux fins de la prise en charge des algorithmes de l'IBS sur le protocole TLS.

- Une structure de données pour ECCSI est définie comme suit (sur la base du Document [IETF RFC 6507]):

```
ECCSI-Sig-Value::= SEQUENCE {
r INTEGER,
s INTEGER,
pvt  OCTET STRING
}
```

où pvt (tel que PVT est défini dans [IETF RFC 6507]) est codé de la façon suivante: 0x04 || axe des abscisses de [v]G || axe des ordonnées de [v]G.

- Une structure de données pour IBS1 est définie comme suit:

  IBS1-Sig-Value::= SEQUENCE {
  r INTEGER,
  s ECPoint
  }

ECPoint::= OCTET STRING tel que défini dans [IETF RFC 5480]

- Une structure de données pour IBS2 est définie comme suit:

  IBS2-Sig-Value::= SEQUENCE {
  r ECPoint,
  s ECPoint
  }

- Une structure de données pour SM9-IBS est définie comme suit:

  SM9-Sig-Value::= SEQUENCE {
  r INTEGER,
  s ECPoint
  }

Pour utiliser un algorithme de signature avec le protocole TLS, un OID pour l'algorithme de signature doit être fourni. Le Tableau D.1 indique les informations essentielles requises pour une utilisation des algorithmes de signature IBS avec le protocole TLS.

Tableau D.1 – Algorithmes de signature fondée sur l'identité

Type de clé	Document	OID
ISO/IEC 14888-3 ibs-1	ISO/IEC 14888-3: IBS-1 mechanism	1.0.14888.3.0.7
ISO/IEC 14888-3 ibs-2	ISO/IEC 14888-3: IBS-2 mechanism	1.0.14888.3.0.8
SM9-IBS	ISO/IEC 14888-3: Chinese IBS mechanism	1.2.156.10197.1.302.1
Signatures à courbe elliptique sans certificat pour le chiffrement fondé sur l'identité (ECCSI)	Paragraphe 5.2 du Document [IETF RFC 6507]	1.3.6.1.5.5.7.6.29

D.2.4 Certificat client

Afin de prendre en charge l'IBS, le certificat client est étendu de la même façon que le certificat de serveur.

D.3 EAP-TLS-IBS

Dans le paragraphe suivant, le protocole d'authentification EAP-TLS est étendu afin de prendre en charge l'IBS. Tant le réseau que l'ÉU sont dotés de justificatifs d'identité, lesquels sont constitués d'une identité, d'une clé privée à utiliser aux fins de signature et de paramètres publics associés au KMS (par exemple, la KPAK, telle que définie dans le Document [IETF RFC 6507]). Voir la Figure D.3.

L'EAP-TLS est modifié comme suit:

1) Identique à EAP-TLS.
2) Suite à la réception de la réponse EAP contenant l'identité de l'ÉU, ID_UE.
3) L'AU envoie l'ID_UE à la RSF aux fins de validation.
4) La RSF valide l'ID_UE en fonction de la liste de révocation enregistrée.

5) La RSF renvoie le résultat de la validation à l'AU.

6) Si l'ID_UE est valide, l'AU envoie le message de début de l'EAP-TLS à l'ÉU.

7-9) Identiques à ceux décrits dans le TLS-IBS susmentionné.

10) Succès EAP.

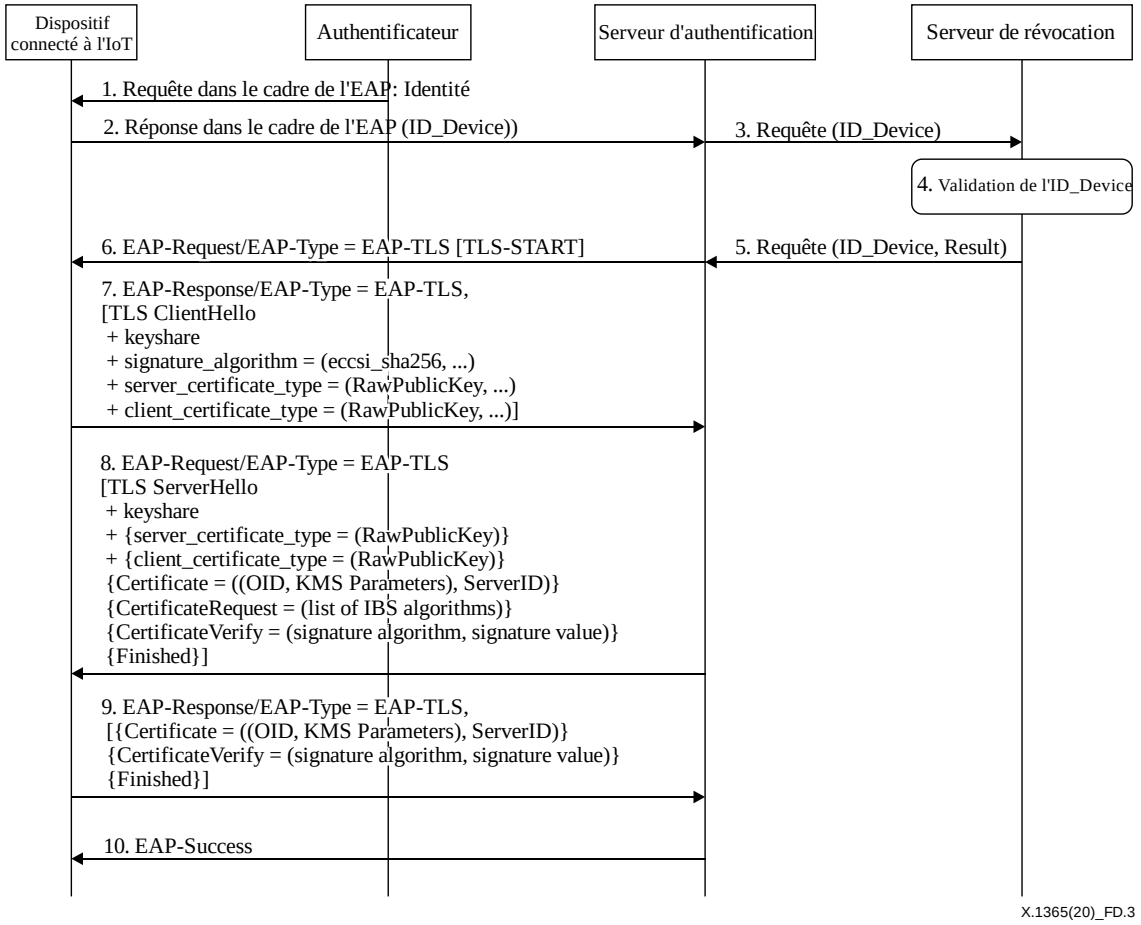

Figure D.3 – EAP-TLS-IBS

D.3.1 Demande EAP

Le format du message de demande EAP est identique à celui décrit dans le Document [IETF RFC 5216].

D.3.2 Réponse EAP

Le format du message de réponse EAP est identique à celui décrit dans le Document [IETF RFC 5216].

D.3.3 ClientHello

Le format du message ClientHello est identique à celui décrit dans le paragraphe D.2.1.

D.3.4 ServerHello

Le format du message ServerHello est identique à celui décrit dans le paragraphe D.2.2.

D.3.5 Certificat de serveur

Le format du certificat de serveur est identique à celui décrit dans le paragraphe D.2.3.

D.3.6 Certificat client

Le format du certificat client est identique à celui décrit dans le paragraphe D.2.4.

D.4 EAP-PSK-ECCSI

Dans le paragraphe qui suit, l'EAP-PSK est étendu en vue de la prise en charge de l'un des algorithmes de l'IBS, à savoir ECCSI, aux fins d'authentification. Tant l'ÉU que l'AU sont dotés de justificatifs d'identité, lesquels sont constitués d'une identité, d'une clé de signature secrète (SSK), un jeton de vérification public (PVT) et une clé d'authentification publique du KMS (KPAK, telle que définie dans le Document [IETF RFC 6507] en tant que paramètre de calcul).

Avec les justificatifs d'identité fournis, l'ÉU et l'AU peuvent dériver les clés symétriques fondées sur Diffie-Hellman statique en échangeant les informations d'identité et le PVT, puis utiliser la SSK détenue par chaque entité. Par exemple, un ÉU peut dériver une clé après avoir reçu l'identité de l'AU et son PVT, désignés respectivement par ID_AU et PVT_AU, comme suit:

K_UE = [SSK_UE](KPAK+[hash(G || KPAK || ID_AU || PVT_AU)]PVT_AU))

où G est un point de génération sur la courbe elliptique utilisé par le KMS pour générer des clés pour l'ÉU et les réseaux. Il est fourni à l'ÉU et à l'AU par le KMS avec la SSK, le PVT et la KPAK, pour ne citer que quelques éléments. Le recours à la fonction de hachage peut avoir lieu conformément à l'Annexe A du Document [IETF RFC 6507].

De la même manière, l'AU peut également dériver K_AU après avoir reçu l'identité et le PVT de l'ÉU, comme indiqué ci-dessous:

K_AU = [SSK_AU](KPAK+[hash(G || KPAK || ID_UE || PVT_UE)]PVT_UE))

Il peut être démontré que K_UE équivaut en réalité à K_AU.

À l'aide des propriétés ci-dessus, l'EAP-PSK peut être utilisé en vue d'une authentification mutuelle, comme suit.

1) L'ÉU envoie une demande d'attachement à l'AU et indique que l'EAP-PSK doit être utilisé aux fins de l'authentification mutuelle.
2) L'AU vérifie le type d'authentification et décide d'une méthode d'authentification.
3) L'AU envoie à l'ÉU le premier message concernant l'EAP-PSK avec un champ d'identité contenant l'ID_AU et le PVT_AU, ainsi qu'un nombre aléatoire RAND_S, comme l'exige l'EAP-PSK.
4) L'ÉU dérive une clé symétrique telle que K = [SSK_UE](KPAK+[hash(G || KPAK || ID_AU || PVT_AU)]PVT_AU)). L'ÉU génère un nombre aléatoire RAND_P, puis dérive K' = KDF(K, RAND_P, RAND_S). L'ÉU dérive une clé d'authentification (AK) ainsi qu'une clé de dérivation de la clé (KDK) sur la base du Document [b-IETF RFC4764] pour l'EAP-PSK.
5) L'ÉU envoie le deuxième message concernant l'EAP-PSK à l'AU, lequel contient RAND_S, RAND_P, a MAC_P (MAC_P=CMAC-AES-128(AK, ID_P||ID_S||RAND_S||RAND_P)) aux fins d'authentification, ainsi qu'un champ d'identité comprenant ID_UE et PVT_UE.
6) L'AU envoie l'ID_UE à la RSF aux fins de validation.
7) La RSF valide l'ID_UE en fonction de sa liste de révocation.
8) La RSF renvoie les résultats de la validation à l'AU.
9) Si l'ID est valide, l'AU dérive une clé symétrique telle que K = [SSK_AU](KPAK+[hash(G || KPAK || ID_UE || PVT_UE)]PVT_UE)). L'AU dérive ensuite K' = KDF(K, RAND_P, RAND_S). L'AU dérive une AK et une KDK sur la base du Document [IETF RFC 4764] pour l'EAP-PSK. L'AU authentifie l'ÉU grâce au MAC_P reçu dans le message. L'AU dérive ensuite une clé de session en fonction de RAND_P et de la KDK.

10) L'AU envoie à l'ÉU le troisième message concernant l'EAP-PSK contenant un MAC_S (MAC_S=CMAC-AES-128(AK, ID_S||RAND_P)) aux fins d'authentification, ainsi que d'autres champs requis par l'EAP-PSK.

11) L'ÉU authentifie l'AU à l'aide du MAC_S reçu et dérive une clé de session à l'aide de RAND_P et de la KDK dérivés précédemment.

12) L'ÉU envoie le dernier message concernant l'EAP-PSK à l'AU afin d'achever la procédure d'authentification de l'EAP-PSK.

Voir la Figure D.4.

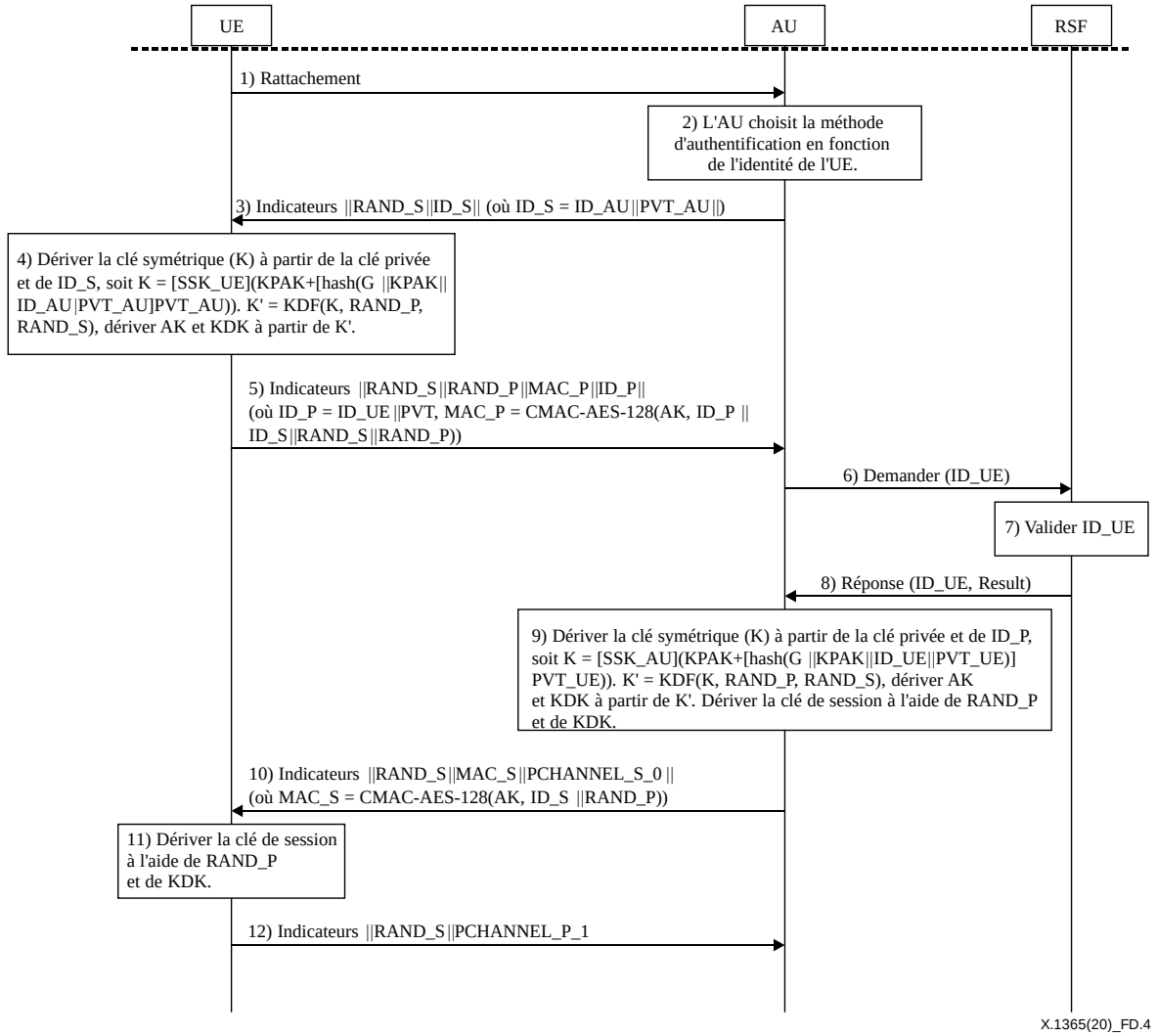

Figure D.4 – EAP-PSK--ECCSI

D.4.1 Rattachement

Ce message reproduit la procédure d'authentification.

D.4.2 Premier message EAP-PSK--ECCSI (message 3 sur la Figure D.4)

Le premier message EAP-PSK--ECCSI est envoyé par le serveur à l'entité homologue. Le format est le suivant.

Le premier message EAP-PSK--ECCSI est composé des éléments suivants:

– un champ Indicateurs d'un octet;

– un nombre aléatoire de 16 octets: RAND_S;

– un champ de longueur variable transmettant le NAI du serveur: ID_S. La longueur de ce champ est déduite à partir de la longueur du champ de l'EAP. La longueur de ce NAI ne doit pas excéder 966 octets. Cette restriction vise à éviter les problèmes de fragmentation.

La Figure D.5 offre un exemple de format du premier message concernant l'EAP-PSK.

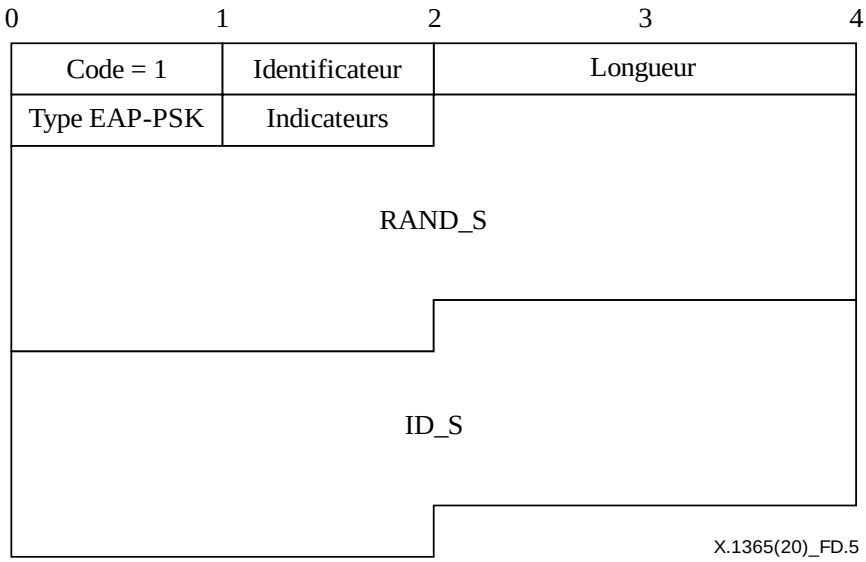

Figure D.5 – Format de l'EAP-PSK

Afin de prendre en charge l'authentification EAP-PSK fondée sur l'IBC, l'ID_S en lien avec le protocole EAP-PSK est utilisé pour transporter l'ID_AU et le PVT_AU. L'ID_S et le PVT_AU sont transportés au sein de la structure de données d'étiquette-longueur-vecteur (TLV), dans laquelle le premier octet transporte un indicateur d'étiquette, et le deuxième octet transporte un champ de longueur indiquant la longueur du champ suivi. Le champ vecteur transporte la valeur.

Le Tableau D.2 définit le TLV pour l'ID et le PVT utilisés avec l'EAP-PSK.

Tableau D.2 – Définition de l'étiquette, de la longueur et du vecteur pour l'identité le jeton de vérification public

	Étiquette	**Longueur**	**Valeur**
Identité	1	Variable (≤ 255)	Définie par le fournisseur de services
PVT	2	65	Nombre en hexadécimal

La Figure D.6 offre une illustration du format du message EAP-PSK--ECCSI transportant l'identité et le PVT au sein du champ ID_S.

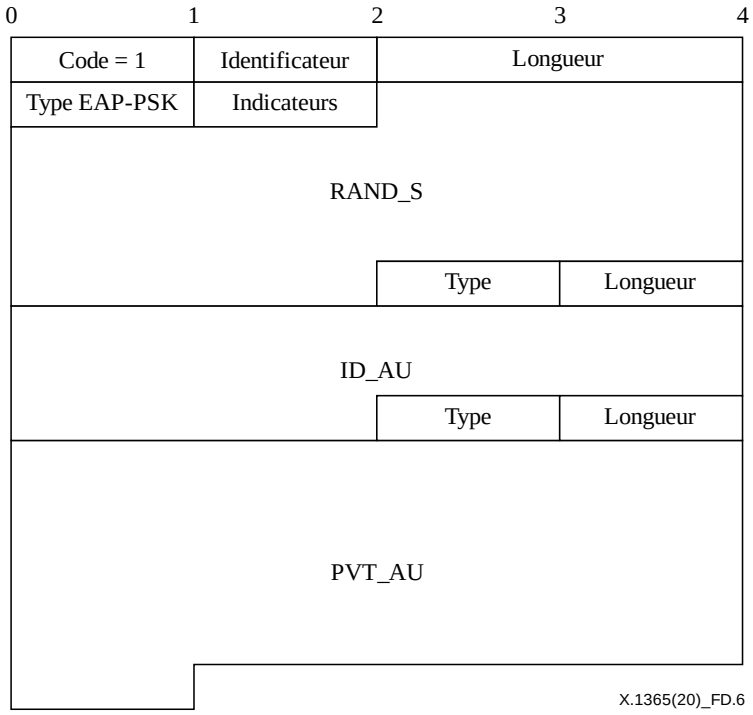

Figure D.6 – Format de message pour l'EAP-PSK--ECCSI

D.4.3 Deuxième message EAP-PSK--ECCSI (message 5 sur la Figure D.4)

Le deuxième message EAP-PSK-ECCSI est envoyé par l'entité homologue au serveur. Ce format consiste en ce qui suit:

– un champ Indicateurs d'un octet;

– le nombre aléatoire de 16 octets envoyé par le serveur dans le premier message EAP-PSK--ECCSI (RAND_S) qui fait office d'ID de session;

– un nombre aléatoire de 16 octets: RAND_P;

– une commande d'accès au support (MAC) de 16 octets: MAC_P;

– un champ de longueur variable transmettant le NAI de l'entité homologue: ID_P. La longueur de ce champ est déduite à partir de la longueur du champ de l'EAP. La longueur de ce NAI ne doit pas excéder 966 octets.

De la même manière, le champ ID_S de l'EAP-PSK est utilisé pour transporter les champs ID_UE et PVT_UE. La Figure D.7 offre une illustration du format du deuxième message EAP-PSK.

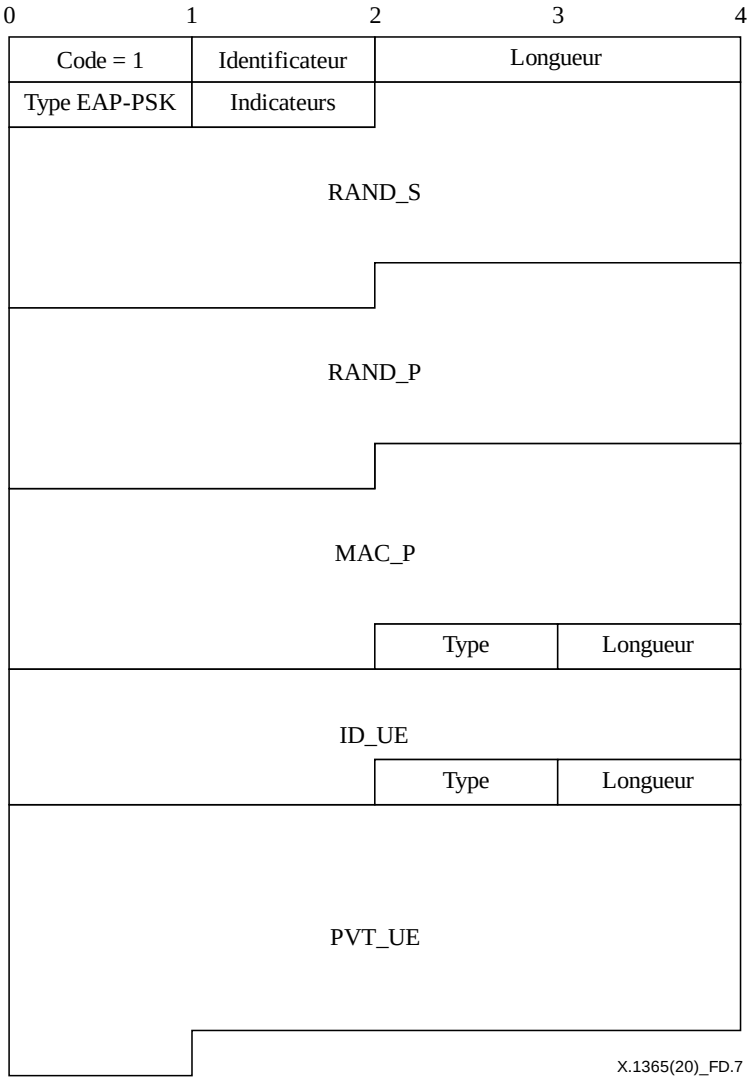

Figure D.7 – Format du message pour le deuxième message concernant l'EAP-PSK--ECCSI

D.4.4 Troisième message EAP-PSK--ECCSI (message 10 sur la Figure D.4)

Le troisième message EAP-PSK--ECCSI est envoyé par le serveur à l'entité homologue. Le format est identique à celui présenté dans le Document [IETF RFC 4764].

D.4.5 Quatrième message EAP-PSK--ECCSI (message 12 sur la Figure D.4-1)

Le quatrième message EAP-PSK-ECCSI est envoyé par l'entité homologue au serveur. Le format est identique à celui présenté dans le Document [IETF RFC 4764].

Appendice I

Dénomination de l'identité

(Cet appendice ne fait pas partie intégrante de la présente Recommandation.)

L'ID d'une application IoT peut être l'ID d'un terminal ou l'ID d'une plate-forme IoT. L'ID désigne le nom servant à l'identification. Un ID constitue une représentation pratique d'un objet et permet à cet objet d'être référencé ou mentionné dans une base de données ou des protocoles de communication, par exemple. En vue d'atteindre cet objectif, les ID doivent être uniques ou un ID doit être unique dans un système indépendant. Par exemple, un code postal est unique dans un pays. Le caractère unique de l'ID s'inscrit dans un cadre précis. En outre, un ID peut non seulement représenter un objet unique, mais aussi un groupe d'objets, ce qui assure une gestion et un fonctionnement uniformes de ce groupe.

Les OID, tels que définis dans les Documents [b-UIT-T X.660] et [b-UIT-T X-Sup.31], sont élaborés conjointement par l'ISO/CEI et l'UIT-T et présentent de nombreuses caractéristiques. Un OID possède une structure arborescente hiérarchique, apte à étendre ses couches et la longueur des ID de façon flexible. Un OID correspond à un nœud sur l'arbre à OID, qui est apte à identifier tout élément quel qu'il soit (qu'il s'agisse d'un élément physique ou virtuel et qu'il s'agisse ou non d'un dispositif) et de le connecter à des infrastructures mondiales d'information et de communication. La racine de l'arbre contient les trois arcs suivants: 0 (UIT-T), 1 (ISO) et 2 (joint-iso-itu-t). Chaque nœud de l'arbre est représenté par une série de nombres entiers séparés par des points, qui correspondent au chemin menant de la racine au nœud, en passant par la série de nœuds ancêtres. Chaque niveau d'ID d'autorité d'enregistrement doit être attribué par l'autorité d'enregistrement de niveau supérieur. Par exemple, l'OID désignant le centre national d'enregistrement des cartes à puce de Chine, à savoir 1.2.156.20005, est attribué par 1.2.156 (ISO.member.china), l'OID du centre national d'enregistrement des OID de Chine.

Un OID complet est en somme l'association d'un ID d'autorité d'enregistrement et d'un ID d'entité, ces deux composants étant séparés par un point, comme l'illustre la Figure I-1. Si une entreprise possède déjà un OID émanant de l'autorité d'enregistrement de niveau supérieur, seul l'ID d'entité doit être conçu.

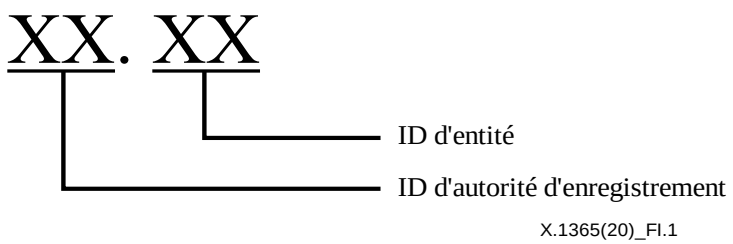

Figure I.1 – Structure d'OID complets pour objets

Par exemple, l'ID d'entité doit présenter la structure décrite dans le Tableau I.1.

Tableau I.1 – Informations détaillées sur l'ID d'entité

Octets	Parties constitutives	Interprétation
1	version et réservés	4 bits pour la version de l'ID d'entité et 4 bits pour les chiffres réservés pour l'avenir
2	entreprise	type d'entreprise
3~11	temps écoulé	période de non-validité de l'identité, 5 octets pour l'heure d'émission en format Unix et 4 octets pour la période de validité en secondes
12	type	valeur 0 pour les nombres insignifiants, 1 pour le MAC et 2 pour l'IMSI
13	longueur (valeur *l*)	la taille de la partie valeur (en octets), 6 pour le MAC et 8 pour l'IMSI
14~13+*l*	value	numéro d'identification individuel

La longueur de l'ID d'entité est de 19 octets lorsque le MAC est utilisé à titre de numéro d'identification individuel et de 21 octets lorsque l'IMSI est utilisé. Un IMSI se présente généralement sous la forme d'un nombre comportant un maximum de 15 chiffres dont le premier est autre que zéro, sauf pour le réseau de test (Document [b-UIT-T E.212]). Lorsque l'on remplit de zéros l'espace précédant l'IMSI jusqu'à atteindre 16 chiffres et que l'on utilise 4 octets pour un chiffre, 8 octets sont suffisants pour un IMSI.

La plate-forme IoT tient à jour une liste aux fins d'adressage. Lorsqu'un dispositif terminal est enregistré pour la première fois, la plate-forme ajoute une ligne, sur laquelle figurent l'ID et l'adresse IP du dispositif. L'adresse IP d'un dispositif peut être obtenue en recherchant l'ID du dispositif dans la liste. Voir Tableau I.2.

Tableau I.2 – Exemple de liste d'adressage

Identificateur	Adresse IP
1.2.9c.4e25.10.1.5b3e408003c26700.1.6.38B1DBC3156F	192.168.0.1

Appendice II

Extensions du KMIP aux fins de la prise en charge de l'IBC

(Cet Appendice ne fait pas partie intégrante de la présente Recommandation.)

Il est possible d'étendre le KMIP comme indiqué ci-après afin de prendre en charge les opérations d'IBC avec le KMS, en particulier l'initialisation du système avec le KMIP et la génération de clés privées lors des opérations avec le KMIP, tel que le décrivent, respectivement, les paragraphes C.1 et C.4.

La charge utile des demandes pour l'opération de création de paires de clés se compose des éléments indiqués dans le Tableau II.1.

Tableau II.1 – Charge utile des demandes

Objet	Requis	Description
Modèle de clé privée-attribut	Oui	Spécifie les attributs lorsque la fonction **IBSetup** génère *ib.msk* et *ib.pubparam*.

Le modèle de clé privée-attribut comprend les attributs énumérés dans le Tableau II.2.

Tableau II.2 – Modèle de clé privée-attribut

Objet	Requis	Codage	Description
Algorithme de chiffrement	Oui	Énumération, voir le Tableau II.3	Spécifie la fonction **IBSetup**.
Longueur de chiffrement	Non	Entier	Indique la longueur en bits des caractéristiques du champ premier sur lequel la courbe elliptique est basée.
Masque d'utilisation cryptographique	Oui	Entier	Indique l'utilisation d'*ib.msk* qui est de signer aux fins de la génération de clés. **IBExtract** est essentiellement un processus de signature.
Paramètres de domaine cryptographique	Oui	Objet	Décrit des paramètres supplémentaires pour choisir les paramètres système, tels que la courbe elliptique utilisée.
Paramètres de chiffrement	Oui	Objet	Décrit d'autres fonctions, telles que la fonction de hachage, qui doit être utilisée avec les fonctions **IBExtract**.

L'algorithme de chiffrement compte parmi les valeurs répertoriées dans le Tableau II.3.

Tableau II.3 – Algorithme de chiffrement (génération de clés)

Nom	Valeur
IBC-KGA-BB1	00000030
IBC-KGA-BF	00000031
IBC-KGA-ECCSI	00000032
IBC-KGA-SK	00000033
IBC-KGA-SM9	00000034

La longueur de chiffrement est une valeur égale ou supérieure à 110.

L'utilisation cryptographique doit être définie comme 00000001 (signature).

Les paramètres de domaine cryptographique comprennent les attributs énumérés dans le Tableau II.4.

Tableau II.4 – Paramètres de domaine cryptographique

Objet	Requis	Codage	Description
QLength	Non	Entier	Indique la longueur en bits de l'ordre du groupe dans lequel *ib.msk* est choisi.
Courbe recommandée	Oui	Énumération, voir le Tableau II.5	Indique la courbe utilisée.
Type de couplage	Non	Énumération, voir le Tableau II.6	Si utilisé, décrit le couplage dans un algorithme fondé sur l'identité.
Nom de domaine	Non	CHAÎNE DE TEXTE	Indique un nom unique pour les paramètres système générés *ib.pubparam*.
Série du domaine	Non	INTEGER	Indique un numéro de version pour les paramètres système générés *ib.pubparam*.

La courbe recommandée compte parmi les valeurs répertoriées dans le Tableau II.5.

Tableau II.5 – Courbe recommandée

Nom	Valeur
IBC-CURVE-SS1	00000070
IBC-CURVE-SS2	00000071
IBC-CURVE-BN-254-1	00000072
IBC-CURVE-BN-256-1	00000073
IBC-CURVE-BN-256-2	00000074
IBC-CURVE-BN-382-1	00000077
IBC-CURVE-BLS-12-381-1	0000007A
IBC-CURVE-BLS-12-442-1	0000007B
IBC-CURVE-BLS-12-455-1	0000007C
IBC-CURVE-BLS-12-461-1	0000007D
IBC-CURVE-KSS-16-340-1	0000007E
IBC-CURVE-KSS-18-348-1	0000007F

Le type de couplage compte parmi les valeurs répertoriées dans le Tableau II.6.

Tableau II.6 – Type de couplage

Nom	Valeur
Weil-Pairing	00000001
Tate-Pairing	00000002
Optimal-Ate-Pairing	00000003

Les paramètres de chiffrement comprennent les attributs énumérés dans le Tableau II.7.

Tableau II.7 – Paramètres de chiffrement

Objet	Requis	Codage	Description
Algorithme de hachage	Oui	Énumération, voir le Tableau II.8	Décrit la fonction de hachage qui est utilisée avec la fonction de génération de clés.
Groupe de clés privées	Non	Énumération, voir le Tableau II.9	Indique le groupe dans lequel la clé privée est générée en cas d'utilisation d'un couplage.

L'algorithme de chiffrement compte parmi les valeurs répertoriées dans le Tableau II.8.

Tableau II.8 – Algorithme de chiffrement (hachage)

Nom	Valeur
SHA224	00000040
SHA256	00000041
SHA384	00000042
SHA512	00000043
SHA3-224	00000044
SHA3-256	00000045
SHA3-384	00000046
SHA3-512	00000047
SM3	00000048

Le groupe de clés privées compte parmi les valeurs répertoriées dans le Tableau II.9.

Tableau II.9 – Groupe de clés privées

Nom	Valeur
IBC-PRK-GROUP1	00000001
IBC-PRK-GROUP2	00000002
IBC-PRK-TWOGROUPS	00000003

La charge utile des réponses pour l'opération de création d'une paire de clés se compose des éléments indiqués dans le Tableau II.10.

Tableau II.10 – Charge utile des réponses

Objet	Requis	Description
Identificateur unique de clé privée	Oui	Identificateur unique d'un objet clé privée nouvellement créé pouvant être utilisé pour accéder à *ib.msk*. L'identificateur est codé en tant que chaîne de texte.
Identificateur unique de clé publique	Oui	Identificateur unique d'un objet clé publique nouvellement créé pouvant être utilisé pour accéder à *ib.pubparam*. L'identificateur est codé en tant que chaîne de texte.

La charge utile des demandes pour l'opération d'obtention se compose des éléments indiqués dans le Tableau II.11.

Tableau II.11 – Charge utile des demandes

Objet	Requis	Description
Identificateur unique de clé publique	Oui	Identificateur unique d'un objet clé publique pouvant être utilisé pour accéder à *ib.pubparam*. L'identificateur est codé en tant que chaîne de texte.

La charge utile des réponses pour l'opération d'obtention se compose des éléments indiqués dans le Tableau II.12.

Tableau II.12 – Charge utile des réponses

Objet	Requis	Description
Type d'objet	Oui	Type d'objet
Identificateur unique	Oui	Identificateur unique de l'objet
Clé publique	Oui	Structure de clé publique encapsulant les données des paramètres publics de l'IBC *ib.pubparam*.

L'ID unique est identique à l'ID unique de clé publique envoyé dans la charge utile des réponses pour l'opération d'obtention.

Le type d'objet est 00000003 (clé publique).

Le bloc de clé dans le champ clé publique se compose des éléments indiqués dans le Tableau II.13.

Tableau II.13 – Bloc de clé dans le champ clé publique

Objet	Requis	Codage	Description
Type de format de clé	Oui	Énumération, voir le Tableau II.14	Décrit le format de la valeur clé.
Compression de la clé	Non	Énumération	Indique si la valeur clé doit être compressée.
Valeur clé	Oui	Structure de clé transparente pour les paramètres publics d'IBC.	Structure de clé transparente nouvellement définie pour la technologie IBC de clé publique.
Algorithme de chiffrement	Oui	Énumération, voir le Tableau II.15	Identique à la charge utile des demandes pour l'opération de création de paires de clés

Le type de format de clés correspond à la valeur indiquée dans le Tableau II.14.

Tableau II.14 – Type de format de clé

Nom	Valeur
Paramètres publics d'IBC transparents	00000016

La compression de la clé équivaut à 00000001 (non compressé) ou 00000002 (clé d'origine compressée).

La valeur clé possède les attributs énumérés dans le Tableau II.15:

Tableau II.15 – Valeur clé

Objet	Requis	Codage	Description
P	Non	Grand nombre entier	Pour les courbes basées sur un champ premier, P désigne la caractéristique (p) du champ premier.
Q	Non	Grand nombre entier	Q est l'ordre du sous-groupe de points (*G*1) dans lequel les opérations de chiffrement sont calculées.
J	Non	Grand nombre entier	J est le cofacteur tel que J*Q = X-1, où X désigne l'ordre du groupe de points de la courbe spécifiée.
CHAÎNE P1	Oui	CHAÎNE D'OCTETS	Pour les algorithmes fondés sur des couplages, P1 désigne le générateur de groupe *G*1 du couplage. Pour les algorithmes non fondés sur des couplages, P1 désigne un générateur du sous-groupe de points de fonctionnement.
CHAÎNE P2	Non	CHAÎNE D'OCTETS	Pour les algorithmes fondés sur des couplages, P2 désigne le générateur de groupe *G*2 de couplage.
CHAÎNE sP1	Non	CHAÎNE D'OCTETS	sP1 est le résultat scalaire de [*ib.msk*]P1 ou le résultat scalaire d'un composant entier de *ib.msk* avec P1.
CHAÎNE sP2	Non	CHAÎNE D'OCTETS	Pour les algorithmes fondés sur des couplages, sP2 est le résultat scalaire de [*ib.msk*]P2 ou le résultat scalaire d'un composant entier de *ib.msk* avec P2.
CHAÎNE sP3	Non	CHAÎNE D'OCTETS	Pour certains algorithmes fondés sur le couplage (en particulier les algorithmes utilisant la fonction de génération de clés BB1), sP3 désigne le résultat scalaire d'un autre composant entier d'ib.msk avec P1.
CHAÎNE de couplage public	Non	CHAÎNE D'OCTETS	Pour certains algorithmes fondés sur le couplage, le couplage public est le résultat du couplage(P1, [s]P2), du couplage([s]P1, P2) ou du couplage(P1, P2), où s désigne *ib.msk* pour les algorithmes tels que SM9, SK-KEM ou ([s1]P1, [s2]P2) pour BB1-KEM, où s1, s2 sont des composants entiers d'*ib.msk*.

Les nouvelles définitions d'étiquettes sont énumérées dans le Tableau II.16.

Tableau II.16 – Définitions d'étiquettes

Objet	Valeur de l'étiquette
Type de couplage	420100
Groupe de clés privées	420101
Nom de domaine	420102
Série du domaine	420103
CHAÎNE P1	420104
CHAÎNE sP1	420105
CHAÎNE P2	420106
CHAÎNE sP2	420107
CHAÎNE sP3	420108
CHAÎNE de couplage public	420109

La charge utile des demandes pour l'opération de signature se compose des éléments indiqués dans le Tableau II.17.

Tableau II.17 – Charge utile des demandes pour l'opération de signature

Objet	Requis	Description
Identificateur unique	Non	Identificateur unique de l'objet cryptographique géré, à savoir la clé *ib.msk* à utiliser pour l'opération **IBExtract**. En cas d'omission, la valeur de l'espace réservé de l'ID est utilisée par le serveur à titre d'identificateur unique.
Paramètres de chiffrement	Non	Les paramètres de chiffrement peuvent indiquer le groupe dans lequel la clé privée est générée.
Données	Oui	Les données indiquent la valeur d'identité à partir de laquelle la clé privée est extraite.

Les paramètres de chiffrement comprennent les attributs énumérés dans le Tableau II.18.

Tableau II.18 – Paramètres de chiffrement

Objet	Requis	Codage	Description
Groupe de clés privées	Non	Énumération, voir Tableau II.9	Indiquent le groupe (*G*1 ou *G*2) dans lequel la clé privée est générée.

Bibliographie

[b-UIT-T E.101] Recommandation UIT-T E.101 (2009), *Définition des termes utilisés pour les identificateurs (noms, numéros, adresses et autres identificateurs) concernant les services et les réseaux publics de télécommunication dans les Recommandations de la série E*.

[b-UIT-T E.212] Recommandation UIT-T E.212 (2016), *Plan d'identification international pour les réseaux publics et les abonnements*.

[b-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*.

[b-UIT-T X.660] Recommandation UIT-T X.660 (2011), *Technologies de l'information – Procédures opérationnelles des autorités d'enregistrement des identificateurs d'objet: procédures générales et arcs sommitaux de l'arborescence des identificateurs d'objet internationaux*.

[b-UIT-T X.1361] Recommandation UIT-T X.1361 (2018), *Cadre de sécurité applicable à l'Internet des objets fondé sur le modèle passerelle*.

[b-UIT-T X-Sup.31] Supplément 31 aux Recommandations UIT-T de la série X (2017), *UIT-T X.660 – Lignes directrices relatives à l'utilisation des identificateurs d'objet pour l'Internet des objets*.

[b-UIT-T Y.2720] Recommandation UIT-T Y.2720 (2009), *Cadre de gestion d'identité des réseaux NGN*.

[b-UIT-T Y.4000] Recommandation UIT-T Y.4000/Y.2060 (2012), *Présentation générale de l'Internet des objets*.

[b-UIT-T Y.4100] Recommandation UIT-T Y.4100/Y.2066 (2014), *Exigences communes relatives à l'Internet des objets.*

[b-ISO/CEI 9798-3] ISO/CEI 9798-3:2019. *Techniques de sécurité IT – Authentification d'entité – Partie 3: Mécanismes utilisant des techniques de signature numériques*.

[b-ETSI TR 118 508] ETSI TR 118 508 V1.0.0 (2014), *Analysis of Security Solutions for the oneM2M System*.
<https://www.etsi.org/deliver/etsi_tr/118500_118599/118508/01.00.00_60/tr_118508v010000p.pdf>

[b-ETSI TS 133.501] ETSI TS 133 501 V15.2.0 (2018), *5G; Security architecture and procedures for 5G system (3GPP TS 33.501 version 15.1.0 Release 15)*.
<https://www.etsi.org/deliver/etsi_ts/133500_133599/133501/15.01.00_60/ts_133501v150100p.pdf>

[b-GM/T 0044.2] GM/T 0044.2-2016, *Identity-based cryptographic algorithms SM9 – Part 2: Digital signature algorithm*.

[b-GSMA SGP.02] GSMA Official Document SGP.02 Version 3.1 (2016), *Remote Provisioning Architecture for Embedded UICC – Technical Specification*.

[b-IANA TLS REG] Internet Assigned Numbers Authority (IANA), *Transport Layer Security (TLS) Parameters*. Disponible sur Internet, consulté pour la dernière fois le 12/07/2019.
<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml>

[b-IEEE 1363] IEEE 1363-2000, *IEEE Standard Specifications for Public-Key Cryptography*.

[b-IEEE P1363.3] IEEE P1363.3/D9 (mai 2013), *IEEE Standard for Identity-Based Cryptographic Techniques using Pairings*.

[b-IETF RFC 3748] IETF RFC 3748 (2004). *Protocole d'authentification extensible (EAP)*.

[b-OASIS KMIP] OASIS (2016), *Key Management Interoperability Protocol Specification Version 1.3*. <http://docs.oasis-open.org/kmip/spec/v1.3/os/kmip-spec-v1.3-os.pdf>

[b-Barreto] Barreto, P. S. L. M., Libert, B., McCullagh, N., Quisquater, J.-J. (2005). Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Roy B. (éd.). *Advances in Cryptology – ASIACRYPT 2005*, pp. 515-532. *Lecture Notes in Computer Science*, vol. 3788. Berlin: Springer

[b-Chen] Chen, L., Malone-Lee, J. (2005). Improved identity-based signcryption. In: Vaudenay S. (éd). *Public Key Cryptography – PKC 2005*, p. 362-379. *Lecture Notes in Computer Science*, vol. 3386. Berlin: Springer.

[b-Ducas] Ducas, L., Lyubashevsky, V., Prest, T. (2014). Efficient identity-based encryption over NTRU lattices. In: Sarkar P., Iwata T. (éd.). *Advances in Cryptology – ASIACRYPT 2014*, p. 22-41. *Lecture Notes in Computer Science*, vol. 8874. Berlin: Springer.

[b-Freeman] Freeman, D., Scott, M., Teske, E. (2010). A taxonomy of pairing-friendly elliptic curves. *J. Cryptol*. **23**, p. 224-280.

[b-Galbraith] Galbraith, S.D., Paterson, K.G., Smart, N.P. (2008). Pairings for cryptographers. *Discrete Appl. Math.*, **156**, p. 3113-3121.

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	Équipements 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

Imprimé en Suisse
Genève, 2020