Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.1341

(09/2015)

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

Applications et services sécurisés – Recommandations relatives aux infrastructures de clé publique

Protocoles de transport de courrier certifié et de bureau de poste certifié

Recommandation UIT-T X.1341

Union internationale des télécommunications

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	
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	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 pollupostage	X.1230–X.1249
Gestion des identités	X.1250–X.1279
APPLICATIONS ET SERVICES SÉCURISÉS	
Communications d'urgence	X.1300–X.1309
Sécurité des réseaux de capteurs ubiquitaires	X.1310–X.1339
Recommandations relatives aux infrastructures de clé publique	**X.1340–X.1349**
ECHANGE D'INFORMATIONS SUR LA CYBERSÉCURITÉ	
Aperçu général de la cybersécurité	X.1500–X.1519
Echange concernant les vulnérabilités/les états	X.1520–X.1539
Echange concernant les événements/les incidents/l'heuristique	X.1540–X.1549
Echange de politiques	X.1550–X.1559
Heuristique et demande d'informations	X.1560–X.1569
Identification et découverte	X.1570–X.1579
Echange 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 oeuvre de la sécurité de l'informatique en nuage	X.1660–X.1679
Sécurité de l'informatique en nuage (autres)	X.1680–X.1699

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

Recommandation UIT-T X.1341

Protocoles de transport de courrier certifié et de bureau de poste certifié

Résumé

La Recommandation UIT-T X.1341 définit le protocole de transfert de courrier certifié (CMTP) et le protocole de bureau de poste certifié (CPOP) afin de favoriser l'échange de courriers électroniques certifiés dans le monde de manière sécurisée en assurant la confidentialité, l'identification des correspondants, l'intégrité et la non-répudiation.

Historique

Edition	Recommandation	Approbation	Commission d'études	ID unique*
1.0	ITU-T X.1341	2015-09-17	17	11.1002/1000/12352

Mots clés

Protocole de transfert de courrier certifié (CMTP), protocole de bureau de poste certifié (CPOP), confidentialité, intégrité, non-répudiation, protocole de bureau de poste (POP), sécurité, protocole simple de transfert de courrier (SMTP)

* 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 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 avait 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

Introduction

La présente Recommandation étend les capacités du protocole simple de transfert de courrier (SMTP) et de la version 3 du protocole de bureau de poste (POP3) pour prendre en charge l'authentification, la sécurité et la non-répudiation.

Pour ce faire, deux protocoles sont définis:

– le protocole de transfert de courrier certifié (CMTP), qui est une extension du protocole simple de transfert de courrier (SMTP), est le protocole qui prend en charge les communications entre l'expéditeur des courriers électroniques et un serveur de messagerie, appelé serveur de messagerie certifiée (Cmail);

– le protocole de bureau de poste certifié (CPOP), qui est une extension de la version 3 du protocole de bureau de poste (POP3), est le protocole qui prend en charge les communications entre le destinataire des courriers électroniques et le serveur Cmail.

Selon les protocoles SMTP et POP3, un type de message est identifié par une commande, c'est-à-dire par un mot clé au début du message. Pour les protocoles CMTP et CPOP, on a défini de nouvelles commandes et étendu des commandes SMTP et POP3. En particulier, certaines commandes ont été étendues en vue de transmettre des notifications (documents électroniques) permettant d'étayer et de vérifier les différentes étapes de la communication depuis l'expéditeur vers le destinataire.

Les protocoles CMTP et CPOP introduisent en outre le concept de serveur Cmail, partenaire actif de la communication entre l'expéditeur et le destinataire qui permet d'attester que l'échange entre les deux parties a effectivement eu lieu.

La messagerie certifiée suppose qu'une infrastructure de clé publique (PKI) soit établie.

L'Annexe A, qui fait partie intégrante de la présente Recommandation, donne les spécifications formelles des notifications utilisant la technique de notation de définition de schéma XML (XSD).

L'Annexe B, qui fait partie intégrante de la présente Recommandation, donne les spécifications formelles des notifications utilisant la notation de syntaxe abstraite numéro un (ASN.1).

L'Annexe C, qui fait partie intégrante de la présente Recommandation, définit les exigences concernant les certificats de clé publique délivrés aux clients (expéditeur et destinataire des courriers électroniques) et aux serveurs Cmail.

L'Annexe D, qui fait partie intégrante de la présente Recommandation, définit les exigences concernant l'utilisation de la spécification de sécurité dans la couche transport (TLS).

L'Annexe E, qui fait partie intégrante de la présente Recommandation, spécifie les identificateurs d'objet définis pour les serveurs Cmail.

Recommandation UIT-T X.1341

Protocoles de transport de courrier certifié et de bureau de poste certifié

1 Domaine d'application

La présente Recommandation spécifie comment assurer la fiabilité des courriers électroniques en termes d'identification et de confidentialité.

Le protocole de transfert de courrier certifié/protocole de bureau de poste certifié (CMTP/CPOP) permet:

– de résoudre les problèmes de répudiation avec l'utilisation d'une signature électronique;

– de résoudre les problèmes de confidentialité avec l'utilisation d'un chiffrement;

– de produire des notifications fiables de dépôt, de transit et de réception;

– d'utiliser un serveur de messagerie certifiée (Cmail) pour suivre les courriers certifiés afin d'éviter leur perte pendant le processus;

– d'utiliser une connexion TLS (sécurité dans la couche transport) afin d'assurer une identification plus sûre. Ce niveau plus élevé d'identification est exigé par le serveur Cmail.

La conformité à la présente Recommandation ne doit pas être considérée comme une preuve permettant de déclarer la conformité à une législation, une réglementation ou une politique nationale ou régionale. Les moyens techniques et ceux relatifs à l'organisation et aux procédures décrits dans la présente Recommandation ne garantissent en aucune façon de parvenir à un niveau de sécurité susceptible d'être imposé pour certaines correspondances par une législation, une réglementation ou une politique nationale ou régionale spécifique.

2 Références

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

[UIT-T X.520] Recommandation UIT-T X.520 (2012) | ISO/CEI 9594-6:2014, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés*.

[UIT-T X.680] Recommandation UIT-T X.680 (2008) | ISO/CEI 8824-1:2008, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

[UIT-T X.690] Recommandation UIT-T X.690 (2008) | ISO/CEI 8825-1:2008, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*

[UIT-T X.693] Recommandation UIT-T X.693 (2008) | ISO/CEI 8825-4:2008, *Technologies de l'information – Règles de codage ASN.1: règles de codage XML (XER).*

[ISO 3166-1] ISO 3166-1:2013, *Codes pour la représentation des noms de pays et de leurs subdivisions – Partie 1: Codes de pays*.

[IETF RFC 822] IETF RFC 822 (1982), *Standard for the format of ARPA Internet text messages.*

[IETF RFC 1939] IETF RFC 1939 (1996), *Post office protocol – Version 3*.

[IETF RFC 2045] IETF RFC 2045 (1996), *Multipurpose internet mail extensions (MIME) – Part One: Format of internet message bodies*.

[IETF RFC 5246] IETF RFC 5246 (2008), *The transport layer security (TLS) Protocol – Version 1.2*.

[IETF RFC 5321] IETF RFC 5321 (2008), *Simple mail transfer protocol*.

[XML] W3C Recommendation XML1.0 (2000), *Extensible markup language (XML) 1.0 (fifth edition).*

[XSD] W3C Recommendation XML Schema (2001), *XML schema Part 1: Structures*.

3 Définitions

3.1 Termes définis ailleurs

La présente Recommandation utilise les termes suivants définis ailleurs:

3.1.1 autorité de certification (CA, *certification authority***)** [b-UIT-T X.509]: autorité jouissant de la confiance d'un ou de plusieurs utilisateurs pour la création et l'attribution de certificats de clé publique. L'autorité de certification peut, de manière optionnelle, créer les clés des utilisateurs.

3.1.2 validation de certificat [b-UIT-T X.509]: processus consistant à s'assurer qu'un certificat était valide à un instant donné, impliquant éventuellement la construction et le traitement d'un itinéraire de certification avec la garantie que tous les certificats de l'itinéraire étaient valides (c'est-à-dire, non caducs ou révoqués) à l'instant donné.

3.1.3 fonction de hachage [b-UIT-T X.509]: fonction (mathématique) qui fait correspondre un argument pris dans un domaine étendu (éventuellement très étendu) à une valeur appartenant à un domaine plus réduit. Une "bonne" fonction de hachage est telle que l'application de la fonction à un ensemble (étendu) d'arguments du premier domaine fournira des valeurs réparties de manière égale (apparemment aléatoire) dans le second domaine.

3.1.4 clé privée [b-UIT-T X.509]: (dans un système de chiffrement avec clé publique) celle des clés d'une paire de clés d'une entité qui est connue uniquement par l'entité concernée.

3.1.5 clé publique [b-UIT-T X.509]: (dans un système de chiffrement avec clé publique) celle des clés d'une paire de clés d'un utilisateur qui est connue de manière publique.

3.1.6 certificat de clé publique (PKC, *public key certificate***)** [b-UIT-T X.509]: clé publique d'un utilisateur, associée à certaines autres informations qui sont rendues non falsifiables par signature numérique en utilisant la clé privée de l'autorité de certification émettrice.

3.1.7 infrastructure de clé publique (PKI, *public key infrastructure***)** [b-UIT-T X.509]: infrastructure pouvant prendre en charge la gestion de clés publiques afin de fournir des services d'authentification, de chiffrement, d'intégrité et de non-répudiation.

3.2 Termes définis dans la présente Recommandation

La présente Recommandation définit les termes suivants:

3.2.1 courrier certifié: courrier électronique échangé à l'aide du protocole de transfert de courrier certifié (CMTP) et du protocole de bureau de poste certifié (CPOP).

3.2.2 protocole de transfert de courrier certifié (CMTP, *certified mail transfer protocol***)**: protocole de couche application sur la connexion TCP/IP (protocole de commande de

transmission/protocole Internet) fondé sur le protocole simple de transfert de courrier (SMTP) et utilisé pour envoyer des courriers certifiés.

3.2.3 protocole de bureau de poste certifié (CPOP, *certified post office protocol***)**: protocole de couche application sur la connexion TCP/IP (protocole de commande de transmission/protocole Internet) fondé sur la version 3 du protocole de bureau de poste (POP3) et utilisé pour recevoir des courriers certifiés.

3.2.4 serveur Cmail: entité de confiance impliquée dans des transactions de courrier certifié.

3.2.5 notification de dépôt: document électronique signé par l'expéditeur et le serveur Cmail, contenant des renseignements permettant d'attester qu'un courrier certifié a été déposé.

3.2.6 notification de réception: document électronique signé par le destinataire et le serveur Cmail, contenant des renseignements permettant d'attester qu'un courrier certifié a été reçu par le destinataire.

3.2.7 notification de transit: document électronique signé par les serveurs Cmail impliqués dans la transaction contenant des renseignements permettant d'attester qu'un courrier certifié a été transmis au serveur Cmail.

3.2.8 version 3 du protocole de bureau de poste (POP3, *post office protocol version 3*): protocole de couche application sur la connexion TCP/IP (protocole de commande de transmission/protocole Internet) utilisé pour recevoir des courriers électroniques.

3.2.9 protocole simple de transfert de courrier (SMTP, *simple mail transfer protocol***)**: protocole de couche application sur la connexion TCP/IP (protocole de commande de transmission/protocole Internet) utilisé pour envoyer des courriers électroniques.

4 Abréviations et acronymes

La présente Recommandation utilise les abréviations et acronymes suivants:

AES norme de cryptage évoluée (*advanced encryption standard*)

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

CA autorité de certification (*certification authority*)

CBC chaînage de bloc de chiffrement (*cipher block chaining*)

Cmail messagerie certifiée (*certified mail*)

CMTP protocole de transfert de courrier certifié (*certified mail transfer protocol*)

CPOP protocole de bureau de poste certifié (*certified post office protocol*)

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

DNS système de noms de domaine (*domain name system*)

id identité

MIME extensions de courrier Internet à fonctions multiples (*multipurpose Internet mail extensions*)

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

POP3 version 3 du protocole de bureau de poste (*post office protocol version 3*)

RSA algorithme de Rivest, Shamir et Adleman

RSCK clé de chiffrement symétrique aléatoire (*random symmetric cipher key*)

S/MIME extensions de courrier Internet à fonctions multiples/sécurisées (*secure multipurpose Internet mail extensions*)

SMTP protocole simple de transfert de courrier (*simple mail transfer protocol*)

TCP/IP protocole de commande de transmission/protocole Internet (*transmission control protocol/Internet protocol*)

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

UTF-8 format de transformation à 8 bits pour le jeu de caractères universel (*universal character set transformation format 8*)

XER règles de codage XML (*XML encoding rules*)

XML langage de balisage extensible (*extensible markup language*)

XSD définition de schéma XML (*XML schema definition*)

5 Conventions

Aucune.

6 Concepts de base du courrier certifié

Pour les communications par courrier électronique classiques utilisant le protocole simple de transfert de courrier (SMTP) et la version 3 du protocole de bureau de poste (POP3), le destinataire d'un courrier électronique peut nier avoir reçu le courrier en question, et ce même avec l'ajout d'extensions de courrier Internet à fonctions multiples/sécurisées (S/MIME) à la suite de protocoles. Les extensions S/MIME permettent le chiffrement des messages et l'authentification de l'expéditeur, mais ne fournissent pas de preuves que le message a été remis.

La présente Recommandation définit une suite de protocoles appelée messagerie certifiée, comprenant le protocole de transfert de courrier certifié (CMTP) et le protocole de bureau de poste certifié (CPOP).

Dans le cas de communications utilisant le protocole SMTP/POP3, le serveur de messagerie, qui ne participe pas activement aux communications, fait uniquement suivre les messages tels qu'ils sont reçus lorsque le destinataire se connecte au serveur de messagerie, et ce même avec l'utilisation d'extensions S/MIME.

Dans le cas de courriers certifiés, le serveur de messagerie participe activement à la communication entre l'expéditeur et le destinataire, de sorte que le serveur Cmail peut vérifier que le destinataire a accepté de recevoir le courrier. Le courrier est envoyé sous une forme cryptée, ce qui ne permet pas au serveur Cmail de lire le contenu effectif du courrier électronique. On trouvera ci-après une description de la procédure, tandis que des spécifications détaillées sont données dans la section 8.

Les interactions entre l'expéditeur et le serveur Cmail sont définies dans la section 8.

Les interactions entre le destinataire et le serveur Cmail sont définies dans la section 9.

7 Types de commandes de messagerie certifiée

La messagerie certifiée utilise des commandes SMTP et POP3 existantes, des commandes SMTP et POP3 améliorées et des commandes propres à la messagerie certifiée. Dans les Tableaux 1 et 2, les commandes qui n'ont pas d'équivalents dans le protocole SMTP/POP3 sont indiquées par la mention "Nouvelle", les commandes correspondant à des commandes SMTP/POP3 améliorées sont indiquées par la mention "Modifiée" et les commandes SMTP/POP3 qui sont utilisées sans modification sont indiquées par la mention "Identique".

Un type de commande est un mot clé en lettres majuscules identifiant un type de message précis associé à des spécifications additionnelles pour ce type de message.

7.1 Types de commandes CMTP

Tableau 1 – Commandes CMTP

Commande	Fonction de la commande
CELO Nouvelle	Permet au serveur d'identifier sa prise en charge des commandes CMTP.
DELV Nouvelle	Identifie le mode de remise: certifiedMail.
MAIL FROM Modifiée	Identifie l'expéditeur du message; utilisée sous la forme "MAIL FROM". Si le compte existe sur le serveur, celui-ci renvoie alors le certificat de clé publique de l'expéditeur connu au format base64.
RCPT TO Modifiée	Identifie les destinataires du message; utilisée sous la forme "RCPT TO". Si le compte existe sur le serveur, celui-ci renvoie alors le certificat de clé publique de l'expéditeur connu au format base64. Si le compte existe sur un autre serveur CMTP avec lequel des clés ont été échangées, le serveur interroge le deuxième serveur et envoie le certificat de clé publique du destinataire connu au format base64 avec la commande CHCK RCPT.
CHCK RCPT Nouvelle	Envoyée uniquement si le destinataire est rattaché à un autre serveur Cmail que celui auquel est rattaché l'expéditeur.
DATA Modifiée	Envoyée par un client pour lancer le transfert du contenu du message. Le serveur renvoie une notification de dépôt signée par le serveur que l'expéditeur doit signer.
DEPO Nouvelle	Envoyée par un client pour lancer le transfert du contenu de la notification de dépôt signée par le serveur et contresignée par l'expéditeur.
SEND EVLP Nouvelle	Fait suivre l'enveloppe d'un serveur Cmail à l'autre.
HELP Identique	Retourne une liste de commandes prises en charge par le serveur CMTP.
QUIT Identique	Met fin à la session.

7.2 Types de commandes CPOP

Tableau 2 – Commandes CPOP

Commande	Fonction de la commande
USER Identique	Utilisée pour spécifier le nom de l'utilisateur qui ouvre une session.
PASS Identique	Mot de passe de l'utilisateur qui ouvre une session
LIST Modifiée	Utilisée pour obtenir la liste des messages et leur taille cumulée. Par exemple, l'invocation de la commande LIST sans paramètre associé se traduira par l'affichage de 2 messages+OK (320 octets) et de la liste de messages: identité (id), longueur et mode de remise (le cas échéant) comme CertifiedMail.
RETR Modifiée	Où *N* est un nombre compris entre 1 et le dernier nombre retourné par la commande LIST. Cette commande ne peut pas être utilisée pour extraire un message qui a été marqué comme supprimé. En l'absence de type de remise, le serveur envoie le courrier électronique en utilisant le codage à extensions sécurisées de courrier Internet à fonctions multiples (MIME). Si le mode de remise est défini, le serveur traite le message selon ce mode. Par exemple avec CertifiedMail, le serveur interroge le destinataire avant d'envoyer l'enveloppe en utilisant la commande RCPT.
CHLG RESP Nouvelle	Envoyé par le client pour adresser la notification de réception pour le message et donner la réponse à la question secrète. Si la réponse est correcte, le serveur renvoie alors l'enveloppe MIME.
SEND NORP Nouvelle	Envoie la notification de réception signée.
HELP Identique	Retourne une liste de commandes prises en charge par le service CPOP.
QUIT Identique	Met fin à la session.

8 Spécification CMTP détaillée

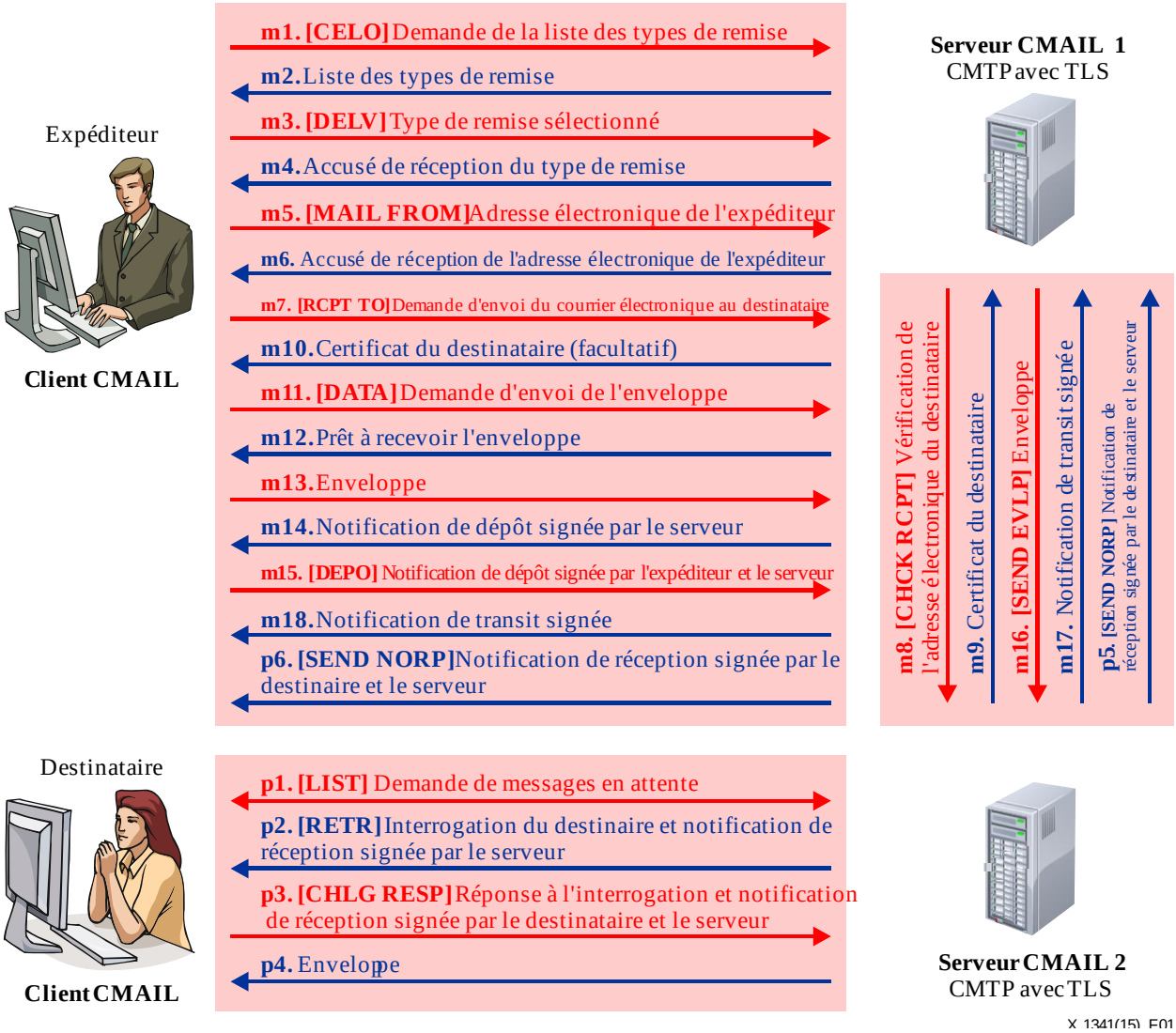

Figure 1 – Description des échanges de protocole

Les commandes précédées de la lettre "m" sont utilisées pour le protocole CMTP et les commandes précédées de la lettre "p" sont utilisées pour le protocole CPOP. Les paragraphes 8.1 à 8.18 spécifient de manière détaillée les échanges m1 à m8 dans la Figure 1, tandis que le § 9 spécifie de manière détaillée les échanges p1 à p6.

8.1 CELO: Demande de la liste des types de remise

Ce type de commande est envoyé sous la forme d'un message SMTP, identique à la commande HELO, suivi d'un nom de domaine pleinement qualifié. Le rôle de cette commande est d'extraire une liste des types de remise.

8.2 Liste des types de remise

La liste des types de remise est donnée en réponse à la commande CELO. Elle est au format SMTP et son contenu est le suivant (insensible à la casse):

250-<nom de domaine pleinement qualifié du serveur Cmail>

250-8BITMIME

250-Delivery-Types CertifiedMail <autres types de remise>

250 OK

La présente Recommandation donne des spécifications uniquement pour le mode CertifiedMail. Des éditions futures pourront spécifier d'autres types de remise.

8.3 Type de remise sélectionné

Ce message identifie le type de remise sélectionné parmi ceux figurant dans la liste des types de remise. Son format (SMTP) est le suivant:

DELV <type de remise>

8.4 Accusé de réception du type de remise

Si le type de remise sélectionné est accepté, ce message a le format SMTP suivant (insensible à la casse):

250 Delivery-Type <type de remise>OK

La réponse ci-après est donnée en cas d'erreur de syntaxe dans le message indiquant le type de remise sélectionné:

501 Syntax: DELV <type de remise>

La réponse ci-après est donnée lorsque le message indiquant le type de remise sélectionné est émis hors séquence:

501 Syntax: use CELO command first

La réponse ci-après est donnée lorsque le message indiquant le type de remise sélectionné est inconnu:

501 Unknown Delivery-Type: <type de remise>

8.5 Adresse électronique de l'expéditeur

Ce message est envoyé au serveur Cmail pour demander l'envoi d'un courrier certifié et, à titre facultatif, demander le certificat de clé publique de l'expéditeur au serveur Cmail.

MAIL FROM <adresse électronique de l'expéditeur> [CertificateRequested]

8.6 Accusé de réception de l'adresse électronique de l'expéditeur

Ce message est envoyé pour confirmer que l'adresse électronique de l'expéditeur existe dans la base de données du serveur Cmail. Si l'expéditeur a demandé son certificat de clé publique, le certificat de clé publique de l'expéditeur est inclus:

[250 User-Certificate: <certificat de clé publique codé au format Base64>]

250 OK

8.7 Demande d'envoi du courrier électronique au destinataire

Ce message est envoyé au serveur Cmail pour demander l'envoi d'un courrier certifié au destinataire et, à titre facultatif, demander le certificat de clé publique du destinataire au serveur Cmail.

RCPT TO <adresse électronique du destinataire> [CertificateRequested]

Cette commande peut être utilisée autant de fois que nécessaire pour ajouter chaque destinataire en cas de destinataires multiples. Les informations indiquant si le destinataire est "To" (destinataire principal) ou "CC" (en copie) figurent dans l'en-tête de l'enveloppe [IETF RFC 5321]. Les destinataires "BCC" (copie cachée) ne sont pas autorisés.

8.8 Vérification de l'adresse électronique du destinataire par le serveur Cmail distant

Ce message est envoyé uniquement si le destinataire est rattaché à un serveur Cmail autre que celui auquel l'expéditeur est rattaché. Il est envoyé par le serveur Cmail de l'expéditeur au serveur Cmail du destinataire pour vérifier la validité de l'adresse électronique et, à titre facultatif, demander le certificat de clé publique du destinataire.

CHCK RCPT <adresse électronique du destinataire> [CertificateRequested]

8.9 Accusé de réception de l'adresse électronique du destinataire

Ce message est envoyé en réponse à la demande de vérification de l'adresse électronique du destinataire par le serveur Cmail distant.

Le message ci-après confirme l'adresse électronique et comprend le certificat de clé publique du destinataire, s'il est demandé:

[250 User-Certificate: <certificat de clé publique codé au format Base64>]

250 OK

Si l'adresse électronique ne peut être confirmée, les messages d'erreur ci-après peuvent être envoyés:

503 Sender already specified

est envoyé en cas de réponse à une demande dupliquée.

501 Syntax: CHCK RCPT <adresse>

est envoyé en cas d'erreur de syntaxe dans l'adresse électronique du destinataire.

501 Syntax: CHCK RCPT <adresse> Error in parameters <paramètre>

est envoyé si le paramètre après l'adresse électronique n'est pas reconnu.

553 <adresse électronique> Invalid email address

est envoyé si l'adresse électronique n'existe pas dans le serveur Cmail distant.

8.10 Accusé de réception de l'adresse électronique du destinataire

Ce message est envoyé pour confirmer que l'adresse électronique du destinataire existe. Si l'expéditeur demande le certificat de clé publique du destinataire, ce certificat est inclus.

Le message ci-après confirme l'adresse électronique et comprend le certificat de clé publique du destinataire, s'il est demandé:

[250 User-Certificate: <certificat de clé publique codé au format Base64>]

250 OK

Si l'adresse électronique ne peut être confirmée, les messages d'erreur ci-après peuvent être envoyés:

503 Error: need MAIL FROM command

est envoyé si le message est envoyé hors séquence.

452 Error: too many recipients

est envoyé si un trop grand nombre de destinataires est spécifié.

501-6.1.1 Syntax: RCPT TO <adresse>

est envoyé en cas d'erreur de syntaxe dans l'adresse électronique du destinataire.

501-6.1.2 Syntax: RCPT TO <adresse> Error in parameters: <paramètres>

est envoyé si le paramètre après l'adresse électronique n'est pas reconnu.

550-5.1.1 <adresse électronique> Invalid email address.

est envoyé si l'adresse électronique n'existe pas.

8.11 Demande d'envoi de l'ENVELOPPE

Le format ci-après est utilisé par l'expéditeur pour demander au serveur Cmail la permission d'envoyer des données.

DATA

8.12 Prêt à recevoir l'ENVELOPPE

Le message ci-après est envoyé si le serveur Cmail est prêt à recevoir des données:

354 Start mail input; end with <CRLF>.<CRLF>

Le message ci-après est envoyé lorsque la commande MAIL FROM n'a pas été envoyée:

503 Error: need MAIL FROM command

Le message ci-après est envoyé lorsque la commande RCPT TO n'a pas été envoyée:

503 Error: need RCPT TO command

Le message ci-après est envoyé lorsque la commande DELV n'a pas été envoyée:

503 Error: need DELV command

8.13 ENVELOPPE

Le client doit:

1) générer une clé de chiffrement symétrique aléatoire (RSCK), par exemple au moyen de la norme de cryptage évoluée AES-256;
2) chiffrer le corps du message et les pièces jointes, le cas échéant, en utilisant cette clé;
3) construire un message MIME comprenant une partie nommée ENVELOPE qui contient le message chiffré (voir [IETF RFC 2045]);
4) finir le message avec <CR><LF>.<CR><LF>; et
5) envoyer le message MIME.

8.14 Notification de dépôt signée par le serveur

250 Notice-of-deposit:

<notification de dépôt signée par le serveur Cmail codée au format base64>

250 Ok

Le serveur génère une notification de dépôt contenant des informations relatives à l'enveloppe (identifiant de l'enveloppe, type de remise et hachage MIME) et la signe avec sa clé privée.

8.15 Notification de dépôt signée par l'expéditeur et le serveur

L'expéditeur doit:

1) décoder la notification de dépôt reçue;
2) construire l'interrogation pour chaque destinataire;
3) signer la notification de dépôt signée par le serveur en utilisant sa propre clé privée;
4) coder le résultat au format base64; et
5) le transmettre au serveur Cmail en utilisant:

DEPO <notification de dépôt codée au format base64>

L'interrogation est définie dans la Figure A.6.

L'interrogation contient les champs **SecretQuestion** et **CipherEnvelopeKey** ainsi que le certificat de clé publique du destinataire.

Le champ **SecretQuestion** est composé d'un champ **Request** et d'un champ **Response**.

La requête peut contenir un champ **RandomNumber**. La réponse contient le champ **AlgorithmIdentifier**, que l'expéditeur doit recalculer pour recevoir l'ENVELOPPE. Ce champ **AlgorithmIdentifier** identifie l'algorithme utilisé pour calculer le hachage. L'interrogation consiste tout d'abord à retrouver la clé de chiffrement RSCK, chiffrée par la clé publique du destinataire, puis à effectuer la concaténation du champ **RandomNumber** et de la clé RSCK, et à calculer le hachage pour construire la réponse.

Exemple d'interrogation en langage de balisage extensible (XML):

```
<Entity EmailAddress="john.doe@example.org" Type="to">
  <SecretQuestion>
    <Request RandomNumber="30987497498789739837"/>
   <Response AlgorithmIdentifier="2.16.840.1.101.3.4.2.1"
Encoding="base64">5mYZWhtl0yxBa/wl7VLiiQ=</response>
  </SecretQuestion>
  <CipherEnvelopeKey Algorithm="AES" CipheredKey="RSA" Encoding="base64-DER"
KeySize="256">UjBg…b1PHDOOM4IFnTpzHn9TQ==</cipherEnvelopeKey>
  <Certificate
Encoding="base64">MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AM…sdjn7VDBlb+WS10j2rJcAHHsUyr…
/gy7</Certificate>
</Entity>r
```

NOTE 1 – Cette interrogation pourrait utiliser les règles de codage distinctives (DER) de la notation de syntaxe abstraite numéro un (ASN.1).

NOTE 2 – Le serveur n'est pas capable de recalculer le hachage étant donné qu'il ne connaît pas la clé de chiffrement. Néanmoins, seul le serveur connaît le résultat attendu du calcul du hachage.

NOTE 3 – Pendant l'interrogation du destinataire, le serveur envoie uniquement la question secrète et attend la réponse du destinataire.

8.16 ENVELOPPE entre serveurs Cmail

Le message défini au § 8.13 est retransmis à un autre serveur Cmail uniquement si l'expéditeur et le destinataire sont rattachés à des serveurs Cmail différents (voir l'élément m16 dans la Figure 1).

SEND EVLP <message MIME>

8.17 Notification de transit signée par les serveurs Cmail

Le format ci-après doit être utilisé:

250 Notice-of-transit:

<notification de transit codée au format base64>

Le message ci-après est envoyé si le serveur Cmail reçoit une notification de transit:

250 Ok

Le message ci-après est envoyé lorsque la notification de transit est erronée:

503 Error: incorrect Notice-of-transit

La notification de transit est construite par le serveur Cmail qui a reçu l'ENVELOPPE.

Ce serveur Cmail génère une notification de transit comprenant des informations relatives à l'enveloppe (identifiant de l'enveloppe, type de remise et hachage MIME) et la signe avec sa clé privée. Cette notification est identique à la notification de dépôt.

8.18 Notification de transit signée

Le serveur Cmail de l'expéditeur doit:

1) décoder la notification de transit reçue;
2) signer la notification de transit signée par le serveur en utilisant sa propre clé privée;
3) coder le résultat au format base64; et
4) le transmettre au serveur Cmail en utilisant:

   250 Signed-notice-of-transit:

   <notification de transit signée codée au format base64>

   250 Signed-notice-of-deposit:

   <notification de transit signée codée au format base64>

   250 Ok

9 Protocole de bureau de poste certifié (CPOP)

Les paragraphes 9.1 à 9.6 décrivent les fonctions p1 à p6 de la Figure 1.

9.1 Demande de messages en attente

L'information concernant les messages en attente est communiquée grâce à la procédure définie au paragraphe 5 sous la commande LIST dans [IETF RFC 1939] avec un paramètre supplémentaire. Pour chaque ligne indiquant un message en attente, on ajoute ce paramètre supplémentaire précisant le type de remise s'il ne s'agit pas d'un courrier électronique standard (voir la fonction p1 dans la Figure 1). Par exemple:

C: LIST

S: +OK 2 messages (320 octets)

S: 1 120

S: 2 200 CertifiedMail

S: .

Cette procédure comprend en outre l'extraction de tous les courriers électroniques standards en ne laissant que les messages étiquetés avec le type de remise sur le serveur Cmail.

9.2 Interrogation du destinataire et notification de réception signée par le serveur

Pour les messages étiquetés avec un type de remise, la commande RETR n'extrait pas le message mais l'interrogation et la notification de réception signée par le serveur codée au format base64. Le client vérifie la signature numérique et le certificat de l'expéditeur contenu dans la notification de réception.

Exemple:

C: RETR 2

Le message ci-après est envoyé si le serveur Cmail envoie la notification de réception

S: +OK 200 octets

S: <le serveur Cmail envoie la notification de réception comprenant l'interrogation>

S: .

Le message ci-après est envoyé lorsque le serveur ne peut pas envoyer la notification de réception:

503 Error: impossible to send Notice-of-reception

Le serveur Cmail trouve dans la notification de dépôt le noeud **Entity** rattaché au destinataire. Le serveur Cmail copie alors ce noeud dans la notification de réception et supprime le contenu du noeud **Response** figurant dans le noeud **Entity**.

Exemple de noeud dans la notification de dépôt:

```
<Entity EmailAddress="john.doe@example.org" Type="to">
  <SecretQuestion>
    <Request RandomNumber="30987497498789739837"/>
   <Response AlgorithmIdentifier="2.16.840.1.101.3.4.2.1"
Encoding="base64">5mYZWhtl0yxBa/wl7VLiiQ=</response>
  </SecretQuestion>
  <CipherEnvelopeKey Algorithm="AES" CipheredKey="RSA" Encoding="base64-DER"
KeySize="256">UjBg…b1PHDOOM4IFnTpzHn9TQ==</cipherEnvelopeKey>
  <Certificate
Encoding="base64">MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AM…sdjn7VDBlb+WS10j2rJcAHHsUyr…
/gy7</Certificate>
</Entity>
```

NOTE 1 – Cette interrogation pourrait utiliser le codage DER de la notation ASN.1.

Et ce même noeud copié dans la notification de réception:

```
<Entity EmailAddress="john.doe@example.org" Type="to">
  <SecretQuestion>
    <Request RandomNumber="30987497498789739837"/>
   <Response AlgorithmIdentifier="2.16.840.1.101.3.4.2.1" Encoding="base64" />
  </SecretQuestion>
  <CipherEnvelopeKey Algorithm="AES" CipheredKey="RSA" Encoding="base64-DER"
KeySize="256">UjBg…b1PHDOOM4IFnTpzHn9TQ==</cipherEnvelopeKey>
  <Certificate
Encoding="base64">MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AM…sdjn7VDBlb+WS10j2rJcAHHsUyr…
/gy7</Certificate>
</Entity>
```

NOTE 2 – Cette interrogation pourrait utiliser le codage DER de la notation ASN.1.

9.3 Réponse à l'interrogation et notification de réception signée par le destinataire et le serveur

Le destinataire doit:

1) décoder la notification de réception reçue;
2) extraire la clé RSCK;
3) calculer la réponse à l'interrogation;
4) signer la notification de réception signée par le serveur en utilisant sa propre clé privée;
5) coder le résultat au format base64; et
6) le transmettre au serveur Cmail en utilisant:

CHLG RESP <réponse à l'interrogation et notification de réception signée par le destinataire et le serveur>

Le destinataire déchiffre le message comme suit:

```
<Entity EmailAddress="john.doe@example.org" Type="to">
  <SecretQuestion>
    <Request RandomNumber="30987497498789739837"/>
   <Response AlgorithmIdentifier="2.16.840.1.101.3.4.2.1"
Encoding="base64"></response>
  </SecretQuestion>
```

```
  <CipherEnvelopeKey Algorithm="AES" CipheredKey="RSA" Encoding="base64-DER"
KeySize="256">UjBg…b1PHDOOM4IFnTpzHn9TQ==</cipherEnvelopeKey>
  <Certificate
Encoding="base64">MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AM…sdjn7VDBlb+WS10j2rJcAHHsUyr…
/gy7</Certificate>
</Entity>
```

NOTE – Cette interrogation pourrait utiliser le codage DER de la notation ASN.1.

Le destinataire récupère la clé RSCK à l'aide de sa clé privée en déchiffrant le contenu du noeud **`CipherEnvelopeKey`**. Le destinataire effectue ensuite la concaténation du champ **`RandomNumber`** et de la clé RSCK, effectue le hachage en utilisant le champ **`AlgorithmIdentifier`** défini et obtient le résultat pour le champ **`SecretQuestion`**.

Le destinataire copie ce résultat dans la notification de réception signée, la signe et l'envoie au serveur Cmail.

9.4 ENVELOPPE

Si l'interrogation est OK, le serveur Cmail envoie l'ENVELOPPE de la même façon qu'il a envoyé le résultat de la commande RETR. Le destinataire a maintenant le message et la clé permettant de l'ouvrir.

Le message ci-après est envoyé lorsque le serveur ne peut pas envoyer l'ENVELOPPE:

503 Error: impossible to send ENVELOPE

9.5 Notification de réception signée par le destinataire et le serveur entre serveurs Cmail (facultatif)

Ce message est envoyé uniquement si l'expéditeur et le destinataire sont rattachés à des serveurs Cmail différents.

SEND NORP <notification de réception signée par le destinataire et le serveur codée au format base64>

9.6 Notification de réception signée par le destinataire et le serveur

Ce message est envoyé uniquement si l'expéditeur et le destinataire sont rattachés à des serveurs Cmail différents.

SEND NORP <notification de réception signée par le destinataire et le serveur codée au format base64>

Annexe A

Notifications utilisant la définition de schéma XML (XSD)

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

La présente Annexe contient les spécifications des notifications utilisant la définition de schéma XML (XSD) spécifiée dans [XSD]. Une instance de communication est codée au format XML défini dans [XML] et doit être conforme aux spécifications XSD données dans la présente Annexe.

A.1 Aperçu du format XSD

Voir les Figures A.1 à A.10.

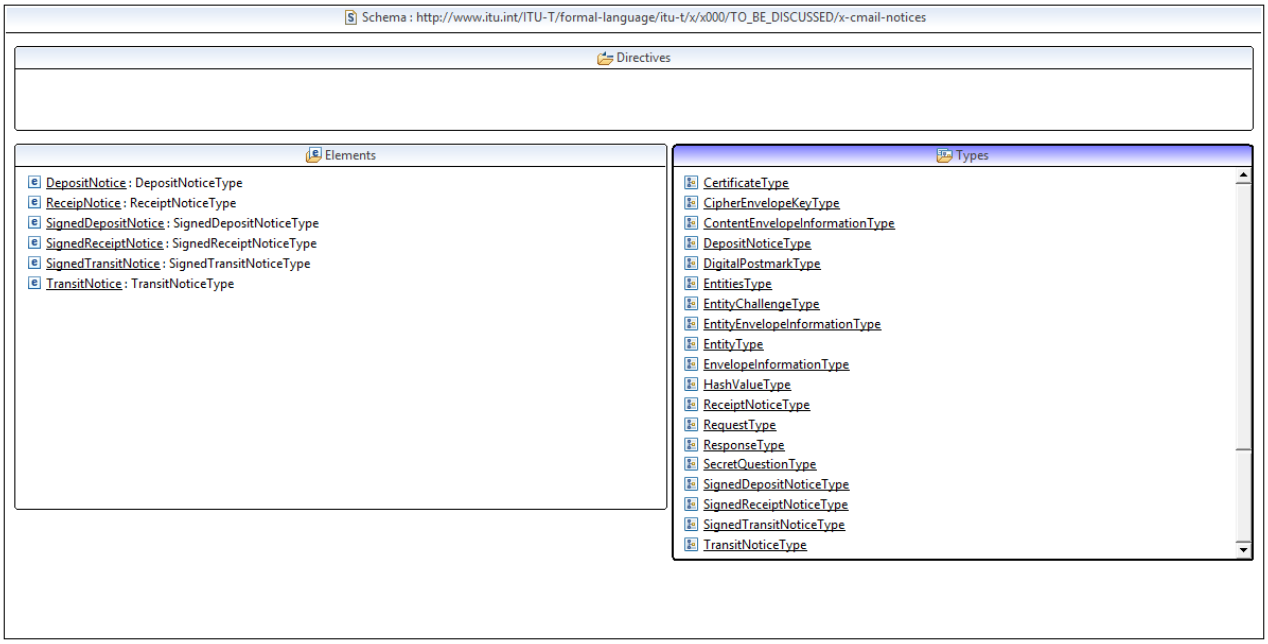

Figure A.1 – Eléments et liste des types

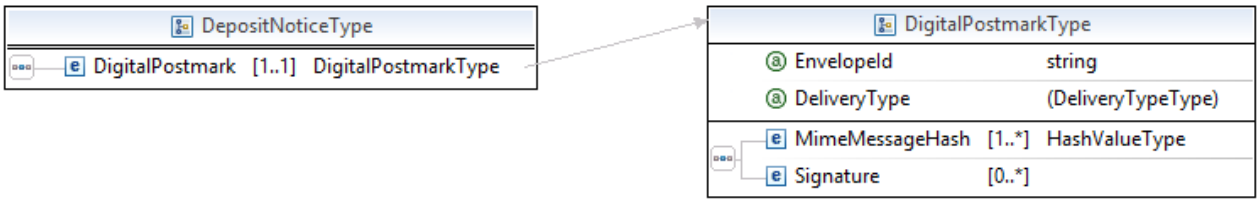

Figure A.2 – Notification de dépôt

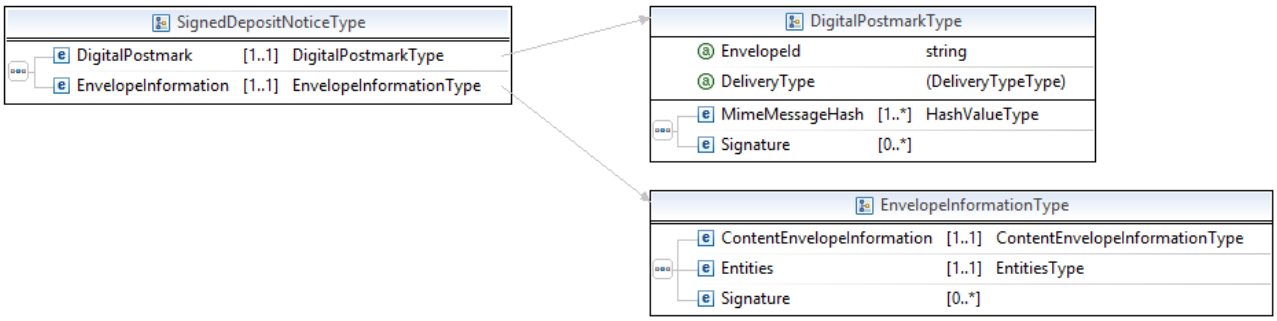

Figure A.3 – Notification de dépôt signée

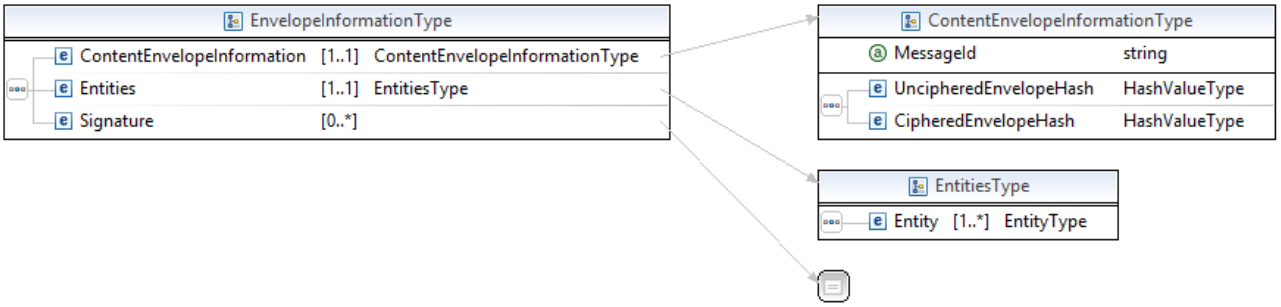

Figure A.4 – Type d'informations d'enveloppe

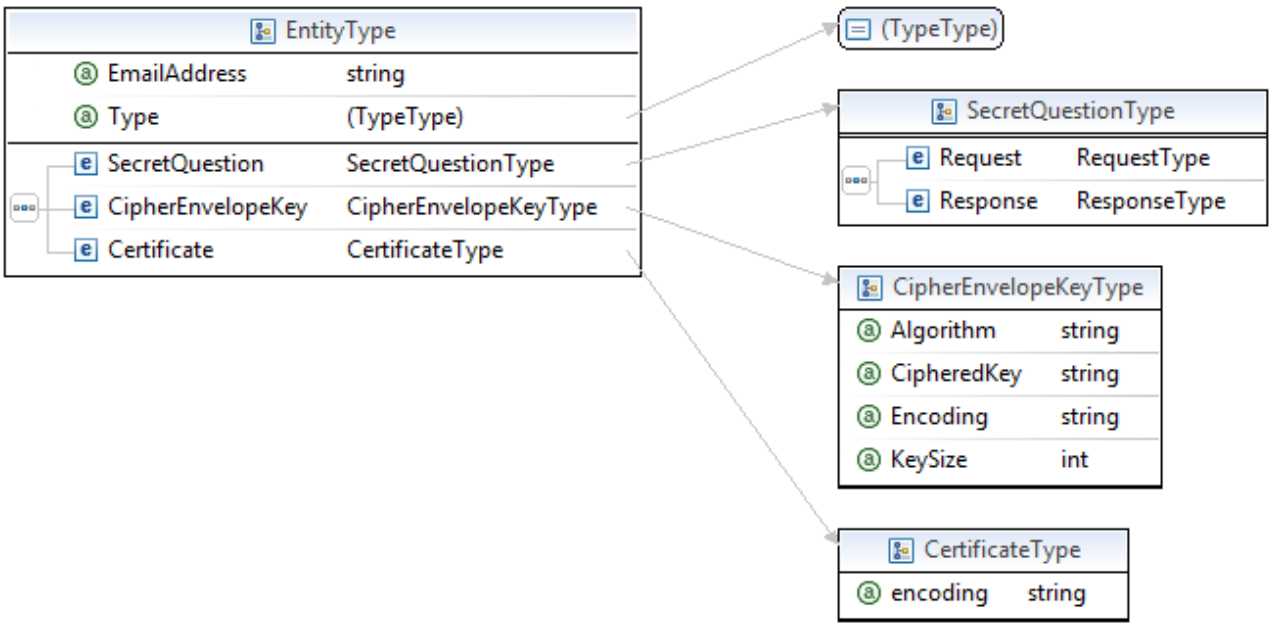

Figure A.5 – Type d'entité

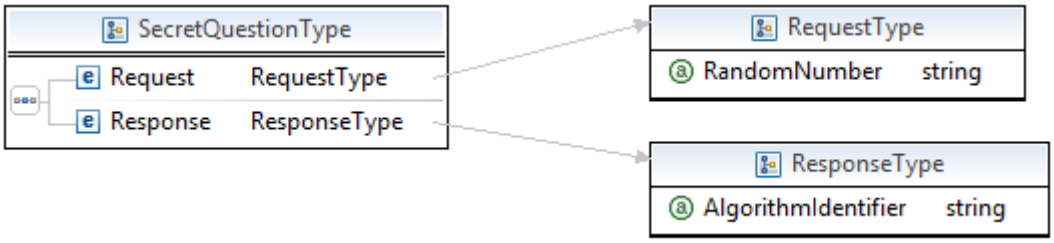

Figure A.6 – Interrogation

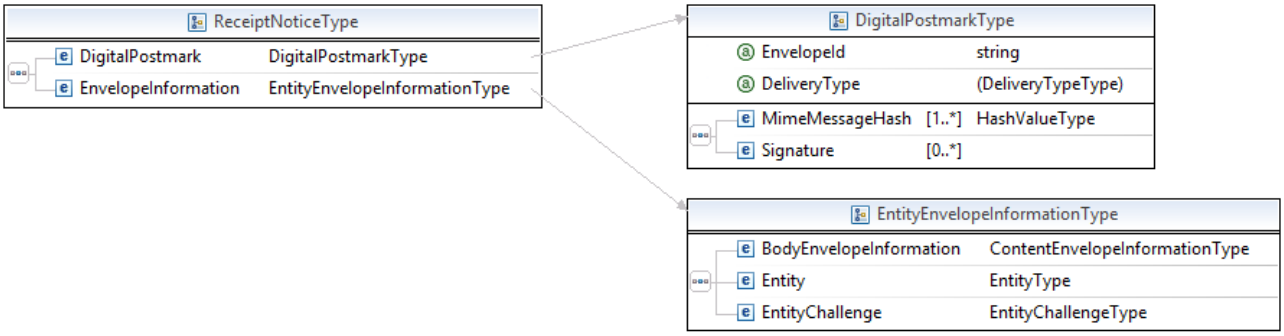

Figure A.7 – Notification de réception

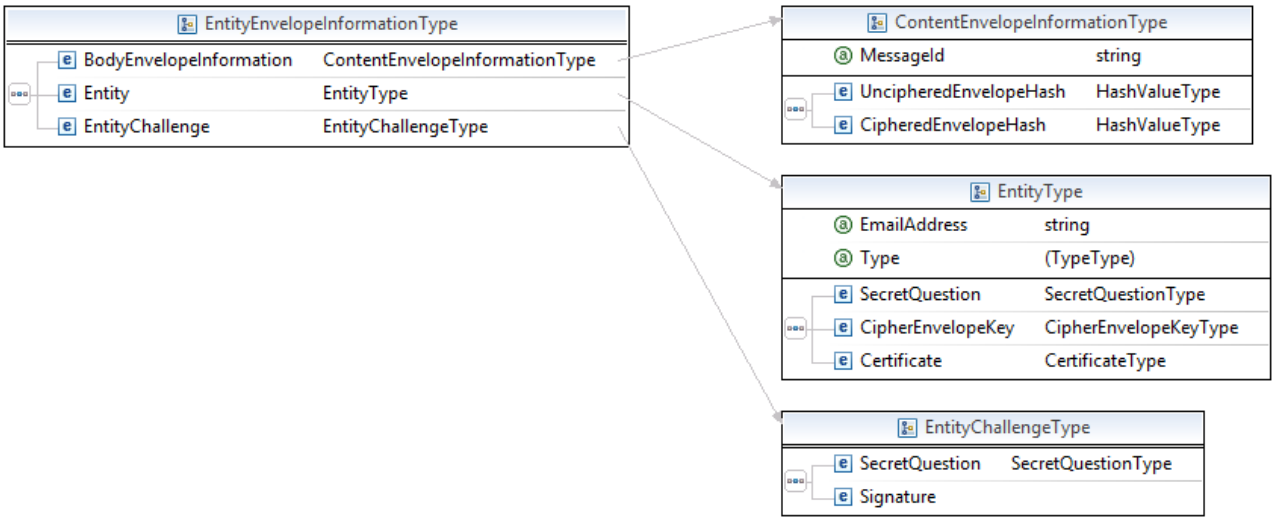

Figure A.8 – Réponse du destinataire à l'interrogation

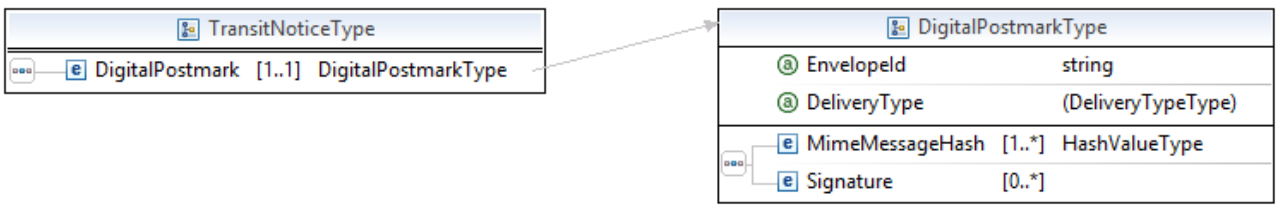

Figure A.9 – Notification de transit

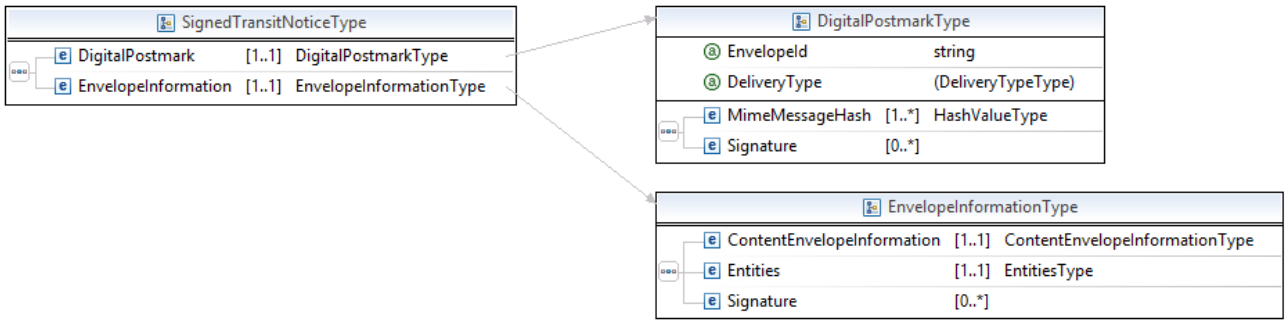

Figure A.10 – Notification de transit signée

A.2 Spécifications formelles des notifications au format XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.1341/x-cmail-notices"
     elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.itu.int/xml-namespace/itu-t/x.1341/x-cmail-notices"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

  <import namespace="http://www.w3.org/2009/xmldsig11#"
    schemaLocation="http://www.w3.org/TR/xmldsig-core1/xmldsig11-schema.xsd" />
  <import namespace="http://www.w3.org/2009/xmldsig-properties"
    schemaLocation="http://www.w3.org/TR/xmldsig-properties/xmldsig-
properties.xsd" />

  <import namespace=http://www.w3.org/2000/09/xmldsig#
    schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd" />

  <element name="DepositNotice" type="tns:DepositNoticeType"></element>
  <element name="SignedDepositNotice"
type="tns:SignedDepositNoticeType"></element>
  <element name="TransitNotice" type="tns:TransitNoticeType"></element>
  <element name="SignedTransitNotice"
type="tns:SignedTransitNoticeType"></element>
  <element name="ReceipNotice" type="tns:ReceiptNoticeType"></element>
  <element name="SignedReceiptNotice"
type="tns:SignedReceiptNoticeType"></element>

  <complexType name="DigitalPostmarkType">
    <sequence>
      <element name="MimeMessageHash" type="tns:HashValueType"
        maxOccurs="unbounded" minOccurs="1">
      </element>
      <element name="Signature" type="ds:SignatureType"
        maxOccurs="unbounded" minOccurs="0">
      </element>
    </sequence>
    <attribute name="EnvelopeId" type="string" use="required"></attribute>
    <attribute name="DeliveryType" use="required">
      <simpleType>
        <restriction base="string">
          <enumeration value="CertifiedMail"></enumeration>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>

  <complexType name="EnvelopeInformationType">
    <sequence>
      <element name="ContentEnvelopeInformation"
```

```
      type="tns:ContentEnvelopeInformationType" maxOccurs="1" minOccurs="1">
    </element>
    <element name="Entities" type="tns:EntitiesType"
      maxOccurs="1" minOccurs="1">
    </element>
    <element name="Signature" type="ds:SignatureType"
      maxOccurs="unbounded" minOccurs="0">
    </element>
  </sequence>
</complexType>

<complexType name="ContentEnvelopeInformationType">
  <sequence>
    <element name="UncipheredEnvelopeHash" type="tns:HashValueType"></element>
    <element name="CipheredEnvelopeHash" type="tns:HashValueType"></element>
  </sequence>
  <attribute name="MessageId" type="string"></attribute>
</complexType>

<complexType name="SecretQuestionType">
  <sequence>
    <element name="Request" type="tns:RequestType"></element>
    <element name="Response" type="tns:ResponseType"></element>
  </sequence>
</complexType>

<complexType name="EntityType">
  <sequence>
    <element name="SecretQuestion" type="tns:SecretQuestionType"></element>
    <element name="CipherEnvelopeKey"
      type="tns:CipherEnvelopeKeyType">
    </element>
    <element name="Certificate" type="tns:CertificateType"></element>
  </sequence>
  <attribute name="EmailAddress" type="string" use="required">
    <annotation>
      <documentation>Email address has to be in RFC 822format</documentation>
    </annotation></attribute>
  <attribute name="Type" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="from"></enumeration>
        <enumeration value="to"></enumeration>
        <enumeration value="cc"></enumeration>
        <enumeration value="transit"></enumeration>
      </restriction>
    </simpleType>
  </attribute>
</complexType>

<complexType name="CipherEnvelopeKeyType">
  <attribute name="Algorithm" type="string"></attribute>
  <attribute name="CipheredKey" type="string"></attribute>
  <attribute name="Encoding" type="string"></attribute>
  <attribute name="KeySize" type="int"></attribute>
</complexType>

<complexType name="CertificateType">
  <attribute name="encoding" type="string"></attribute>
</complexType>

<complexType name="EntitiesType">
  <sequence>
    <element name="Entity" type="tns:EntityType"
```

```
      maxOccurs="unbounded" minOccurs="1">
    </element>
  </sequence>
</complexType>

<complexType name="SignedDepositNoticeType">
  <sequence>
    <element name="DigitalPostmark" type="tns:DigitalPostmarkType"
      maxOccurs="1" minOccurs="1">
    </element>
    <element name="EnvelopeInformation"
      type="tns:EnvelopeInformationType" maxOccurs="1" minOccurs="1">
    </element>
  </sequence>
</complexType>

<complexType name="DepositNoticeType">
  <sequence>
    <element name="DigitalPostmark" type="tns:DigitalPostmarkType"
      maxOccurs="1" minOccurs="1">
    </element>
  </sequence>
</complexType>

<complexType name="TransitNoticeType">
  <sequence>
    <element name="DigitalPostmark" type="tns:DigitalPostmarkType"
      maxOccurs="1" minOccurs="1">
    </element>
  </sequence>
</complexType>

<complexType name="SignedTransitNoticeType">
  <sequence>
    <element name="DigitalPostmark" type="tns:DigitalPostmarkType"
      maxOccurs="1" minOccurs="1">
    </element>
    <element name="EnvelopeInformation"
      type="tns:EnvelopeInformationType" maxOccurs="1" minOccurs="1">
    </element>
  </sequence>
</complexType>

<complexType name="ReceiptNoticeType">
  <sequence>
    <element name="DigitalPostmark"
      type="tns:DigitalPostmarkType">
    </element>
    <element name="EnvelopeInformation"
      type="tns:EntityEnvelopeInformationType">
    </element>
  </sequence>
</complexType>

<complexType name="SignedReceiptNoticeType">
  <sequence>
    <element name="DigitalPostmark"
      type="tns:DigitalPostmarkType">
    </element>
    <element name="EnvelopeInformation"
      type="tns:EntityEnvelopeInformationType">
    </element>
  </sequence>
</complexType>
```

```
  <complexType name="HashValueType">
    <attribute name="AlgorithmOID">
      <simpleType>
        <restriction base="string">
          <enumeration value="1.3.14.3.2.26"></enumeration>
          <enumeration value="2.16.840.1.101.3.4.2.1"></enumeration>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>

  <complexType name="EntityEnvelopeInformationType">
    <sequence>
      <element name="BodyEnvelopeInformation"
type="tns:ContentEnvelopeInformationType">
      </element>
      <element name="Entity" type="tns:EntityType"></element>
      <element name="EntityChallenge" type="tns:EntityChallengeType"></element>
    </sequence>
  </complexType>

  <complexType name="EntityChallengeType">
    <sequence>
      <element name="SecretQuestion" type="tns:SecretQuestionType"></element>
      <element name="Signature" type="ds:SignatureType"></element>
    </sequence>
  </complexType>

  <complexType name="RequestType">
    <attribute name="RandomNumber" type="string"></attribute>
  </complexType>

  <complexType name="ResponseType">
    <attribute name="AlgorithmIdentifier" type="string"></attribute>
  </complexType>

</schema>
```

Annexe B

Notifications utilisant la notation ASN.1

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

La présente annexe contient les spécifications des notifications utilisant la notation de syntaxe abstraite numéro un (ASN.1) spécifiée dans [UIT-T X.680]. Ces notifications peuvent être codées selon les règles de codage distinctives (DER) de la notation ASN.1 définies dans [UIT-T X.690] ou selon les règles de codage XML étendues (EXTENDED-XER) spécifiées dans [UIT-T X.693]. Dans le dernier cas, les notifications XML résultant de ce codage sont identiques aux notifications XML générées conformément aux spécifications XDS définies dans l'Annexe A.

```
CMAIL {itu-t(0) recommendation(0) x(24) cmail(1341) asn1Module(1) cmail(1)}
DEFINITIONS AUTOMATIC TAGS::=
BEGIN

IMPORTS String
     FROM XSDv2 {joint-iso-itu-t asn1(1) specification(0) modules(0)
          xsd-module(2) version2(2)};

DepositNotice      ::= DepositNoticeType

SignedDepositNotice     ::= SignedDepositNoticeType

TransitNotice      ::= TransitNoticeType

SignedTransitNotice     ::= SignedTransitNoticeType

ReceiptNotice      ::= ReceiptNoticeType

SignedReceiptNotice     ::= SignedReceiptNoticeType

DigitalPostmarkType     ::= SEQUENCE {
     mimeMessageHash    SEQUENCE (SIZE(1..MAX)) OF
          mimeMessageHash HashValueType,
     signature SEQUENCE (SIZE(0..MAX)) OF
          signature SignatureType,
     envelopeId     String,
     deliveryType   ENUMERATED {
          certifiedMail,
          ...
          }
     }

EnvelopeInformationType ::= SEQUENCE {
     contentEnvelopeInformation   ContentEnvelopeInformationType,
     entities            EntitiesType,
     signature      SEQUENCE (SIZE(0..MAX)) OF
          signature SignatureType
     }

ContentEnvelopeInformationType    ::= SEQUENCE {
     uncipheredEnvelopeHash  HashValueType,
     cipheredEnvelopeHash    HashValueType,
     messageId      String
     }

SecretQuestionType ::= SEQUENCE {
     request        RequestType,
```

```
     response  ResponseType
     }

EntityType     ::= SEQUENCE {
     secretQuestion      SecretQuestionType,
     cipheredEnvelopeKey     CipheredEnvelopeKeyType,
     certificate         CertificateType,
     emailAddress        String
          (CONSTRAINED BY
          {-- "Email address has to be in IETF RFC 822 format --}),
     type ENUMERATED {
          from,
          to,
          cc,
          transit
          }
     }

CipheredEnvelopeKeyType ::= SEQUENCE {
     algorithm String,
     cipherededKey  String,
     encoding  String,
     keySize        String
     }

CertificateType     ::= SEQUENCE {
     encoding  String
     }

EntitiesType   ::= SEQUENCE {
     entity         SEQUENCE(SIZE(1..MAX)) OF entity EntityType
     }

SignedDepositNoticeType ::= SEQUENCE {
     digitalPostmark         DigitalPostmarkType,
     envelopeInformation     EnvelopeInformationType
     }

DepositNoticeType  ::= SEQUENCE {
     digitalPostmark         DigitalPostmarkType
     }

TransitNoticeType  ::= SEQUENCE {
     digitalPostmark         DigitalPostmarkType
     }

SignedTransitNoticeType ::= SEQUENCE {
     digitalPostmark         DigitalPostmarkType,
     envelopeInformation     EnvelopeInformationType
     }

ReceiptNoticeType  ::= SEQUENCE {
     operatorPostmark   DigitalPostmarkType
     }

SignedReceiptNoticeType ::= SEQUENCE {
     operatorPostmark   DigitalPostmarkType,
     envelopeInformation     EntityEnvelopeInformationType
     }

HashValueType ::= SEQUENCE {
     algorithmOID  ENUMERATED {
          sha-1,
          sha-256
```

```
          }
     }

EntityEnvelopeInformationType    ::= SEQUENCE {
     bodyEnvelopeInformation ContentEnvelopeInformationType,
     entity             EntityType,
     entityChallenge         EntityChallengeType
     }

EntityChallengeType     ::= SEQUENCE {
     secretQuestion _SecretQuestionType,
     signature SignatureType
     }

RequestType    ::= SEQUENCE {
     randomNumer    String
     }

ResponseType   ::= SEQUENCE {
     algorithmIdentifier     String
     }

SignatureType ::= String

ENCODING-CONTROL XER
     GLOBAL-DEFAULTS MODIFIED-ENCODINGS
     [NAME AS CAPITALIZED] DigitalPostmarkType.mimeMessageHash
     [UNTAGGED] DigitalPostmarkType.mimeMessageHash
     [NAME AS CAPITALIZED] DigitalPostmarkType.signature.*
     [UNTAGGED] DigitalPostmarkType.signature
     [NAME AS CAPITALIZED] DigitalPostmarkType.envelopeId
     [ATTRIBUTE] DigitalPostmarkType.envelopeId
     [NAME AS CAPITALIZED] DigitalPostmarkType.deliveryType
     [ATTRIBUTE] DigitalPostmarkType.deliveryType
     [TEXT AS CAPITALIZED] DigitalPostmarkType.delivetyType:certifiedMail
     [NAME AS CAPITALIZED] EnvelopeInformationType.contentEnvelopeInformation
     [NAME AS CAPITALIZED] EnvelopeInformationType.entities
     [NAME AS CAPITALIZED] EnvelopeInformationType.signature
     [UNTAGGED] EnvelopeInformationType.signature
     [NAME AS CAPITALIZED]
          ContentEnvelopeInformationType.uncipheredEnvelopeHash
     [NAME AS CAPITALIZED]
          ContentEnvelopeInformationType.cipheredEnvelopeHash
     [NAME AS CAPITALIZED] ContentEnvelopeInformationType.messageId
     [ATTRIBUTE] ContentEnvelopeInformationType.messageId
     [NAME AS CAPITALIZED] SecretQuestionType.request
     [NAME AS CAPITALIZED] SecretQuestionType.response
     [NAME AS CAPITALIZED] EntityType.secretQuestion
     [NAME AS CAPITALIZED] EntityType.cipheredEnvelopeKey
     [NAME AS CAPITALIZED] EntityType.certificate
     [NAME AS CAPITALIZED] EntityType.emailAddress
     [ATTRIBUTE] EntityType.emailAddress
     [NAME AS CAPITALIZED] EntityType.type
     [ATTRIBUTE] EntityType.type
     [NAME AS CAPITALIZED] CipheredEnvelopeKeyType.algorithm
     [ATTRIBUTE] CipheredEnvelopeKeyType.algorithm
     [NAME AS CAPITALIZED] CipheredEnvelopeKeyType.cipheredKey
     [ATTRIBUTE] CipheredEnvelopeKeyType.cipheredKey
     [NAME AS CAPITALIZED] CipheredEnvelopeKeyType.encoding
     [ATTRIBUTE] CipheredEnvelopeKeyType.encoding
     [NAME AS CAPITALIZED] CipheredEnvelopeKeyType.keysize
     [ATTRIBUTE] CipheredEnvelopeKeyType.keysize
     [NAME AS CAPITALIZED] CertificateType.encoding
     [ATTRIBUTE] CertificateType.encoding
```

```
    [UNTAGGED] EntitiesType.entity
    [NAME AS CAPITALIZED] EntitiesType.entity.*
    [NAME AS CAPITALIZED] SignedDepositNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] SignedDepositNoticeType.envelopeInformation
    [NAME AS CAPITALIZED] DepositNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] TransitNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] SignedTransitNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] SignedTransitNoticeType.envelopeInformation
    [NAME AS CAPITALIZED] ReceiptNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] SignedReceiptNoticeType.digitalPostmark
    [NAME AS CAPITALIZED] SignedReceiptNoticeType.envelopeInformation
    [NAME AS CAPITALIZED] HashValueType.algorithmOID
    [ATTRIBUTE] HashValueType.algorithmOID
    [TEXT AS "1.3.14.3.2.26"] HashValueType.algorithmOID:sha-1
    [TEXT AS "2.16.840.1.101.3.4.2.1"] HashValueType.algorithmOID:sha-256
    [NAME AS CAPITALIZED]
        EntityEnvelopeInformationType.BodyEnvelopeInformation
    [NAME AS CAPITALIZED]
        EntityEnvelopeInformationType.entityChallenge
    [NAME AS CAPITALIZED] EntityChallengeType.secretQuestion
    [NAME AS CAPITALIZED] EntityChallengeType.signature
    [NAME AS CAPITALIZED] RequestType.randomNumber
    [ATTRIBUTE] RequestType.randomNumber
    [NAME AS CAPITALIZED] ResponseType.algorithmIdentifier
    [ATTRIBUTE] ResponseType.algorithmIdentifier

END
```

Annexe C

Exigences concernant les éléments de l'infrastructure de clé publique

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

C.1 Introduction

La présente annexe donne les exigences concernant les certificats de clé publique délivrés aux serveurs et aux clients Cmail.

C.2 Certificat de clé publique d'entité finale délivré à un serveur Cmail

Le contenu d'un certificat de clé publique d'entité finale délivré à un serveur Cmail est le suivant:

a) La version 3 sera spécifiée.

b) L'autorité de certification générera des numéros de série non séquentiels.

c) Le champ de sujet comprendra un nom distinctif d'annuaire avec un élément unique utilisant le type d'attribut **dnsName** défini dans [UIT-T X.520]. La valeur sera un nom enregistré dans le système de noms de domaine (DNS).

d) L'extension d'autre nom de sujet sera présente avec deux éléments:

- l'un des éléments sera le champ **rfc822Name**, qui sera l'adresse électronique de l'administrateur du serveur Cmail;
- l'autre élément sera le champ **directoryName**, qui comprendra un nom distinctif avec les éléments suivants:
  - l'attribut **countryName** sera présent et comprendra le code à trois lettres (alpha-3) de [ISO 3166-1];
  - l'attribut **organizationName** sera présent et comprendra le nom de confiance de l'organisation gérant le serveur Cmail;
  - l'attribut **streetAddress** sera présent et comprendra le nom de la rue et le numéro du bâtiment;
  - l'attribut **localityName** sera présent et comprendra le nom de la localité;
  - l'attribut **stateOrProvinceName** sera présent s'il est nécessaire pour l'identification unique. Dans le cas contraire, il sera absent;
  - l'attribut **postalCode** sera présent et comprendra le code postal de la localité.

e) L'extension **certificatePolicies** sera présente et comprendra au moins l'identificateur d'objet **{itu-t(0) recommendation(0) x(24) cmail(1341) certificatePolicy(2) cmailServer(1)}** pour signaler que le certificat de clé publique est délivré conformément à la présente Recommandation.

C.3 Certificat de clé publique d'entité finale délivré à un client Cmail

Le contenu d'un certificat de clé publique d'entité finale délivré à un serveur Cmail est le suivant:

a) La version 3 sera spécifiée.

b) L'autorité de certification génèrera des numéros de série non séquentiels.

c) Le champ de sujet comprendra un nom distinctif d'annuaire composé comme suit:

- l'attribut **surname** sera présent si le client est un particulier, mais sera absent si le client est une organisation;
- l'attribut **givenName** sera présent si l'attribut **surname** est présent. Dans le cas contraire, il sera absent;

– l'attribut **initials** pourra être présent si l'attribut **surname** est présent. Dans le cas contraire, il sera absent;

– l'attribut **generationQualifier** pourra être présent si l'attribut **surname** est présent. Dans le cas contraire, il sera absent;

– l'attribut **organizationName** sera présent si le client n'est pas une personne privée. Dans le cas contraire, il sera absent. S'il est présent, il comprendra le nom de confiance de l'organisation à laquelle le client appartient;

– l'attribut **streetAddress** sera présent et comprendra le nom de la rue et le numéro du bâtiment;

– l'attribut **localityName** sera présent et comprendra le nom de la localité;

– l'attribut **stateOrProvinceName** sera présent s'il est nécessaire pour l'identification unique. Dans le cas contraire, il sera absent;

– l'attribut **postalCode** sera présent et comprendra le code postal de la localité;

– l'attribut **countryCode3c** sera présent et comprendra le code à trois lettres (alpha-3) de [ISO 3166-1].

d) L'extension **subjectAltName** sera présente. Elle contiendra un élément comme indiqué ci-après:

– l'attribut **rfc822Name** comprendra l'adresse électronique de l'administrateur du serveur Cmail.

e) L'extension certificatePolicies sera présente et comprendra au moins l'identificateur d'objet **{itu-t(0) recommendation(0) x(24) cmail(1341) certificatePolicy(2) cmailClient(2)}** pour signaler que le certificat de clé publique est délivré conformément à la présente Recommandation.

C.4 Exigences de validation des informations

Avant de délivrer un certificat de clé publique, l'entité émettrice devra vérifier:

a) que le sujet (demandeur) est le titulaire enregistré du nom de domaine à faire figurer sur le certificat de clé publique;

b) que le sujet existe physiquement;

c) que le sujet existe sur le plan opérationnel (activité commerciale);

d) que le sujet est une entité reconnue de confiance;

e) les informations concernant le nom et l'adresse à faire figurer sur le certificat de clé publique;

f) que le champ **organizationName** à insérer dans le certificat de clé publique est un nom de confiance et reconnu qui identifie le sujet.

Annexe D

Exigences concernant la sécurité dans la couche transport (TLS)

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

[IETF RFC 5246] ou les versions ultérieures doivent être prises en charge.

Dans la négociation, ni le serveur Cmail ni le client ne doivent accepter une connexion en cas de tentative de négociation avec une version TLS antérieure à la version TLS 1.2.

Une application doit prendre en charge la suite de chiffrement suivante:

– TLS_DH_RSA_WITH_AES_256_CBC_SHA256

Annexe E

Identificateurs d'objet définis dans la présente Recommandation

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

La présente Recommandation définit les identificateurs d'objet suivants:

a) identificateur d'objet associé au module ASN.1:

{itu-t recommendation(0) x(24) cmail(1341) asn1module(0) cmail(1)}

b) identificateur d'objet utilisé par l'extension certificatePolicies d'un serveur Cmail:

{itu-t recommendation(0) x(24) cmail(1341) certificateProfile(2) cmailServer(1)}

c) identificateur d'objet utilisé par l'extension certificatePolicies d'un client Cmail:

{itu-t recommendation(0) x(24) cmail(1341) certificateProfile(2) cmailClient(2)}

Appendice I

Format de l'enveloppe et des notifications

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

Le présent appendice donne des exemples de codage des notifications.

I.1 Notification de dépôt

La notification de dépôt contient des informations sur l'expéditeur et l'enveloppe, et est cosignée par le serveur Cmail et l'expéditeur.

Elle constitue une preuve de dépôt pour l'expéditeur, qui peut l'utiliser en cas de litige.

La spécification formelle de la notification de dépôt figure dans l'Annexe A.

Exemple: fichier "1373360283931.deposit.notice"

```
Received: from localhost ([127.0.0.1])
        by begmeil
        with SMTP (SubEthaSMTP null) id HIWV8HF9
        for laura.prin@legalbox.com;
        Tue, 09 Jul 2013 10:58:14 +0200 (CEST)
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=depositNotice.xml

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+Cjxs
ZXR0ZXJEZXBvc2l0UG9zdG1hcms+CiAgPG9wZXJhdG9yUG9zdG1hcms+CiAgICA8ZW52ZWxvcElk
bG9wZWQtc2lnbmF0dXJlIi8+CiAgICAgICA8RyYW5zZm9ybXM+CiAgICAgICAgICA8RGln
...
ICAgPFJTQUtleVZhbHVlPgogICAgICAgICA8TW9kdWx1cz5tMkFSUURXUGJBMmgvMzJEQWs4
ICAgICAgICAgIDxFeHBvbmVudD5BUUFCPC9FeHBvbmVudD4KICAgICAgICAgIDwvUlNBS2V5VmFs
dWU+CiAgICAgICAgPC9LZXlWYWx1ZT4KICAgICAgICA8L0tleUJmZvPgogICAgPC9TaWduYXR1cmU+
CiAgPC9lbnZlbG9wZSW5mb3JtYXRpb24+CjwvbGV0dGVyRGVwb3NpdFBvc3RtYXJrPgo=
```

I.2 Notification de réception

La notification de réception contient des informations sur l'expéditeur, l'enveloppe et l'interrogation, permettant d'ouvrir l'enveloppe, et est cosignée par le serveur Cmail et le destinataire.

Elle constitue une preuve de réception pour l'expéditeur, qui peut l'utiliser en cas de litige.

La spécification formelle de la notification de réception figure dans l'Annexe A.

Exemple: fichier "1373360283931.laura.prin@legalbox.com.receipt.notice"

```
Received: from begmeil get hostname ([127.0.0.1])
        by localhost
        with SMTP (LegalBox POP Server v1.0) id HIWX27L5
        for laura.prin@legalbox.com;
        Tue, 09 Jul 2013 11:49:01 +0200 (CEST)
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=receiptNotice.xml

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+Cjxs
ZXR0ZXJEZXBvc2l0UG9zdG1hcms+CiAgPG9wZXJhdG9yUG9zdG1hcms+CiAgICA8ZW52ZWxvcEl
PjEzNzMzNjAyODM5MzE8L2VudmVsb3BJZD4KICAgIDxkZWxpdmVyeU1vZGU+Y2VydGlmaWVkTGV0
dGVyPC9kZWxpdmVyeU1vZGU+CiAgICA8bWltZU1lc3NhZ2VIYXNoPgogICAgICA8c2hhMT5hNTVk
ZDhmYWU0Mzg2M2VmYWRmMWY3ZjM3MmEwYmU1MmEwMGRhYTFkPC9zaGExPgogICAgPC9taW1lTWVz
...
MDkwSDl0NFVkTTdWVU92bjY3WlU2aTJvVSt3b3lGR2tYMDJ3YkVMM2pDYmpJCm5VR1BwUGpoT3Zo
dzNPTy9mYmhKVk13dkM2NXB1MTl1cnA2M05kS0tHNlBuNjZtQkVnUldxZ2cvTVBITmZmWkhrOXFs
WExSSXhETi8Kb0ZnS285RmI0NExlSzBnZ3VyblY2azNicm1TeGM1UnpYVWNxTzdwbldUN0FoNFl6
WXJJUHdYL1hjS1VqbXYxZi9JZjQ5VHVnWGtLcgpodklyOG9qUkdQcEdPdlB4cWR5QWNQR1BOUVRY
NFJrc29kSEVwdz09PC9Nb2R1bHVzPgogICAgICAgICAgICAgICA8RXhwb25lbnQ+QVFBQjwvRXhwb25l
bnQ+CiAgICAgICAgICAgICAgICA8L1JTQUtleVZhbHVlPgogICAgICAgICAgICAgICAgIDwvS2V5VmFsdWU+CiAgICAgICAgICAgICAgICAgIDwv
S2V5SW5mbz4KICAgIDwvU2lnbmF0dXJlPgogICAgPCEtLSBEYXJrZW50Q2hhbGVuZ2U+CjwvbGV0dGVy
RGVwb3NpdFBvc3RtYXJrPgo=
```

I.3 Notification de transit

La notification de transit contient des informations sur l'expéditeur, l'enveloppe et l'interrogation permettant d'ouvrir l'enveloppe, et est cosignée par les serveurs Cmail.

Elle constitue une preuve de transit pour l'expéditeur, qui peut l'utiliser en cas de litige.

La spécification formelle de la notification de transit figure dans l'Annexe A.

Exemple: fichier "1373360283931.laura.prin@legalbox.com.receipt.notice"


```
Received: from begmeil get hostname ([127.0.0.1])
        by localhost
        with SMTP (LegalBox POP Server v1.0) id HIWX27L5
        for laura.prin@legalbox.com;
        Tue, 09 Jul 2013 11:49:01 +0200 (CEST)
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=receiptNotice.xml

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+Cjxs
ZXR0ZXJEZXBvc2l0UG9zdG1hcms+CiAgPG9wZXJhdG9yUG9zdG1hcms+CiAgICA8ZW52ZWxvcEl
PjEzNzMzNjAyODM5MzE8L2VudmVsb3BJZD4KICAgIDxkZWxpdmVyeU1vZGU+Y2VydGlmaWVkTGV0
dGVyPC9kZWxpdmVyeU1vZGU+CiAgICA8bWltZU1lc3NhZ2VIYXNoPgogICAgICA8c2hhMT5hNTVk
ZDhmYWU0Mzg2M2VmYWRmMWY3ZjM3MmEwYmU1MmEwMGRhYTFkPC9zaGExPgogICAgPC9taW1lTWVz
...
MDkwSDl0NFVkTTdWVU92bjY3WlU2aTJvVSt3b3lGR2tYMDJ3YkVMM2pDYmpJCm5VR1BwUGpoT3Zo
dzNPTy9mYmhKVk13dkM2NXB1MTl1cnA2M05kS0tHNlBuNjZtQkVnUldxZ2cvTVBITmZmWkhrOXFs
WExSSXhETi8Kb0ZnS285RmI0NExlSzBnZ3VyblY2azNicm1TeGM1UnpYVWNxTzdwbldUN0FoNFl6
WXJJUHdYL1hjS1VqbXYxZi9JZjQ5VHVnWGtLcgpodklyOG9qUkdQcEdPdlB4cWR5QWNQR1BOUVRY
NFJrc29kSEVwdz09PC9Nb2R1bHVzPgogICAgICAgICAgICAgICA8RXhwb25lbnQ+QVFBQjwvRXhwb25l
bnQ+CiAgICAgICAgICAgICAgICA8L1JTQUtleVZhbHVlPgogICAgICAgICAgICAgICAgIDwvS2V5VmFsdWU+CiAgICAgICAgICAgICAgICAgIDwv
S2V5SW5mbz4KICAgIDwvU2lnbmF0dXJlPgogICAgPCEtLSBEYXJrZW50Q2hhbGVuZ2U+CjwvbGV0dGVy
RGVwb3NpdFBvc3RtYXJrPgo=
```

I.4 ENVELOPPE

ENVELOPPE est un message MIME contenant le contenu du courrier électronique chiffré selon la norme de codage AES.

Exemple: fichier "1373360283931.certifiedLetter.msg"

```
Received: from localhost ([127.0.0.1])
        by begmeil
        with SMTP (SubEthaSMTP null) id HIWV8HF9
        for laura.prin@legalbox.com;
        Tue, 09 Jul 2013 10:58:03 +0200 (CEST)
Date: Tue, 9 Jul 2013 10:57:51 +0200 (CEST)
From: david.keller@legalbox.com
To: laura.prin@legalbox.com
Message-ID: proto_cmtp_1373360269856
Subject: =?UTF-8?Q?Bienvenue_=C3=A0_CMTP!?=
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="----=_Part_1_1013939722.1373360271613"

------=_Part_1_1013939722.1373360271613
Content-Type: multipart/mixed;
        boundary="----=_Part_0_2062834323.1373360271584"

------=_Part_0_2062834323.1373360271584
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=envelop

RG44gUlyr1A/L+ps0R+yKMUpgPcJACmcRQdLZSMoLnm07gtRataSAWkG5qnc/f5Q

------=_Part_0_2062834323.1373360271584--

------=_Part_1_1013939722.1373360271613--
```

Bibliographie

[b-UIT-T X.509] Recommandation UIT-T X.509 (2012) | ISO/CEI 9594-8:2014, *Technologies de l'information - Interconnexion des systèmes ouverts – L'annuaire: cadre général des certificats de clé publique et d'attribut*.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série D	Principes généraux de tarification
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	Terminaux et méthodes d'évaluation subjectives et objectives
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	**Réseaux de données, communication entre systèmes ouverts et sécurité**
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2016