UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.36

(07/97)

SÉRIE T: TERMINAUX DES SERVICES TÉLÉMATIQUES

Capacités de sécurité à utiliser avec les télécopieurs du Groupe 3

Recommandation UIT-T T.36

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE T

TERMINAUX DES SERVICES TÉLÉMATIQUES

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

RECOMMANDATION UIT-T T.36

CAPACITÉS DE SÉCURITÉ À UTILISER AVEC LES TÉLÉCOPIEURS DU GROUPE 3

Résumé

La présente Recommandation définit les deux solutions techniques indépendantes, fondées sur les algorithmes HKM/HFX40 et l'algorithme RSA, qui peuvent être appliquées afin d'assurer la sécurité des transmissions par télécopie.

L'Annexe A donne des informations sur les algorithmes HKM/HFX40.

L'Annexe B décrit l'algorithme RSA.

L'Annexe C présente l'utilisation du système HKM qui permet d'assurer la gestion de clés sur des télécopieurs. Ces fonctions sont décrites sous forme de deux procédures principales:

– une procédure permettant d'effectuer un enregistrement unidirectionnel entre les entités X et Y (procREGxy); et

– une procédure assurant la transmission sécurisée d'une clé secrète entre les entités X et Y (procSTKxy).

L'Annexe D traite des procédures concernant l'utilisation du système de chiffrement HFX40 qui permet d'assurer la confidentialité des messages transmis par télécopie.

L'Annexe E décrit l'algorithme de hachage HFX40-I en indiquant son utilisation, les calculs à effectuer et les informations à échanger entre les télécopieurs afin d'assurer l'intégrité d'un message de télécopie, cette fonction pouvant être sélectionnée ou préprogrammée en remplacement du chiffrement du message.

Source

La Recommandation UIT-T T.36, élaborée par la Commission d'études 8 (1997-2000) de l'UIT-T, a été approuvée le 2 juillet 1997 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) 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.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études 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 n° 1 de la CMNT.

Dans certains secteurs de la technologie 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.

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/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 responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

TABLE DES MATIÈRES

Page

CAPACITÉS DE SÉCURITÉ À UTILISER AVEC LES TÉLÉCOPIEURS DU GROUPE 3

(Genève, 1997)

1 Domaine d'application

La présente Recommandation définit les deux solutions techniques indépendantes qui peuvent être appliquées afin d'assurer la sécurité des transmissions par télécopie:

- une solution fondée sur les algorithmes HKM/HFX40 décrits à l'Annexe A et à l'Annexe G/T.30;
- une solution fondée sur l'algorithme RSA décrit à l'Annexe B et à l'Annexe H/T.30.

2 Références normatives

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 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.

– Recommandation UIT-T T.30 (1996), *Procédures pour la transmission de documents par télécopie sur le réseau téléphonique général commuté*.

3 Abréviations

La présente Recommandation utilise les abréviations suivantes:

ASCII	American Standard Code for Information Interchange
B(n)	valeur de base (n)
ESH	valeur de hachage chiffrée et embrouillée (24 chiffres décimaux) (*encrypted and scrambled plain hash*)
ESIM	message d'intégrité embrouillé et chiffré. Nombre de 12 chiffres décimaux (*encrypted scrambled integrity message*)
ESSK	clé secrète embrouillée et chiffrée. Nombre de 12 chiffres décimaux (*encrypted scrambled secret key*)
HKM	algorithme de chiffrement HKM
HKMD+1	double chiffrement à l'aide de l'algorithme HKM
IDx	six derniers chiffres de l'identification télécopieur (numéro de téléphone pour la télécopie) de X
IDy	six derniers chiffres de l'identification télécopieur (numéro de téléphone pour la télécopie) de Y
IM	message d'intégrité servant à confirmer ou à réfuter l'intégrité du message reçu (12 chiffres décimaux) (*integrity message used to confirm or deny integrity of the received message*)
IMy	message d'intégrité émis par Y pour confirmer ou réfuter l'intégrité du message reçu. Nombre de 12 chiffres décimaux (*integrity message generated by Y to confirm or deny integrity of the received message*)

MPx	primitive mutuelle de X. Nombre de 16 chiffres décimaux, qui ne peut être créé que par X au moyen de l'algorithme HKM et des primitives formées avec UINx, UCNx, IDx et IDy (*mutual primitive of X*)
MPy	primitive mutuelle de Y (*mutual primitive of Y*)
OT	clé à usage unique. Nombre de 6 à 64 chiffres décimaux défini par les deux utilisateurs (*one-time key*)
OTx	clé à usage unique utilisée en premier par X lors de son enregistrement auprès de Y (*one-time key as first used by X in X's registration with Y*)
OTy	clé à usage unique utilisée en premier par Y lorsque celui-ci s'enregistre auprès de X en vue de l'enregistrement mutuel, cette clé pouvant être différente ou non d'OTx
PH	valeur de hachage du message (24 chiffres décimaux) (*plain hash*)
P(n)	valeur de phase (n)
Primitive	nombre composé de 64 chiffres formé à partir de UIN et de UCN
procREGxy	procédure d'enregistrement entre X et Y
procSTKxy	procédure permettant à X de procéder à la transmission sécurisée d'une clé secrète à Y
PRS	séquence pseudo-aléatoire (*pseudorandom sequence*)
RCN	nombre crypté enregistré. Nombre de 16 chiffres décimaux (*registered crypt number*)
RNCn	nombre aléatoire non secret associé à une clé SCn. Nombre de 4 chiffres décimaux (*non-secret random number associated with an SCn*)
RNIM	nombre aléatoire non secret associé à un message IM. Nombre de 4 chiffres décimaux (*non-secret random number associated with an IM*)
RNK	nombre aléatoire non secret permettant de faire varier les primitives créées par MPx lors du chiffrement d'une clé SK. Nombre de 4 chiffres décimaux
RNSRn	nombre aléatoire non secret associé à une clé SRn. Nombre de 4 chiffres décimaux
RNSSn	nombre aléatoire non secret associé à une clé SSn. Nombre de 4 chiffres décimaux
SCn	clé d'épreuve secrète, numéro n. Nombre de 12 chiffres décimaux
SH	valeur de hachage embrouillée (24 chiffres décimaux) (*scrambled plain hash*)
SK	clé secrète pouvant être une clé SCn, SRn, SSn, etc. Nombre de 12 chiffres décimaux (*secret key*)
SRn	clé de réponse secrète, numéro n. Nombre de 12 chiffres décimaux (*secret response key, number n*)
SS	clé de session secrète utilisée avec l'algorithme d'intégrité HFX40-I (12 chiffres décimaux) (*secret session key*)
SSK	clé secrète embrouillée. Nombre de 12 chiffres décimaux (*scrambled secret key*)
SSn	clé de session secrète, numéro n, à utiliser avec la fonction de chiffrement ou de hachage. Nombre de 12 chiffres décimaux (*secret session key, number n*)
SSx	clé de session secrète créée par X et à utiliser avec l'algorithme de chiffrement HFX40 (12 chiffres décimaux) (*secret session key generated by X*)
TKx	clé de transfert, résultat du chiffrement de la primitive MPx produite par X. Nombre de 16 chiffres décimaux (*transfer key, an encryption of MPx generated by X*)
UCN	nombre crypté unique, par exemple UCNx, UCNy. Nombre de 16 chiffres décimaux connu du système uniquement (*unique crypt number*)

UIN	numéro d'identité unique, par exemple UINx, UINy. Nombre de 48 chiffres décimaux connu du système uniquement (*unique identity number*)
UIT-T	union internationale des télécommunications – Secteur de la normalisation des télécommunications
X	nom d'une entité
x	suffixe indiquant qu'un élément appartient à X ou est créé par X
XOR	OU exclusif
Y	nom d'une seconde entité
y	suffixe indiquant qu'un élément appartient à Y ou est créé par Y

Annexe A

Procédures visant à assurer la sécurité de transmission de documents par télécopie de Groupe 3 au moyen des systèmes HKM et HFX

A.1 Introduction

A.1.1 La présente annexe décrit les procédures utilisées sur les télécopieurs du Groupe 3 pour assurer la sécurité des communications au moyen des systèmes HKM et HFX.

A.1.2 L'utilisation de l'annexe est facultative.

A.1.3 Il est obligatoire d'appliquer le mode de correction d'erreurs défini à l'Annexe A/T.30 ou à l'Annexe C/T.30 (selon qu'il sera approprié).

A.2 Présentation de la procédure visant à assurer la sécurité de transmission de documents par télécopie

A.2.1 Les systèmes HKM et HFX offrent les fonctions suivantes qui permettent d'assurer la sécurité des communications de documents entre des entités (télécopieurs ou utilisateurs de télécopieurs):

– authentification mutuelle des entités;

– établissement d'une clé de session secrète;

– confidentialité des documents;

– confirmation de réception;

– confirmation ou réfutation de l'intégrité des documents.

A.2.2 Fonctions

La gestion de clés est assurée par le système HKM défini à l'Annexe C. Deux procédures sont spécifiées, la première étant l'enregistrement et la seconde la transmission sécurisée d'une clé secrète. L'enregistrement définit des secrets mutuels et garantit la sécurité de toutes les transmissions ultérieures, au cours desquelles le système HKM assure une authentification mutuelle, fournit une clé de session secrète qui garantit la confidentialité et l'intégrité des documents et assure une confirmation de réception ainsi qu'une confirmation ou une réfutation de l'intégrité des documents.

La confidentialité des documents est assurée grâce à un système de chiffrement défini à l'Annexe D. Ce système utilise une clé de 12 chiffres décimaux équivalant à environ 40 bits.

L'intégrité des documents est assurée au moyen du système spécifié à l'Annexe E, qui définit l'algorithme de hachage utilisé ainsi que les calculs associés et les échanges d'informations.

Annexe B

Sécurité des télécopieurs du Groupe 3 basée sur l'algorithme RSA

B.1 Préambule

(Le préambule est laissé exprès en blanc.)

B.2 Introduction

La présente annexe définit des mécanismes de sécurisation fondés sur le système cryptographique RSA.

B.3 Références

– ISO/CEI 9796:1991, *Technologies de l'information – Techniques de sécurité – Schéma de signature numérique rétablissant le message*.

– RIVEST (R.L.), SHAMIR (A.), ADLEMAN (L.): A method for obtaining digital signatures and public-key cryptosystems, Annexe A: RSA, *CACM (Communications of the ACM)*, Vol. 21, n° 2, p. 120-126, 1978.

– Projet de Norme ISO/CEI CD 10118-3:1995, *Technologies de l'information – Techniques de sécurité – Fonctions d'adressage dispersé*.

– ISO/CEI JTC 1/SC 27 N1108:

  - SHA-1 (Secure Hash Algorithm), décrit dans *Secure Hash Standard*, FIPS (Federal Information Processing Standard) PUB 180-1, avril 1995, algorithme provenant de la NIST (National Institute of Standardization), Etats-Unis.
  - MD-5 (RFC 1321).

– ISO/CEI 9979:1991, *Technologies de l'information – Techniques de sécurité – Procédures pour l'enregistrement des algorithmes cryptographiques*.

B.4 Description technique

Cette solution est décrite en détail à l'Annexe H/T.30.

Annexe C

Procédures pour l'utilisation du système de gestion de clés HKM visant à assurer la sécurité de transmission des documents par télécopie

C.1 Domaine d'application

La présente annexe a pour objet de définir le système de gestion de clés HKM qui permet d'assurer la sécurité des échanges de clés sur des télécopieurs.

Le système HKM est destiné à être utilisé avec tous les types de télécopieurs spécialisés, mais est également applicable aux systèmes de télécopie intégrés à des ordinateurs.

La présente annexe décrit l'utilisation de l'algorithme HKM dans deux procédures principales, procREGxy et procSTKxy, pour assurer:

– une authentification mutuelle (décrite aux C.5.3 et C.6);

– l'utilisation d'un système de chiffrement garantissant la confidentialité des messages (Annexe D);

– l'utilisation d'une fonction de hachage garantissant l'intégrité des messages (Annexe E);

– un échange sécurisé de clés qui peuvent être des clés d'épreuve ou de réponse destinées à une authentification, ainsi que des clés de session visant à assurer la confidentialité ou l'intégrité des messages (voir la description aux C.5 et C.6).

Une notation algébrique a été définie pour faciliter la présentation des protocoles et des procédures de gestion de clés (voir C.2).

Le système HKM repose sur l'utilisation de 19 nombres premiers système, qui sont également employés avec l'algorithme de chiffrement des messages décrit à l'Annexe D et l'algorithme de hachage permettant d'assurer l'intégrité du message principal décrit à l'Annexe E. La présente annexe ne traite toutefois pas des algorithmes de chiffrement et de hachage susmentionnés.

Des exemples de calculs sont donnés au C.6 et peuvent servir à vérifier la mise en application de la présente annexe.

Le système de gestion de clés HKM est visé par des droits de propriété intellectuelle; toutefois, le détenteur de ces droits est convenu d'observer le code de pratique du TSB. Des renseignements complémentaires peuvent être obtenus auprès du TSB.

C.2 Conventions

C.2.1 Généralités

La notation algébrique indiquée au C.2.2 permet de décrire les protocoles et les procédures de gestion de clés.

C.2.2 Symboles

[ ]	symboles délimitant un message
{ }	symboles délimitant un algorithme
( )	symboles délimitant des primitives
< >	symboles délimitant des informations à mémoriser
> <	symboles délimitant des informations archivées à extraire
&	fusion (par exemple, entre UCNx et IDx) ou modification (par exemple, de UCNx par IDx) sans modification de longueur
RCNx>>>>>>	transmission du nombre RCNx à Y
>>>>>> RCNx	réception du nombre RCNx transmis par X
HKM+1	chiffrement à l'aide de l'algorithme HKM
HKM–1	déchiffrement à l'aide de l'algorithme HKM
HKMD+1	double chiffrement à l'aide de l'algorithme HKM
HKMD–1	double déchiffrement à l'aide de l'algorithme HKM

C.3 Description de l'algorithme HKM à utiliser pour les télécopieurs

L'algorithme HKM utilise des nombres secrets propres aux télécopieurs et d'autres variables spécifiques aux utilisateurs pour créer des primitives qui, associées à une clé de chiffrement, fournissent les nombres qui serviront à effectuer des calculs de congruences. Les nombres secrets UIN et UCN sont mémorisés de manière sûre dans le télécopieur lors de la fabrication ou de la mise en service. Il n'est pas nécessaire d'établir une référence croisée entre ces nombres et un numéro de série quelconque.

Les arguments de modulos appliqués dans les calculs de congruences proviennent d'un ensemble de 19 nombres premiers spéciaux mémorisés dans le télécopieur.

On obtient grâce aux opérations arithmétiques précitées des séquences PRS longues qui permettent de chiffrer le message.

L'algorithme HKM est également utilisé dans un mode irréversible, c'est-à-dire lorsqu'un processus de chiffrement a lieu, mais sans que le processus inverse soit possible.

Les caractéristiques susmentionnées constituent la base cryptographique des procédures procREGxy et procSTKxy.

Voir C.6 pour des précisions sur les calculs de congruences et les nombres premiers spéciaux.

C.4 Mode d'enregistrement

C.4.1 Procédure d'enregistrement entre les entités X et Y (procREGxy)

Pour s'enregistrer auprès de Y, X crée un nombre irréversible en combinant, selon le mode cryptographique, UINx et UCNx avec IDx et IDy. Le nombre de 16 chiffres décimaux ainsi obtenu est la primitive MPx, qui sert à effectuer le chiffrement et le transfert sécurisé des clés, comme l'expliquent les sous-paragraphes suivants.

Dans un environnement sûr sans transmission d'enregistrement, les utilisateurs définissent une clé OT. L'utilisateur en X choisit un mode d'enregistrement et compose l'identification du télécopieur (numéro de téléphone pour la télécopie) en Y. IDx et IDy constituent les éléments de base des autres primitives utilisées par l'algorithme. L'utilisateur en X introduit également la clé OTx.

X se sert de la clé OTx et de HKM+1 pour chiffrer la primitive MPx et créer la clé TKx qui est transmise à Y. Y introduit la clé OTx et l'utilise avec HKM–1 pour déchiffrer TKx et déterminer MPx.

Y ne mémorise pas la primitive MPx mais la chiffre immédiatement, à l'aide de HKM+1 et des primitives créées avec les nombres UINy et UCNy modifiés par IDx et IDy, pour créer RCNy. Y transmet RCNy à X pour mémorisation. Il n'est pas nécessaire pour Y de mémoriser RCNy car X retransmettra ouvertement ce nombre à Y la prochaine fois qu'il effectuera une transmission sécurisée vers Y.

Les deux terminaux procèdent implicitement à une authentification mutuelle étant donné que X est le seul terminal capable de créer la primitive MPx relative à Y et que Y est le seul terminal capable de déterminer MPx à partir de RCNy.

La procédure procREGxy peut être représentée ainsi en notation algébrique:

X	**Y**
> UINx, UCNx <	> UINy, UCNy <
MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy] TKx = (OTx){HKM+1}[MPx] TKx >>>>>>	
	>>>>>> TKx MPx = (OTx){HKM–1}[TKx] RCNy = (UINy, UCNy & IDx & IDy){HKM+1}[MPx] RCNy >>>>>>
>>>>>> RCNy <RCNy>	

Voir C.6.4 pour une description et des exemples de tous les calculs concernant la procédure procREGxy.

C.4.2 Procédure d'enregistrement entre les entités Y et X (procREGyx)

Pour s'enregistrer auprès de X, Y suit une procédure identique, procREGyx, qui crée la primitive MPy et le nombre RCNx. (La clé OTy définie par les utilisateurs en Y et en X peut être identique ou non à la clé OTx utilisée pendant la procédure procREGxy.)

La procédure procREGyx peut être représentée ainsi en notation algébrique:

Y	**X**
>UINy, UCNy<	> UINx, UCNx <
MPy = (UINy, UCNy & IDy & IDx){HKM+1}[UCNy & IDy & IDx] TKy = (OTy){HKM+1}[MPy] TKy>>>>>>	
	>>>>>>TKy MPy = (OTy){HKM–1}[TKy] RCNx = (UINx, UCNx & IDy & IDx){HKM+1}[MPy] RCNx>>>>>>
>>>>>>RCNx <RCNx>	

C.4.3 Procédure d'enregistrement en une seule communication

Il est possible de combiner en une seule communication les deux enregistrements effectués séparément à l'aide des procédures procREGxy et procREGyx. Ce processus est représenté ci-après en notation algébrique. Dans l'exemple donné, c'est X qui lance l'appel.

X

> UINx, UCNx <

MPx = (UINx, UCNx & IDx & IDy) {HKM+1} [UCNx & IDx & IDy]
TKx = (OTx) {HKM+1} [MPx]
TKx >>>>>>

>>>>>> RCNy, TKy
< RCNy >
MPy = (OTy) {HKM–1} [TKy]
RCNx = (UINx, UCNx & IDy & IDx) {HKM+1} [MPy]
RCNx >>>>>>

Y

>UINy, UCNy<

>>>>>> TKx
MPx = (OTx) {HKM–1} [TKx]
RCNy = (UINy, UCNy & IDx & IDy) {HKM+1} [MPx]

MPy = (UINy, UCNy & IDy & IDx){HKM+1}[UCNy & IDy & IDx]
TKy = (OTy) {HKM+1} [MPy]
RCNy, Tky >>>>>>

>>>>>> RCNx
< RCNx >

C.4.4 Authentification de l'enregistrement

Les procédures d'enregistrement entre les entités X et Y contiennent une fonction d'authentification des enregistrements par échange d'épreuves et de réponses entre X et Y (procédures procSTKxy et procSTKyx décrites au C.5.1). Un exemple de cette fonction, dans le cadre de la procédure d'enregistrement bilatéral effectuée en une seule communication, est représenté ci-après en notation algébrique.

X

>UINx, UCNx<
MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy]
TKx = (OTx){HKM+1}{MPx}

<SC0x> by procSTKxy = ESSC0x
TKx, RNC0x, ESSC0x>>>>>>

>>>>>>RCNy, TKy, RNSR0y, ESSR0y, RNC0y, ESSC0y

<RCNy>

MPy = (OTy){HKM–1}[TKy]
RCNx = (UINx, UCNx & IDy & IDx){HKM+1}[MPy]

ESSR0y by procSTKyx = SR0y
Compare SR0y with <SC0y>

ESSC0y by STKyx = SC0y
SC0y = SR0x
SR0x by STKxy = ESSR0x

RCNx, RNSR0x, ESSR0x>>>>>>

Y

>UINy, UCNy<

>>>>>TKx, RNC0x, ESSC0x

MPx = (OTx){HKM–1}[TKx]
RCNy = (UINy, UCNy & IDx & IDy){HKM+1}[MPx]

MPy = (UINy, UCNy & IDy & IDx){HKM+1}[UCNy & IDy & IDx]
TKy = (OTy){HKM+1}[MPy]

ESSC0x by procSTKxy = SC0x
SC0x = SR0y
SR0y by procSTKyx = ESSR0y

<SC0y> by procSTKyx = ESSC0y

RCNy, TKy, RNSR0y, ESSR0y, RNC0y, ESSC0y>>>>>>

>>>>>>RCNx, RNR0x, ESSR0x

<RCNx>

ESSR0x by procSTKxy = SR0x
Compare SR0x with <SC0y>

Si la clé d'épreuve SC0x est égale à la clé de réponse SR0y et si la clé d'épreuve SC0y est égale à la clé de réponse SR0x, l'authentification mutuelle des enregistrements entre X et Y est réalisée.

C.5 Mode sécurisé

Après l'enregistrement des primitives MPx et MPy, on utilise l'algorithme HKM pour assurer la communication sécurisée des clés secrètes entre X et Y. Les clés secrètes peuvent être les clés SC, SR ou SS. La procédure procSTKxy utilisée à cette fin est décrite dans les sous-paragraphes suivants.

C.5.1 Procédure permettant la transmission sécurisée de la clé SK de X à Y (procSTKxy)

La clé SKx, qui est une clé secrète que X doit transmettre en mode sécurisé à Y, est embrouillée et chiffrée à l'aide de HKMD+1, de manière à créer la clé ESSKx. Les primitives utilisées avec HKMD+1 sont créées à partir de la primitive MPx et modifiées par le nombre RNKx. RNKx est transmis ouvertement à Y conjointement avec RCNy et ESSKx.

Y détermine MPx à l'aide de RCNy, puis désembrouille et déchiffre ESSKx au moyen de HKMD–1 pour calculer SKx. Les primitives utilisées avec HKMD–1 sont établies à partir de la primitive MPx et modifiées par RNKx en X.

La procédure procSTKxy peut être représentée ainsi en notation algébrique:

X
> UINx, UCNx, RCNy <

MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy]
ESSKx = (MPx & RNKx){HKMD+1}[SKx]
RCNy, RNKx, ESSKx >>>>>>

Y
> UINy, UCNy, RCNx <

>>>>>> RCNy, RNKx, ESSKx
MPx = (UINy, UCNy & IDx & IDy){HKM–1}[RCNy]
SKx = (MPx & RNKx){HKMD–1}[ESSKx]

Voir C.6.5 pour une description et des exemples de tous les calculs concernant la procédure procSTKxy.

C.5.2 Utilisation de procSTKxy et de procSTKyx en mode sécurisé

Une fois l'enregistrement de X et de Y effectué, toutes les transmissions futures peuvent être effectuées en mode sécurisé. On utilise l'algorithme HKM dans ce mode pour recréer les primitives MPx et MPy et assurer le transfert sécurisé des clés secrètes à l'aide des procédures procSTKxy et procSTKyx.

C.5.3 Authentification mutuelle de X et de Y

Dans ce contexte, X a établi une communication pour envoyer un message en mode sécurisé à Y.

En X
- X compose le numéro de téléphone pour la télécopie de Y;
- X recrée la primitive MPx, précédemment utilisée lors de l'enregistrement entre X et Y;
- X crée SC1x et RNC1x, et mémorise SC1x;
- X utilise procSTKxy avec MPx, RNC1x et SC1x pour créer ESSC1x;
- X envoie RCNy, RNC1x et ESSC1x à Y.

En Y
- Y déchiffre RCNy pour créer MPx;
- Y utilise procSTKxy avec MPx, RNC1x et ESSC1x pour déterminer SC1x;
- Y recrée MPy;
- Y utilise SC1x comme clé de réponse secrète SR1y et crée RNSR1y;
- Y utilise procSTKyx avec MPy, RNSR1y et SR1y pour produire ESSR1y;
- Y crée SC1y et RNC1y et mémorise SC1y;
- Y utilise procSTKyx avec MPy, RNC1y et SC1y pour créer ESSC1y;
- Y envoie RCNx, RNSR1y, ESSR1y, RNC1y et ESSC1y à X.

En X
– X déchiffre RCNx pour créer MPy;
– X utilise procSTKyx avec MPy, RNSR1y et ESSR1y pour créer SR1y;
– X compare SR1y avec SC1x et en cas d'égalité, Y est authentifié par rapport à X;
– X utilise procSTKyx avec MPy, RNC1y et ESSC1y pour déterminer SC1y;
– X utilise SC1y comme clé de réponse secrète SR1x et crée RNSR1x;
– X utilise procSTKxy avec MPx, RNSR1x et SR1x pour créer ESSR1x;
– X envoie RNSR1x et ESSR1x à Y.

En Y
– Y utilise procSTKxy avec MPx, RNSR1y et ESSR1x pour déterminer SR1x;
– Y compare SR1x avec SC1y et en cas d'égalité, X est authentifié par rapport à Y.

A ce stade, X et Y ont échangé les nombres RCNx et RCNy et ont procédé aux échanges mutuels d'épreuves et de réponses. En cas d'égalité entre SC1x et SR1y, et entre SC1y et SR1x, l'authentification mutuelle est réalisée.

Le processus d'authentification mutuelle de X et de Y peut être représenté ainsi en notation algébrique:

X	**Y**
> UINx, UCNx, RCNy <	> UINy, UCNy, RCNx <
< SC1x > by procSTKxy = ESSC1x RCNy, RNC1x, ESSC1x >>>>>>	
	>>>>>>RCNy, RNC1x, ESSC1x < RCNy > ESSC1x by procSTKxy = SC1x SC1x = SR1y SR1y by procSTKyx = ESSR1y < SC1y > by procSTKyx = ESSC1y RCNx, RNSR1y, ESSR1y, RNC1y, ESSC1y >>>>>>
>>>>>> RCNx, RNSR1y, ESSR1y, RNC1y, ESSC1y < RCNx > ESSR1y by procSTKyx = SR1y Compare SR1y with < SC1x > ESSC1y by procSTKyx = SC1y SC1y = SR1x SR1x by procSTKxy = ESSR1x RNSR1x, ESSR1x >>>>>>	
	>>>>>> RNSR1x, ESSR1x ESSR1x by procSTKxy = SR1x Compare SR1x with < SC1y >

Si la clé d'épreuve SC1x est égale à la clé de réponse SR1y et si la clé d'épreuve SC1y est égale à la clé de réponse SR1x, l'authentification mutuelle est réalisée.

C.5.4 Définition d'une clé de session secrète entre X et Y

Dans ce contexte, X a établi une communication pour envoyer un message en mode sécurisé à Y et l'authentification mutuelle a été réalisée de manière satisfaisante comme indiqué au C.5.3.

En X
– X recrée la primitive MPx précédemment utilisée lors de l'enregistrement entre X et Y;
– X crée SS1x et RNSS1x;
– X utilise procSTKxy avec MPx, RNSS1x et SS1x pour créer ESSS1x;
– X envoie RCNy, RNSS1x et ESSS1x à Y.

En Y
– Y déchiffre RCNy pour déterminer MPx;
– Y utilise procSTKxy avec MPx, RNSS1x et ESSS1x pour déterminer SS1x.

Le processus de définition d'une clé de session secrète entre X et Y peut être représenté ainsi en notation algébrique:

X	**Y**
< RCNy >	< RCNx >
SS1x by procSTKxy = ESSS1x RCNy, RNSS1x, ESSS1x >>>>>>	
	>>>>>> RCNy, RNSS1x, ESSS1x ESSS1x by procSTKxy = SS1x

X et Y utilisent la clé SS1x avec le système de chiffrement HFX40 pour chiffrer et déchiffrer le message principal et assurer la confidentialité des messages (Annexe D) ou l'algorithme de hachage HFX40-I afin d'assurer l'intégrité des messages (Annexe E) pendant la transmission.

C.5.5 Confirmation de la réception

Dans ce contexte, X a établi une communication pour envoyer un message en mode sécurisé à Y, l'authentification mutuelle a été réalisée comme indiqué au C.5.3, la clé SS1x a été échangée en mode sécurisé comme indiqué au C.5.4 et le message a été envoyé. A la fin du message, X envoie à Y la clé SC2x, qui est utilisée par Y comme clé SR2y si le message a été intégralement reçu.

En X
- X recrée la primitive MPx, précédemment utilisée lors de l'enregistrement entre X et Y;
- X crée SC2x et RNC2x, et mémorise SC2x;
- X utilise procSTKxy avec MPx, RNC2x et SC2x pour créer ESSC2x;
- X envoie RCNy, RNC2x et ESSC2x à Y.

En Y
- Y déchiffre RCNy pour créer MPx;
- Y utilise procSTKxy avec MPx, RNC1x et ESSC2x pour déterminer SC2x;
- Y recrée MPy;
- Y utilise SC2x comme clé SR2y et crée RNSR2y;
- Y utilise procSTKyx avec MPy, RNSR2y et SR2y pour créer ESSR2y;
- Y envoie RCNx, RNSR2y et ESSR2y à X.

En X
- X déchiffre RCNx pour créer MPy;
- X utilise procSTKyx avec MPy, RNSR2y et ESSR2y pour déterminer SR2y;
- X compare SR2y à SC2x et en cas d'égalité, cela signifie que Y a confirmé la réception du message à X.

Le processus de confirmation de réception peut être représenté ainsi en notation algébrique:

X	**Y**
> RCNy <	> RCNx <
< SC2x > by procSTKxy = ESSC2x RCNy, RNC2x, ESSC2x >>>>>>	
	>>>>>>RCNy, RNC2x, ESSC2x ESSC2x by procSTKxy = SC2x SC2x = SR2y SR2y by procSTKyx = ESSR2y RCNx, RNSR2y, ESSR2y >>>>>>
>>>>>> RCNx, RNSR2y, ESSR2y ESSR2y by procSTKyx = SR2y Compare SR2y with < SC2x >	

Si la clé SR2y transmise par Y est égale à la clé SC2x mémorisée par X, X accepte la confirmation de la réception du message par Y.

C.5.6 Confirmation ou réfutation d'intégrité

Dans ce contexte, X a établi une communication pour envoyer à Y un message dont l'intégrité est protégée, l'authentification mutuelle a été réalisée de manière satisfaisante comme indiqué au C.5.3, la clé SSx a été échangée en mode sécurisé comme indiqué au C.5.4 et le message a satisfait ou non au test d'intégrité en Y (Annexe E).

Y émet le message IMy pour confirmer ou réfuter l'intégrité du message reçu. IMy est un nombre aléatoire de 12 chiffres choisis parmi les chiffres 2 à 9. Un de ces chiffres, sélectionné de manière aléatoire, sera remplacé par '1' pour indiquer la confirmation de l'intégrité du message ou par '0' pour en indiquer la réfutation.

En Y
- Y crée un nombre aléatoire de 12 chiffres choisis parmi les chiffres 2 à 9;
- Y choisit parmi ces 12 chiffres un chiffre qui sera le pointeur de remplacement du nombre aléatoire susmentionné;
- Y remplace ce chiffre par '1' ou par '0' pour produire IMy;
- Y crée RNIMy;
- Y utilise procSTKyx avec MPy, RNIMy et IMy pour créer ESIMy;
- Y envoie RCNx, RNIMy et ESIMy à X.

En X

– X déchiffre RCNx pour créer MPy;
– X utilise procSTKyx avec MPy, RNIMy et ESIMy pour déterminer IMy;
– X vérifie si IMy contient '1' (confirmation de l'intégrité du message) ou '0' (réfutation de l'intégrité du message).

Le processus de confirmation d'intégrité peut être représenté ainsi en notation algébrique:

X	**Y**
	> RCNx <
	IMy by procSTKyx = ESIMy RCNx, RNIMy, ESIMy >>>>>>
>>>>>> RCNx, RNIMy, ESIMy ESIMy by procSTKyx = IMy Check IMy for a '1' or a '0'	

Si le message d'intégrité secret contient '1', l'intégrité est confirmée; s'il contient '0', l'intégrité est réfutée.

Exemples de réponses:

IMy = 257795**1**99982 Intégrité confirmée.

IMy = 3**1**7736845378 Intégrité confirmée.

IMy = 738543680892 Intégrité réfutée.

IMy = 4577452**0**4639 Intégrité réfutée.

C.6 L'algorithme HKM

C.6.1 Introduction

Le sous-paragraphe C.6 décrit l'algorithme HKM en indiquant les nombres qui doivent être mémorisés, ainsi que les règles à suivre pour les calculs effectués à l'aide de ces nombres pendant la procédure d'enregistrement (procREG) et la procédure de transmission sécurisée d'une clé (procSTK). Il sera plus facile d'expliquer les règles précitées en se servant d'exemples numériques. Les calculs effectués à l'aide des valeurs de test peuvent servir à vérifier la mise en application du C.6.

C.6.2 Informations mémorisées

Tous les terminaux sont dotés du même ensemble de 19 nombres premiers système qui serviront à effectuer les calculs de congruences.

32603 32507 32183 32003 31847 31607 31583 31547 31259
31139 30803 30539 30467 30347 30323 30203 29879 29759 29663

Les neuf premiers nombres sont utilisés avec l'algorithme HKM à des fins d'enregistrement et d'authentification et pour d'autres fonctions de gestion de clés. La totalité des 19 nombres est utilisée avec l'algorithme qui assure la confidentialité des messages (Annexe D) et l'algorithme qui assure l'intégrité des messages (Annexe E).

C.6.3 Informations mémorisées en mode sécurisé

Chaque terminal est doté, grâce à un processus approprié, de deux nombres composés de chiffres décimaux, créés de manière aléatoire, qui y sont mémorisés en mode sécurisé. Il s'agit du nombre à 48 chiffres UIN et du nombre à 16 chiffres UCN, qui servent avec d'autres nombres d'identification à créer les primitives utilisées dans l'algorithme HKM.

Exemples de valeurs de test pour X et Y:

UINx = 345092978336094172898029844342879120988727823781
UCNx = 1333908734565521

UINy = 97355769383778314835370916743672287344981976735 7
UCNy = 7598247578649467

C.6.4 Mode d'enregistrement

C.6.4.1 Procédure procREGxy en notation algébrique

X	Y
>UINx, UCNx<	> UINy, UCNy <
MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy] TKx = (OTx){HKM+1}[MPx] TKx >>>>>>	
	>>>>>> TKx MPx = (OTx){HKM–1}[TKx] RCNy = (UINy, UCNy & IDx & IDy){HKM+1}[MPx] RCNy >>>>>>
>>>>>> RCNy <RCNy>	

Les sous-paragraphes suivants montrent, à l'aide de valeurs de test, tous les calculs effectués dans la procédure procREGxy.

C.6.4.2 Calculs effectués en X pour déterminer la primitive MPx

MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy]

C.6.4.2.1 Calculs préliminaires effectués avec les primitives (UINx, UCNx & IDx & IDy)

UINx = 3450929783360941728980298443428791209887278237811333908734565521
UCNx = 1333908734565521
IDx = 642092
IDy = 538249

Une primitive de 64 chiffres est créée par concaténation des nombres UINx et UCNx.

Primitive = 3450929783360941728980298443428791209887278237811333908734565521

Cette primitive est subdivisée en deux nombres de 32 chiffres; neuf valeurs primitives de phase [P(0) à P(8)] sont déterminées à partir du premier nombre et neuf valeurs primitives de base [B(0) à B(8)] du second nombre. On calcule les valeurs de phase en subdivisant le premier nombre en 7 ensembles de 4 chiffres et en 2 ensembles de 2 chiffres. Les valeurs de base sont calculées exactement de la même manière à partir du second nombre.

Les valeurs primitives de phase sont modifiées par l'addition de multiples successifs de 101 et les valeurs de base par l'addition de multiples successifs de 79. La modification par 101 et 79 permet de lancer le plus tôt possible les calculs de congruences effectués avec les nombres premiers.

Les valeurs primitives de phase et de base sont représentées ci-après à l'aide de la primitive = 3450929783360941728980298443428791209887278237811333908734565521:

P(0)	3450 + (0 * 101) = 3450	B(0)	9120 + (0 * 79) = 9120
P(1)	9297 + (1 * 101) = 9398	B(1)	9887 + (1 * 79) = 9966
P(2)	8336 + (2 * 101) = 8538	B(2)	2782 + (2 * 79) = 2940
P(3)	941 + (3 * 101) = 1244	B(3)	3781 + (3 * 79) = 4018
P(4)	7289 + (4 * 101) = 7693	B(4)	1333 + (4 * 79) = 1649
P(5)	8029 + (5 * 101) = 8534	B(5)	9087 + (5 * 79) = 9482
P(6)	8443 + (6 * 101) = 9049	B(6)	3456 + (6 * 79) = 3930
P(7)	42 + (7 * 101) = 749	B(7)	55 + (7 * 79) = 608
P(8)	87 + (8 * 101) = 895	B(8)	21 + (8 * 79) = 653

Les nombres de six chiffres IDx (642092) et IDy (538249) sont chacun subdivisés en deux groupes de trois chiffres (642, 092, 538 et 249) et sont utilisés pour modifier les valeurs P(0) à P(3) et B(0) à B(3):

P(0) = 3450 + 642 = 4092	B(0) = 9120 + 642 = 9762
P(1) = 9398 + 092 = 9490	B(1) = 9966 + 092 = 10058
P(2) = 8538 + 538 = 9076	B(2) = 2940 + 538 = 3478
P(3) = 1244 + 249 = 1493	B(3) = 4018 + 249 = 4267

C.6.4.2.2 Calculs préliminaires effectués avec le message [UCNx & IDx & IDy]

IDx et IDy sont concaténés de manière à former un nombre de 12 chiffres qui est ajouté (modulo 10) à UCNx, ce qui donne un nombre modifié, UCNx.

IDx concaténé avec IDy = 642092538249
UCNx = 1333908734565521
UCNx modifié = 7753823016955521

C.6.4.2.3 Calculs effectués avec HKM+1

L'algorithme HKM+1 utilise les neuf premiers nombres premiers décrits au C.6.2 comme arguments de modulos dans les calculs de congruences appliqués aux neuf primitives de phase P(0) à P(8) et aux neuf primitives de base B(0) à B(8) calculées au C.6.4.2.1, de manière à produire une séquence PRS de 16 valeurs (modulo 10). La séquence PRS est ajoutée (modulo 10) au nombre modifié UCNx, ce qui donne la primitive MPx (voir ci-après).

Exemple de calcul avec le premier jeu de valeurs de phase et de base, P(0) et B(0), et le premier nombre premier:

P(0) est multiplié par B(0)
4092 * 9762 = 39946104
39946104 (modulo le premier nombre premier) = 39946104 (modulo 32603) = 7429

7429 est ensuite utilisé comme nouvelle valeur de phase P et multiplié par la valeur de base B(0)
7429 * 9762 = 72521898
72521898 (modulo le premier nombre premier) = 72521898 (modulo 32603) = 12826

Ce processus est effectué 16 fois en tout (ce qui correspond au nombre de chiffres du nombre modifié UCNx). On l'applique également pour les huit autres jeux de nombres premiers, de valeurs de phase et de valeurs de base.

Les résultats du premier calcul effectué pour chacun des neuf "jeux" de nombres premiers, de valeurs de phase et de valeurs de base sont additionnés. Le résultat (modulo 10) est ajouté (modulo 10) au premier chiffre du nombre modifié UCNx, ce qui donne le premier chiffre de la primitive MPx. Le processus se répète pour chaque chiffre du nombre modifié UCNx.

Les résultats des opérations susmentionnées, qui permettent d'obtenir la primitive MPx 4314920574868366 à partir du nombre modifié UCNx 7753823016955521, sont donnés au Tableau C.1.

Tableau C.1/T.36 – Calculs effectués en X pour déterminer la primitive MPx

Nombre premier B(n) P(n)	32603 9762 4092	32507 10058 9490	32183 3478 9076	32003 4267 1493	31847 1649 7693	31607 9482 8534	31583 3930 9049	31547 608 749	31259 653 895	Total	Dernier chiffre	Nombre crypté modifié	**Primitive mutuelle**
	7429	9868	26988	2034	10651	5468	112	13734	21773	98057	7	7	**4**
	12826	8473	18636	6265	15802	12096	29581	21864	26183	151726	6	7	**3**
	11892	20587	31629	10250	6652	24076	27890	12025	30085	175086	6	5	**1**
	23024	26963	4168	20652	13780	22878	14690	23843	14853	164851	1	3	**4**
	27809	20460	13954	17825	16309	10355	29559	16471	8719	161461	1	8	**9**
	18880	17370	48	20147	14673	14768	4596	13969	4369	108820	0	2	**2**
	1801	14842	6029	7191	23904	11166	28387	7009	8388	108717	7	3	**0**
	8345	8692	17729	25123	22957	24169	9754	2627	7039	126435	5	0	**5**
	21596	12813	31017	21794	21857	19708	23041	19866	1394	173086	6	1	**7**
	9154	15406	31893	28283	23236	10672	2669	27574	3771	150658	8	6	**4**
	29128	25186	21236	11049	4223	17897	3614	13535	24261	150129	9	9	**8**
	16773	26244	31006	5664	21081	1371	22253	27060	25379	176831	1	5	**6**
	5760	5312	25818	6023	17492	9345	963	16493	5217	92423	3	5	**8**
	21548	19095	4434	1732	22773	14869	26213	27345	30729	168738	8	5	**3**
	29623	6154	5795	29754	5064	20638	24927	491	29018	151464	4	2	**6**
	23719	3604	8452	4417	6622	10579	24227	14605	5800	102025	5	1	**6**
MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy] = 4314920574868366													

C.6.4.3 Calculs effectués en X pour déterminer la clé TKx

TKx = (OTx){HKM+1}[MPx]

La primitive MPx est chiffrée à l'aide de l'algorithme HKM+1, ce qui permet de créer la clé TKx avec la clé OTx.

OTx = 71628582063812097215

La clé OTx est étendue par concaténation, de manière à produire une primitive de 64 chiffres pour l'algorithme HKM. Les neuf valeurs primitives de phase et les neuf valeurs primitives de base sont créées à partir de cette primitive de la manière qui est indiquée au C.6.4.2.1. Aucune autre modification n'est toutefois apportée aux valeurs P(0) à P(3) et B(0) à B(3).

Ci-après sont donnés les résultats des calculs effectués avec les valeurs de test pour créer les valeurs P(0) à P(8) et B(0) à B(8) à l'aide de la clé OTx.

Primitive = 71628582063812097215 71628582063812097215 71628582063812097215 7162

P(0)	7162 + 0 * 101 = 7162	B(0)	1209 + 0 * 79 = 1209
P(1)	8582 + 1 * 101 = 8683	B(1)	7215 + 1 * 79 = 7294
P(2)	638 + 2 * 101 = 840	B(2)	7162 + 2 * 79 = 7320
P(3)	1209 + 3 * 101 = 1512	B(3)	8582 + 3 * 79 = 8819
P(4)	7215 + 4 * 101 = 7619	B(4)	638 + 4 * 79 = 954
P(5)	7162 + 5 * 101 = 7667	B(5)	1209 + 5 * 79 = 1604
P(6)	8582 + 6 * 101 = 9188	B(6)	7215 + 6 * 79 = 7689
P(7)	6 + 7 * 101 = 713	B(7)	71 + 7 * 79 = 624
P(8)	38 + 8 * 101 = 846	B(8)	62 + 8 * 79 = 694

Les valeurs de phase P(0) à P(8), les valeurs de base B(0) à B(8) et les neuf nombres premiers sont utilisés de la manière indiquée au C.6.4.2.3. La primitive MPx crée le message et est ajoutée (modulo 10) à la séquence PRS, puis chiffrée au moyen de l'algorithme HKM+1, ce qui donne TKx. Les résultats des opérations qui permettent de produire TKx, 5371333066610533, à partir de MPx, 4314920574868366, sont donnés au Tableau C.2.

Tableau C.2/T.36 – Calculs effectués en X pour déterminer la clé TKx

Nombre premier B(n) P(n)	32603 1209 7162	32507 7294 8683	32183 7320 840	32003 8819 1512	31847 954 7619	31607 1604 7667	31583 7689 9188	31547 624 713	31259 694 846	Total	Dernier chiffre	Primitive mutuelle	**Clé de transfert**
	19063	10166	1847	21080	7410	2745	26944	3254	24462	116971	1	4	**5**
	29449	2337	3180	31096	30953	9607	19519	11488	2991	140620	0	3	**3**
	1365	12410	9291	1917	6993	17019	30758	7343	12660	99756	6	1	**7**
	20135	19052	7441	8439	15299	21635	4758	7717	2261	106737	7	4	**1**
	21377	30370	14484	16566	9320	29661	11148	20264	6184	159374	4	9	**3**
	23217	16082	12078	1859	5967	7709	710	25936	9213	102771	1	2	**3**
	30773	16852	4259	8985	23752	6899	26914	453	16986	135873	3	0	**3**
	4534	9521	22736	31290	16191	3546	9930	30296	3641	131685	5	5	**0**
	4302	11222	9227	16644	419	30131	15659	8051	26134	121789	9	7	**6**
	17241	642	21706	17678	17562	3021	7655	7851	6776	100132	2	4	**6**
	11052	1740	449	15669	2626	9813	20166	9239	13694	84448	8	8	**6**
	27241	13830	4014	27960	21138	31373	15427	23582	900	165465	5	6	**1**
	5339	6799	31584	28128	6501	3948	24038	14266	30679	151282	2	8	**0**
	32060	18731	24391	5579	23636	11192	4466	5730	3847	129632	2	3	**5**
	28176	29500	23019	12590	1068	30799	8353	10709	12803	157017	7	6	**3**
	27252	9167	21075	12803	31615	31462	17978	25999	7726	185077	7	6	**3**
TKx = (OTx){HKM+1}[MPx] = 5371333066610533													

C.6.4.4 Calculs effectués en Y pour déterminer la primitive MPx par déchiffrement de la clé TKx

MPx = (OTx){HKM–1}[TKx]

OTx = 71628582063812097215

Les mêmes processus que ceux qui ont servi à calculer TKx au C.6.4.3 sont utilisés avec la clé OTx. La même séquence PRS (modulo 10) est créée; toutefois, pour HKM–1, elle est soustraite (modulo 10) de TKx, ce qui crée le "message". Ce processus de soustraction déchiffre TKx pour déterminer MPx.

Les résultats des calculs effectués avec les valeurs de test pour créer les valeurs P(0) à P(8) et B(0) à B(8) à l'aide de la clé OTx susmentionnée sont représentés ci-après:

Primitive = 71628582063812097215 71628582063812097215 71628582063812097215 7162

P(0) 7162 + (0 * 101) = 7162
P(1) 8582 + (1 * 101) = 8683
P(2) 638 + (2 * 101) = 840
P(3) 1209 + (3 * 101) = 1512
P(4) 7215 + (4 * 101) = 7619
P(5) 7162 + (5 * 101) = 7667
P(6) 8582 + (6 * 101) = 9188
P(7) 6 + (7 * 101) = 713
P(8) 38 + (8 * 101) = 846

B(0) 1209 + (0 * 79) = 1209
B(1) 7215 + (1 * 79) = 7294
B(2) 7162 + (2 * 79) = 7320
B(3) 8582 + (3 * 79) = 8819
B(4) 638 + (4 * 79) = 954
B(5) 1209 + (5 * 79) = 1604
B(6) 7215 + (6 * 79) = 7689
B(7) 71 + (7 * 79) = 624
B(8) 62 + (8 * 79) = 694

Les valeurs de phase P(0) à P(8), les valeurs de base B(0) à B(8) et les neuf nombres premiers sont utilisés de la manière indiquée au C.6.4.3. Une séquence PRS identique est produite et soustraite (modulo 10) de TKx, ce qui donne MPx.

Les résultats des opérations susmentionnées, qui permettent de déterminer MPx, 4314920574868366, à partir de TKx, 5371333066610533, sont donnés au Tableau C.3.

Tableau C.3/T.36 – Calculs effectués en Y pour déterminer MPx par déchiffrement de TKx

Nombre premier B(n) P(n)	32603 1209 7162	32507 7294 8683	32183 7320 840	32003 8819 1512	31847 954 7619	31607 1604 7667	31583 7689 9188	31547 624 713	31259 694 846	Total	Dernier chiffre	Clé de transfert	**Primitive mutuelle**
	19063	10166	1847	21080	7410	2745	26944	3254	24462	116971	1	5	**4**
	29449	2337	3180	31096	30953	9607	19519	11488	2991	140620	0	3	**3**
	1365	12410	9291	1917	6993	17019	30758	7343	12660	99756	6	7	**1**
	20135	19052	7441	8439	15299	21635	4758	7717	2261	106737	7	1	**4**
	21377	30370	14484	16566	9320	29661	11148	20264	6184	159374	4	3	**9**
	23217	16082	12078	1859	5967	7709	710	25936	9213	102771	1	3	**2**
	30773	16852	4259	8985	23752	6899	26914	453	16986	135873	3	3	**0**
	4534	9521	22736	31290	16191	3546	9930	30296	3641	131685	5	0	**5**
	4302	11222	9227	16644	419	30131	15659	8051	26134	121789	9	6	**7**
	17241	642	21706	17678	17562	3021	7655	7851	6776	100132	2	6	**4**
	11052	1740	449	15669	2626	9813	20166	9239	13694	84448	8	6	**8**
	27241	13830	4014	27960	21138	31373	15427	23582	900	165465	5	1	**6**
	5339	6799	31584	28128	6501	3948	24038	14266	30679	151282	2	0	**8**
	32060	18731	24391	5579	23636	11192	4466	5730	3847	129632	2	5	**3**
	28176	29500	23019	12590	1068	30799	8353	10709	12803	157017	7	3	**6**
	27252	9167	21075	12803	31615	31462	17978	25999	7726	185077	7	3	**6**
Primitive mutuelle = (OTx){HKM–1}[TKx] = 4314920574868366													

C.6.4.5 Calculs effectués en Y pour déterminer le nombre RCNy

RCNy = (UINy, UCNy & IDx & IDy){HKM+1}[MPx]

En Y, les calculs préliminaires relatifs à (UINy, UCNy & IDx & IDy) sont les mêmes que ceux qui sont effectués par X pour calculer la primitive MPx (voir C.6.4.2), mais ce sont UINy et UCNy qui servent à créer les valeurs de phase et les valeurs de base. IDx et IDy servent à modifier P(0) à P(3) et B(0) à B(3) comme indiqué précédemment. L'algorithme HKM+1 utilise les valeurs de phase et de base ainsi que les neuf premiers nombres premiers servant d'arguments de modulos pour créer une séquence PRS (modulo 10) qui, ajoutée à MPx, produit le nombre RCNy, qui peut être communiqué ouvertement.

Voir ci-après les résultats des calculs effectués avec les valeurs de test.

Numéro d'identité unique, UINy = 97355769383778314835370916743672287344981976735775982475786494 67
Nombre crypté unique, UCNy = 7598247578649467

Primitive = 9735576938377831483537091674367228734498197673577598247578649467

P(0) 9735 + (0 * 101) = 9735
P(1) 5769 + (1 * 101) = 5870
P(2) 3837 + (2 * 101) = 4039
P(3) 7831 + (3 * 101) = 8134

B(0) 2873 + (0 * 79) = 2873
B(1) 4498 + (1 * 79) = 4577
B(2) 1976 + (2 * 79) = 2134
B(3) 7357 + (3 * 79) = 7594

P(4) 4835 + (4 * 101) = 5239
P(5) 3709 + (5 * 101) = 4214
P(6) 1674 + (6 * 101) = 2280
P(7) 36 + (7 * 101) = 743
P(8) 72 + (8 * 101) = 880

B(4) 7598 + (4 * 79) = 7914
B(5) 2475 + (5 * 79) = 2870
B(6) 7864 + (6 * 79) = 8338
B(7) 94 + (7 * 79) = 647
B(8) 67 + (8 * 79) = 699

Les valeurs P(0) à P(3) et B(0) à B(3) sont modifiées comme indiqué précédemment à l'aide de IDx et de IDy.
IDx = 642092 IDy = 538249

P(0) = 9735 + 642 = 10377
P(1) = 5870 + 092 = 5962
P(2) = 4039 + 538 = 4577
P(3) = 8134 + 249 = 8383

B(0) = 2873 + 642 = 3515
B(1) = 4577 + 092 = 4669
B(2) = 2134 + 538 = 2672
B(3) = 7594 + 249 = 7843

Les résultats des opérations susmentionnées, qui permettent de produire RCNy, 9865418902725854, à partir de MPx, 43149205748688366, sont donnés au Tableau C.4.

Tableau C.4/T.36 – Calculs effectués en Y pour déterminer le nombre RCNy

Nombre premier B(n) P(n)	32603 3515 10377	32507 4669 5962	32183 2672 4577	32003 7843 8383	31847 7914 5239	31607 2870 4214	31583 8338 2280	31547 647 743	31259 699 880	Total	Dernier chiffre	Primitive mutuelle	**Nombre crypté enregistré**
	25001	10586	204	13707	28499	20306	29257	7516	21199	156275	5	4	**9**
	13430	15394	30160	5924	632	26519	29357	4614	1335	127365	5	3	**8**
	29909	1609	1288	25579	1669	31481	10416	19840	26654	148445	5	1	**6**
	18063	3304	30138	21293	23808	17664	26941	28398	782	170391	1	4	**5**
	13404	18058	6870	9345	9660	29659	15762	13152	15215	131125	5	9	**4**
	3725	22151	12330	5965	16440	3679	6693	23201	7225	101409	9	2	**1**
	19572	18252	22551	27112	11165	1992	30656	26222	17576	175098	8	0	**8**
	3250	17741	9696	11484	16232	27780	8509	24895	837	120424	4	5	**9**
	12700	4893	397	12570	21097	15746	12624	18095	22401	120523	3	7	**0**
	6993	25503	30928	17270	19684	24617	24356	3528	28799	181678	8	4	**2**
	30336	366	25855	11914	15499	9145	1638	11232	30964	136949	9	8	**7**
	19230	18490	19842	24745	16289	12340	13788	11294	12608	148626	6	6	**2**
	7431	23725	12423	8843	26337	15960	2224	19861	29213	146017	7	8	**5**
	4962	20676	13583	5148	24250	6657	4491	10438	7760	97965	5	3	**8**
	31428	22961	23535	19981	4478	14962	20103	2328	16433	156209	9	6	**5**
	10456	29330	32121	24295	25028	18634	7833	23507	14614	185818	8	6	**4**
Nombre crypté enregistré, RCNy = (UINy, UCNy & IDx & IDy) {HKM+1} [MPx] = 9865418902725854													

C.6.5 Mode sécurisé

C.6.5.1 Procédure procSTKxy en notation algébrique

Les sous-paragraphes suivants montrent, à l'aide de valeurs de test, tous les calculs effectués dans la procédure procSTKxy, à l'aide de l'algorithme HKMD, pour transférer la clé SKx entre X et Y. La clé secrète peut être spécifiquement la clé SCn, SRn, SSn, etc.

X
> UINx, UCNx, RCNy <

(UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy] = MPx
(MPx & RNKx){HKMD+1}[SKx] = ESSKx
RCNy, RNKx, ESSKx >>>>>>

Y
> UINy, UCNy <

>>>>>> RCNy, RNKx, ESSKx
MPx = (UINy, UCNy & IDx & IDy){HKM–1}[RCNy]
SKx = (MPx & RNKx){HKMD–1}[ESSKx]

C.6.5.2 Calculs effectués en X pour recréer la primitive MPx

> UINx, UCNx, RCNy <
MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy]

Les valeurs de test et les calculs, identiques à ceux qui sont indiqués au C.6.4.2, donnent:

MPx = (UINx, UCNx & IDx & IDy){HKM+1}[UCNx & IDx & IDy] = 4314920574868366

C.6.5.3 Calculs effectués en X pour créer la clé ESSKx à l'aide de HKMD+1

ESSKx = (MPx & RNKx){HKMD+1}[SKx]

La clé SKx est chiffrée deux fois à l'aide de HKMD+1, la première opération de chiffrement étant un processus d'embrouillage fondé sur une séquence PRS de 12 chiffres (modulo 12) produite avec HKM et destiné à créer SSKx, et la seconde opération étant l'addition normale (modulo 10) au "message" SSKx de la séquence PRS (modulo 10) créée avec HKM.

C.6.5.3.1 Calculs effectués en X pour créer la clé SSKx

La primitive MPx est reproduite et concaténée de manière à former la primitive de 64 chiffres qui permet de déterminer les valeurs primitives de phase et de base indiquées ci-après.

Primitive = 4314920574868366 4314920574868366 4314920574868366 4314920574868366

P(0)	4314 + (0 * 101) = 4314	B(0)	4314 + (0 * 79) = 4314
P(1)	9205 + (1 * 101) = 9306	B(1)	9205 + (1 * 79) = 9284
P(2)	7486 + (2 * 101) = 7688	B(2)	7486 + (2 * 79) = 7644
P(3)	8366 + (3 * 101) = 8669	B(3)	8366 + (3 * 79) = 8603
P(4)	4314 + (4 * 101) = 4718	B(4)	4314 + (4 * 79) = 4630
P(5)	9205 + (5 * 101) = 9710	B(5)	9205 + (5 * 79) = 9600
P(6)	7486 + (6 * 101) = 8092	B(6)	7486 + (6 * 79) = 7960
P(7)	83 + (7 * 101) = 790	B(7)	83 + (7 * 79) = 636
P(8)	66 + (8 * 101) = 874	B(8)	66 + (8 * 79) = 698

Le nombre RNKx, associé à SKx, est subdivisé en deux paires de deux chiffres; la première paire est ajoutée à P(0) et la seconde à P(1), ce qui donne les nouvelles valeurs de P(0) et de P(1); les nouvelles valeurs de B(0) et de B(1) sont créées de manière identique.

RNKx = 3958

P(0) = 4314 + 39 = 4353	B(0) = 4314 + 39 = 4353
P(1) = 9306 + 58 = 9364	B(1) = 9284 + 58 = 9342

Les nombres premiers, les valeurs de phase et les valeurs de base servent ensuite, avec HKM, à produire une séquence PRS de 12 chiffres, (modulo 12) + 1 (autrement dit, chiffre de la colonne "Total" modulo 12 plus 1, ce qui donne la séquence PRS). Les résultats des calculs sont donnés au Tableau C.5.

Tableau C.5/T.36 – Calculs effectués en X pour créer la séquence d'embrouillage/désembrouillage PRS (modulo 12) + 1

Nombre premier B(n) P(n)	32603 4353 4353	32507 9342 9364	32183 7644 7688	32003 8603 8669	31847 4630 4718	31607 9600 9710	31583 7960 8092	31547 636 790	31259 698 874	Total	**(modulo 12) + 1 PRS**
	6266	2151	914	12417	29145	6957	14583	29235	16131	117799	**8**
	19790	5316	2905	29440	5611	1609	13155	12277	6198	96301	**2**
	8744	23883	31733	578	23625	22184	16155	16063	12462	155427	**4**
	14931	19445	3781	12069	21152	30041	19407	26387	8474	155687	**12**
	16864	6074	1630	11875	4235	11332	7267	30675	6901	96853	**2**
	19639	18593	4899	7049	22145	27513	16847	13254	3012	132951	**4**
	3501	10905	19127	28865	15857	16708	702	6495	8023	110183	**12**
	14252	30079	31602	14318	10575	22882	29312	29710	4693	187423	**8**
	28050	7510	90	30210	13411	30157	19899	30454	24778	184559	**12**
	3415	8314	12117	267	23127	18687	7295	30433	8817	112472	**9**
	31130	10165	31857	24788	8396	25475	18646	17077	27502	195036	**1**
	10822	8483	18330	15175	20140	16641	13643	8804	3370	115408	**5**

La séquence d'embrouillage/désembrouillage PRS ainsi obtenue sert ensuite à embrouiller la clé SKx. Le premier chiffre de PRS indique quel chiffre de SKx doit être permuté avec le premier chiffre de SKx. Le deuxième chiffre de PRS indique la permutation à faire pour le deuxième chiffre de SKx et ainsi de suite pour les chiffres restants. Les résultats du processus d'embrouillage qui permet d'obtenir SSKx, 721647935700, à partir de SKx, 309126704577, sont reproduits ci-après.

Séquence d'embrouillage PRS			
8	**3**091267**0**4577	=	SKx
2	0**0**9126734577		
4	00**91**26734577		
12	001**9**26734577		
2	0**0**172**2**6734579		
4	021**7**0**6**734579		
12	0216077345**79**		
8	021607934577		
12	0216079345**77**		
9	021607937574		
1	**0**21607935774		
5	7216**0**7935704		
	721647935700	=	**SSKx**

C.6.5.3.2 Calculs effectués en X pour créer la clé ESSKx

Les mêmes calculs que ceux qui sont effectués à l'aide de HKM au C.6.5.3.1 servent à créer une séquence de 12 chiffres PRS (modulo 10), qui est ajoutée (modulo 10) à SSKx pour produire ESSKx.

Les résultats des calculs effectués pour obtenir ESSKx, 638378264968, à partir de SSKx, 721647935700, sont donnés au Tableau C.6.

Tableau C.6/T.36 – Calculs effectués en X pour créer la clé ESSKx

Nombre premier B(n) P(n)	32603 4353 4353	32507 9342 9364	32183 7644 7688	32003 8603 8669	31847 4630 4718	31607 9600 9710	31583 7960 8092	31547 636 790	31259 698 874	Total	Dernier chiffre	Clé secrète embrouillée	**Clé secrète embrouillée chiffrée**
	6266	2151	914	12417	29145	6957	14583	29235	16131	117799	9	7	**6**
	19790	5316	2905	29440	5611	1609	13165	12277	6198	96301	1	2	**3**
	8744	23883	31733	578	23625	22184	16155	16063	12462	155427	7	1	**8**
	14931	19445	3781	12069	21152	30041	19407	26387	8474	155687	7	6	**3**
	16864	6074	1630	11875	4235	11332	7267	30675	6901	96853	3	4	**7**
	19639	18593	4899	7049	22145	27513	16847	13254	3012	132951	1	7	**8**
	3501	10905	19127	28865	15857	16708	702	6495	8023	110183	3	9	**2**
	14252	30079	31602	14318	10575	22882	29312	29710	4693	187423	3	3	**6**
	28050	7510	90	30210	13411	30157	19899	30454	24778	184559	9	5	**4**
	3415	8314	12117	267	23127	18687	7295	30433	8817	112472	2	7	**9**
	31130	10165	31857	24788	8396	25475	18646	17077	27502	195036	6	0	**6**
	10822	8483	18330	15175	20140	16641	13643	8804	3370	115408	8	0	**8**
SSKx = 638378264968 X envoie RCNy, ESSKx et RNKx à Y.													

C.6.5.4 Calculs effectués en Y pour déterminer la clé SKx

Y
> UINy, UCNy <
>>>>>> RCNy, RNKx, ESSKx
MPx = (UINy, UCNy & IDx & IDy){HKM–1}[RCNy]
SKx = (MPx & RNKx){HKMD–1}[ESSKx]

C.6.5.4.1 Calculs effectués en Y pour déterminer la primitive MPx à partir du nombre RCNy

MPx = (UINy, UCNy & IDx & IDy){HKM–1}[RCNy]

Les valeurs de test et les calculs sont identiques à ceux qui sont indiqués au C.6.4.5, sauf que le processus de déchiffrement HKM–1 permet de déterminer MPx à partir de RCNy en soustrayant de RCNy, chiffre par chiffre, la séquence de 16 chiffres PRS.

Les résultats des calculs effectués à l'aide des valeurs de test figurent ci-après.

UINy = 9735576938377831483537091674367228734498197673577598247578649467

UCNy = 7598247578649467

Primitive = 9735576938377831483537091674367228734498197673577598247578649467

P(0)	9735 + 0 * 101 = 9735	B(0)	2873 + 0 * 79 = 2873
P(1)	5769 + 1 * 101 = 5870	B(1)	4498 + 1 * 79 = 4577
P(2)	3837 + 2 * 101 = 4039	B(2)	1976 + 2 * 79 = 2134
P(3)	7831 + 3 * 101 = 8134	B(3)	7357 + 3 * 79 = 7594
P(4)	4835 + 4 * 101 = 5239	B(4)	7598 + 4 * 79 = 7914
P(5)	3709 + 5 * 101 = 4214	B(5)	2475 + 5 * 79 = 2870
P(6)	1674 + 6 * 101 = 2280	B(6)	7864 + 6 * 79 = 8338
P(7)	36 + 7 * 101 = 743	B(7)	94 + 7 * 79 = 647
P(8)	72 + 8 * 101 = 880	B(8)	67 + 8 * 79 = 699

Les valeurs P(0) à P(3) et B(0) à B(3) sont modifiées comme indiqué précédemment à l'aide de IDx et de IDy.

IDx = 642092 IDy = 538249

P(0) = 9735 + 642 = 10377	B(0) = 2873 + 642 = 3515
P(1) = 5870 + 092 = 5962	B(1) = 4577 + 092 = 4669
P(2) = 4039 + 538 = 4577	B(2) = 2134 + 538 = 2672
P(3) = 8134 + 249 = 8383	B(3) = 7594 + 249 = 7843

Les résultats de l'emploi, dans l'algorithme HKM–1, de ces valeurs primitives de phase et de base conjointement avec les neuf nombres premiers pour créer MPx, 43149205748688366, à partir de RCNy, 9865418902725854, sont donnés au Tableau C.7.

Tableau C.7/T.36 – Calculs effectués en Y pour déterminer la primitive MPx à partir de RCNy

Nombre premier B(n) P(n)	32603 3515 10377	32507 4669 5962	32183 2672 4577	32003 7843 8383	31847 7914 5239	31607 2870 4214	31583 8338 2280	31547 647 743	31259 699 880	Total	Dernier chiffre	Nombre crypté enregistré	**Primitive mutuelle**
	25001	10586	204	13707	28499	20306	29257	7516	21199	156275	5	9	**4**
	13430	15394	30160	5924	632	26519	29357	4614	1335	127365	5	8	**3**
	29909	1609	1288	25579	1669	31481	10416	19840	26654	148445	5	6	**1**
	18063	3304	30138	21293	23808	17664	26941	28398	782	170391	1	5	**4**
	13404	18058	6870	9345	9660	29659	15762	13152	15215	131125	5	4	**9**
	3725	22151	12330	5965	16440	3679	6693	23201	7225	101409	9	1	**2**
	19572	18252	22551	27112	11165	1992	30656	26222	17576	175098	8	8	**0**
	3250	17741	9696	11484	16232	27780	8509	24895	837	120424	4	9	**5**
	12700	4893	397	12570	21097	15746	12624	18095	22401	120523	3	0	**7**
	6993	25503	30928	17270	19684	24617	24356	3528	28799	181678	8	2	**4**
	30336	366	25855	11914	15499	9145	1638	11232	30964	136949	9	7	**8**
	19230	18490	19842	24745	16289	12340	13788	11294	12608	148626	6	2	**6**
	7431	23725	12423	8843	26337	15960	2224	19861	29213	146017	7	5	**8**
	4962	20676	13583	5148	24250	6657	4491	10438	7760	97965	5	8	**3**
	31428	22961	23535	19981	4478	14962	20103	2328	16433	156209	9	5	**6**
	10456	29330	32121	24295	25028	18634	7833	23507	14614	185818	8	4	**6**
MPx = (UINy, UCNy & IDx & IDy){HKM–1}[RCNy] = 4314920574868366													

C.6.5.4.2 Calculs effectués en Y pour déterminer SKx à partir de ESSKx à l'aide de HKMD–1

SKx = (MPx & RNKx){HKMD–1}[ESSKx]

La clé ESSKx est déchiffrée deux fois à l'aide de HKMD–1 dans l'ordre inverse de celui du double chiffrement effectué en X. La première opération de déchiffrement consiste à soustraire (modulo 10) du "message" ESSKx la même séquence que la séquence PRS (modulo 10) utilisée par X. La seconde opération est un processus de désembrouillage fondé sur la même séquence que la séquence à 12 chiffres PRS (modulo 10) + 1 utilisée par X.

C.6.5.4.2.1 Calculs effectués en Y lors de la première opération de déchiffrement de ESSKx pour déterminer SSKx

Les mêmes calculs que ceux qui sont effectués à l'aide de HKM comme indiqué au C.6.5.3.2 permettent de créer une séquence de 12 chiffres PRS (modulo 10), qui est soustraite (modulo 10) de ESSKx pour produire SSKx.

La primitive MPx est reproduite et concaténée de manière à créer la primitive de 64 chiffres utilisée pour déterminer les valeurs primitives de phase et de base, comme indiqué ci-après.

Primitive = 4314920574868366 4314920574868366 4314920574868366 4314920574868366

P(0)	4314 + (0 * 101) = 4314	B(0)	4314 + (0 * 79) = 4314
P(1)	9205 + (1 * 101) = 9306	B(1)	9205 + (1 * 79) = 9284
P(2)	7486 + (2 * 101) = 7688	B(2)	7486 + (2 * 79) = 7644
P(3)	8366 + (3 * 101) = 8669	B(3)	8366 + (3 * 79) = 8603
P(4)	4314 + (4 * 101) = 4718	B(4)	4314 + (4 * 79) = 4630
P(5)	9205 + (5 * 101) = 9710	B(5)	9205 + (5 * 79) = 9600
P(6)	7486 + (6 * 101) = 8092	B(6)	7486 + (6 * 79) = 7960
P(7)	83 + (7 * 101) = 790	B(7)	83 + (7 * 79) = 636
P(8)	66 + (8 * 101) = 874	B(8)	66 + (8 * 79) = 698

Le nombre RNKx, associé à ESSKx, est subdivisé en deux paires de deux chiffres; la première paire est ajoutée à P(0) et la seconde à P(1), ce qui donne les nouvelles valeurs de P(0) et de P(1); les nouvelles valeurs de B(0) et de B(1) sont créées de manière identique.

RNKx = 3958

P(0) = 4314 + 39 = 4353	B(0) = 4314 + 39 = 4353
P(1) = 9306 + 58 = 9364	B(1) = 9284 + 58 = 9342

Les résultats des calculs effectués pour déchiffrer ESSKx, 638378264968 et créer SSKx, 721647935700, sont donnés au Tableau C.8.

Tableau C.8/T.36 – Calculs effectués en Y pour créer SSKx à partir de ESSKx

Nombre premier B(n) P(n)	32603 4353 4353	32507 9342 9364	32183 7644 7688	32003 8603 8669	31847 4630 4718	31607 9600 9710	31583 7960 8092	31547 636 790	31259 698 874	Total	Dernier chiffre	Clé secrète embrouillée chiffrée	**Clé secrète embrouillée**
	6266	2151	914	12417	29145	6957	14583	29235	16131	117799	9	6	**7**
	19790	5316	2905	29440	5611	1609	13165	12277	6198	96301	1	3	**2**
	8744	23883	31733	578	23625	22184	16155	16063	12462	155427	7	8	**1**
	14931	19445	3781	12069	21152	30041	19407	26387	8474	155687	7	3	**6**
	16864	6074	1630	11875	4235	11332	7267	30675	6901	96853	3	7	**4**
	19639	18593	4899	7049	22145	27513	16847	13254	3012	132951	1	8	**7**
	3501	10905	19127	28865	15857	16708	702	6495	8023	110183	3	2	**9**
	14252	30079	31602	14318	10575	22882	29312	29710	4693	187423	3	6	**3**
	28050	7510	90	30210	13411	30157	19899	30454	24778	184559	9	4	**5**
	3415	8314	12117	267	23127	18687	7295	30433	8817	112472	2	9	**7**
	31130	10165	31857	24788	8396	25475	18646	17077	27502	195036	6	6	**0**
	10822	8483	18330	15175	20140	16641	13643	8804	3370	115408	8	8	**0**
SSKx = 721647935700													

C.6.5.4.2.2 Calculs effectués en Y pour déterminer la clé SKx à partir de la clé SSKx

Les mêmes calculs que ceux qui ont servi en X à produire PRS (modulo 12) + 1 (voir C.6.5.3.1) sont effectués en Y de manière à créer la même séquence d'embrouillage/désembrouillage PRS, 8 2 4 12 2 4 12 8 12 9 1 5. Cette séquence PRS est inversée et permet de désembrouiller SSKx.

La séquence de désembrouillage PRS est 5 1 9 12 8 12 4 2 12 4 2 8.

Le premier chiffre de cette séquence de désembrouillage indique quel chiffre de SSKx doit être permuté avec le douzième chiffre de SSKx. Le deuxième chiffre de PRS indique la permutation à effectuer pour le onzième chiffre de SSKx et ainsi de suite pour les chiffres restants. Les différentes étapes de ce processus de désembrouillage qui permet de créer SKx, 309126704577, à partir de SSKx, 721647935700, sont décrites ci-après.

Séquence de désembrouillage PRS			
5	7216**4**7935700	=	SSKx
1	**7**2160793570**4**		
9	0216079357**7**4		
12	021607937574		
8	0216079345**7**7		
12	021607934577		
4	021607734579		
2	021706734579		
12	001726734579		
4	001926734577		
2	009126734577		
8	009126734577		
	309126704577	=	**SKx**

X et Y sont maintenant en possession de la même clé SKx.

C.6.6 Utilisation de l'algorithme HKM en mode sécurisé

L'algorithme HKM utilisé en mode sécurisé permet de recréer MPx et MPy en X et en Y. Il permet également, à l'aide des procédures procSTKxy et procSTKyx, de sécuriser le transfert des clés secrètes en vue de l'authentification mutuelle, de la définition d'une clé de session secrète garantissant la confidentialité ou l'intégrité des messages, de la confirmation de la réception des messages et de la confirmation ou de la réfutation de l'intégrité des messages.

Les procédures qui permettent d'assurer ces fonctions sont décrites au C.5.

Annexe D

Procédures pour l'utilisation du système de chiffrement HFX40 visant à assurer la confidentialité des messages et la sécurité de transmission des documents par télécopie

D.1 Domaine d'application

La présente annexe décrit le système de chiffrement HFX40 à utiliser sur les télécopieurs pour assurer la confidentialité des messages. Des exemples de calculs effectués à l'aide de valeurs de test sont donnés au D.3 et peuvent servir à vérifier la mise en application de la présente annexe.

Le système de chiffrement HFX40 est destiné à être utilisé avec tous les types de télécopieurs spécialisés, mais est également applicable aux systèmes de télécopie intégrés à des ordinateurs.

Le système HFX40 repose sur l'utilisation de 19 nombres premiers système, qui sont également employés avec le système de gestion de clés HKM décrit à l'Annexe C et l'algorithme de hachage assurant l'intégrité des messages décrit à l'Annexe E. La présente annexe ne traite toutefois pas du système de gestion de clés ni de l'algorithme de hachage permettant d'assurer l'intégrité du message principal.

Des exemples de calculs sont donnés au D.3 et peuvent servir à vérifier la mise en application de la présente annexe.

Le système HFX40 est visé par des droits de propriété intellectuelle; toutefois, le détenteur de ces droits est convenu d'observer le code de pratique du TSB. Des renseignements complémentaires peuvent être obtenus auprès du TSB.

D.2 Description de l'algorithme HFX40 à utiliser sur les télécopieurs en mode sécurisé

L'algorithme HFX40 permet d'assurer la confidentialité des messages par le biais d'un chiffrement. A cet effet, il emploie une clé secrète pour fournir les nombres qui serviront à effectuer des calculs de congruences. Les arguments de modulos proviennent de trois nombres premiers choisis dans un ensemble de 19 nombres premiers système mémorisés dans le télécopieur et identiques aux nombres premiers utilisés par le système de gestion de clés HKM (Annexe C) et par le système assurant l'intégrité des messages HFX40-I (Annexe E).

On obtient grâce aux calculs de congruences des séquences PRS longues qui permettent de chiffrer le message de télécopie compressé. Celui-ci ne peut être recréé qu'au moyen de la clé de chiffrement secrète.

L'algorithme HFX40 se sert d'une clé de 12 chiffres décimaux équivalant à peu près à une longueur de 40 bits.

Les procédures de gestion de clés sont décrites en détail à l'Annexe C.

Si un enregistrement mutuel a été effectué entre X et Y, le mode sécurisé peut être utilisé pour procéder à une authentification mutuelle de X et de Y, la clé SSx pouvant ensuite être créée en X et transférée en mode sécurisé à Y. S'il n'y a pas eu d'enregistrement, on peut recourir au mode outrepassement (voir la Recommandation T.30) pour autoriser les utilisateurs en X et en Y à introduire manuellement une clé de session secrète prédéfinie pour créer SSx.

A l'aide de la clé SSx et de l'algorithme HFX40, X produit trois séquences PRS (modulo 2) qui sont mémorisées dans trois tableaux P, Q et R. Le nombre d'éléments mémorisés dans les tableaux varie avec chaque tableau. Le tableau P est créé avec 1021 éléments, le tableau Q avec 1019 éléments et le tableau R avec 1013 éléments.

Les quatre derniers éléments de chaque tableau permettent de créer un multiplexeur qui modifie les éléments du tableau à mesure que le message est chiffré, les tableaux réduits étant utilisés pour chiffrer le message. Le tableau P comprend maintenant 1017 éléments, le tableau Q 1015 éléments et le tableau R 1009 éléments.

Le premier bit du message de télécopie compressé est ajouté (modulo 2) à la somme (modulo 2) des bits des premiers éléments des tableaux P, Q et R, ce qui donne le premier bit du message chiffré. Après que chaque bit du message a été traité, les éléments correspondants des tableaux P, Q et R sont modifiés en fonction des éléments du multiplexeur. La modification de chaque tableau correspond à une permutation entre l'élément de ce tableau et un élément du multiplexeur qui est déterminé par les éléments des deux autres tableaux.

L'élément du tableau P est permuté avec l'élément du multiplexeur déterminé par les tableaux Q et R.

L'élément du tableau Q est permuté avec l'élément du multiplexeur déterminé par les tableaux R et P.

L'élément du tableau R est permuté avec l'élément du multiplexeur déterminé par les tableaux P et Q.

Il convient de noter que l'ordre des opérations est important.

Le deuxième bit du message principal est traité de la même manière par rapport aux bits des deuxièmes éléments des tableaux. Le nouveau multiplexeur créé par permutation des premiers éléments des tableaux permet ensuite de procéder à une permutation avec les deuxièmes éléments des tableaux.

La procédure susmentionnée est appliquée pour chacun des bits du message. Lorsque l'élément 1009 du tableau R a été utilisé, le premier élément (modifié) est le prochain élément à être utilisé. La procédure se poursuit avec les nouveaux éléments. Le même processus recommence pour les tableaux P et Q après utilisation des éléments 1017 et 1015 respectivement.

Le message chiffré est ensuite envoyé à Y. A l'aide de la même clé SSx et de l'algorithme HFX40, Y crée les mêmes tableaux et le même multiplexeur de modification, puis soustrait du message chiffré la somme (modulo 2) des éléments des tableaux pour récupérer le message de télécopie compressé original. La clé SSx utilisée par Y est soit transférée en mode sécurisé par X soit créée à partir de la clé de session prédéfinie introduite manuellement par l'utilisateur en Y dans le mode outrepassement.

D.3 Exemples de calculs effectués avec l'algorithme HFX40

D.3.1 Introduction

Le présent sous-paragraphe décrit l'algorithme HFX40 en indiquant les nombres qui doivent être mémorisés, ainsi que les règles à suivre pour les calculs effectués à l'aide de ces nombres afin de créer les séquences PRS permettant de chiffrer le message de télécopie compressé. Ces calculs peuvent servir à vérifier l'implémentation de l'algorithme HFX40.

D.3.2 Informations mémorisées

Tous les terminaux sont dotés du même ensemble de 19 nombres premiers système qui serviront à effectuer des calculs de congruences.

32603 32507 32183 32003 31847 31607 31583 31547 31259
31139 30803 30539 30467 30347 30323 30203 29879 29759 29663

D.3.3 Choix des nombres premiers

Pour chaque communication par télécopie, une clé SSx différente permet de sélectionner trois nombres premiers parmi les 19 nombres premiers système utilisés dans les calculs de congruences. La clé SSx est divisée en quatre groupes de trois chiffres. On applique une opération OU exclusif aux deux premiers groupes de trois chiffres, g1 et g2, pour produire un troisième groupe, g3, ainsi qu'aux deux seconds groupes de 3 chiffres, g4 et g5, pour former un sixième groupe, g6. On ajoute le nombre 1024 aux groupes g1 à g3 pour créer les valeurs P(0) à P(2) et aux groupes g4 à g6 pour créer les valeurs B(0) à B(2). Cette procédure est représentée ci-après à l'aide d'un exemple numérique.

Dans cet exemple, la clé SSx est 149162536496.

g1 = 149	P(0) = 149 + 1024 = 1173
g2 = 162	P(1) = 162 + 1024 = 1186
g3 = g1 XOR g2 = 55	P(2) = 55 + 1024 = 1079
g4 = 536	B(0) = 536 + 1024 = 1560
g5 = 496	B(1) = 496 + 1024 = 1520
g6 = g4 XOR g5 = 1000	B(2) = 1000 + 1024 = 2024

Trois nombres de huit chiffres sont ensuite formés par concaténation des paires P(0) et B(0), P(1) et B(1), et P(2) et B(2) et par application à chaque nombre d'une opération modulo 19.

P(0) et B(0)	donnent	11731560 (modulo 19) = 10
P(1) et B(1)	donnent	11861520 (modulo 19) = 10
P(2) et B(2)	donnent	10792024 (modulo 19) = 5

Dans la liste des 19 nombres premiers indiqués au D.3.2 ci-dessus, le premier nombre, 32603, est appelé nombre premier (0) et le dernier nombre, 29663, est appelé nombre premier (18).

A l'aide de la première valeur, 10, obtenue à partir de P(0) et B(0), le premier nombre, c'est-à-dire le nombre premier (0), 32603, est permuté avec le nombre premier (10), 30803.

A l'aide de la deuxième valeur, 10, obtenue à partir de P(1) et B(1), le deuxième nombre, c'est-à-dire le nombre premier (1), 32507, est permuté avec le nombre premier (10), 32603.

A l'aide de la troisième valeur, 5, obtenue à partir de P(2) et B(2), le troisième nombre, c'est-à-dire le nombre premier (2), 32183, est permuté avec le nombre premier (5), 31607.

Ce processus est représenté ci-après:

Nombres premiers	32603	32507	32183	32003	31847	31607	31583	31547	31259	
non modifiés	31139	30803	30539	30467	30347	30323	30203	29879	29759	29663
10	**30803**	32507	32183	32003	31847	31607	31583	31547	31259	
	31139	**32603**	30539	30467	30347	30323	30203	29879	29759	29663
10	30803	**32603**	32183	32003	31847	31607	31583	31547	31259	
	31139	**32507**	30539	30467	30347	30323	30203	29879	29759	29663
5	30803	32603	**31607**	32003	31847	**32183**	31583	31547	31259	
	31139	32507	30539	30467	30347	30323	30203	29879	29759	29663

D.3.4 Calculs effectués à l'aide de l'algorithme HFX40 pour créer trois séquences PRS

Les trois premiers nombres premiers de la dernière série, 30803, 32603 et 31607 sont utilisés dans les calculs de congruences effectués avec les trois valeurs primitives de phase et les trois valeurs primitives de base déterminées à partir de SSx au D.3.3, ce qui donne trois séquences PRS (modulo 2) à mémoriser dans les tableaux P, Q et R. Le tableau P comprend 1021 éléments, le tableau Q, 1019 éléments et le tableau R, 1013 éléments.

Exemples de calculs effectués pour le premier "jeu", c'est-à-dire le nombre premier (0) = 30803, la valeur de phase P(0) = 1173 et la valeur de base B(0) = 1560:

P(0) est multiplié par B(0):
1173 * 1560 = 1829880
1829880 [modulo nombre premier (0)] = 1829880 (modulo 30803) = 12503
12503 (modulo 2) = 1

Le nombre 12503 est ensuite utilisé comme nouvelle valeur de phase et multiplié par B(0):

12503 * 1560 = 19504680
19504680 [modulo nombre premier (0)] = 19504680 (modulo 30803) = 6381
6381 (modulo 2) = 1

Le processus est appliqué à nouveau et on obtient ainsi 1021 valeurs qui seront mémorisées dans le tableau P.

Des calculs similaires sont effectués avec les deuxième et troisième "jeux" de valeurs de phase, de valeurs de base et de nombres premiers, de manière à créer les éléments qui seront mémorisés dans les tableaux Q et R. Le Tableau D.1 indique 16 autres séries de calculs effectués à l'aide de HFX40.

Tableau D.1/T.36 – Calculs effectués à l'aide de HFX40 pour créer trois séquences PRS

Phase	B(0)	Nouvelle phase (mod 30803)	Tableau P (mod 2)	Phase	B(0)	Nouvelle phase (mod 32603)	Tableau Q (mod 2)	Phase	*B(0)	Nouvelle phase (mod 31607)	Tableau R (mod 2)
1173	1560	12503	1	1186	1520	9555	1	1079	2024	3013	1
12503	1560	6381	1	9555	1520	15265	1	3013	2024	29768	0
6381	1560	4991	1	15265	1520	22067	1	29768	2024	7490	0
4991	1560	23604	0	22067	1520	25956	0	7490	2024	20007	1
23604	1560	12655	1	25956	1520	3490	0	20007	2024	5601	1
12655	1560	27780	0	3490	1520	23114	0	5601	2024	21118	0
27880	1560	29767	1	23114	1520	19849	1	21118	2024	10168	0
29767	1560	16399	1	19849	1520	12705	1	10168	2024	3875	1
16399	1560	15950	0	12705	1520	10624	0	3875	2024	4464	0
15950	1560	23979	1	10624	1520	9995	1	4464	2024	27141	1
23979	1560	12398	0	9995	1520	32005	1	27141	2024	418	0
12398	1560	27399	1	32005	1520	3924	0	418	2024	24250	0
27399	1560	18679	1	3924	1520	30734	0	24250	2024	27936	0
18679	1560	30405	1	30734	1520	28184	0	27936	2024	29148	0
30405	1560	25983	1	28184	1520	31941	1	29148	2024	16890	0
25983	1560	27535	1	31941	1520	4453	1	16890	2024	18193	1
:	:	:	:	:	:	:	:	:	:	:	
:	:	:	:	:	:	:	:	:	:	:	

Les tableaux P, Q et R sont représentés dans leur intégralité au Tableau D.2.

Les quatre derniers éléments de chaque tableau servent à créer les éléments initiaux d'un multiplexeur qui, utilisé avec une "table de vérité", permet de modifier les éléments des tableaux P, Q et R à mesure que le message est chiffré.

Multiplexeur			**"Table de vérité"**	
P	**Q**	**R**		
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
0	1	1	1	1

Les tableaux réduits, c'est-à-dire le tableau P avec 1017 éléments, le tableau Q avec 1015 éléments et le tableau R avec 1009 éléments, permettent de chiffrer le message, comme expliqué ci-après.

D.3.5 Utilisation des tableaux pour chiffrer le message et du multiplexeur pour modifier les tableaux

Le premier bit du message principal est ajouté (modulo 2) à la somme (modulo 2) des premiers éléments de chacun des tableaux réduits P, Q et R de manière à former le premier bit du message chiffré. Le deuxième bit du message principal est traité de la même manière par rapport aux deuxièmes éléments des tableaux et ainsi de suite. Après que chaque bit du message a été traité, les éléments correspondants des tableaux P, Q et R sont modifiés en fonction des éléments mémorisés dans le multiplexeur. La modification de chaque tableau correspond à une permutation entre l'élément de ce tableau et un élément du multiplexeur qui est déterminé par les éléments des deux autres tableaux.

Tableau D.2/T.36 – Tableaux P, Q et R complets

Tableau P (1021 éléments)

1110101101011111000101000111001001011011000101001111110000000000100110111000011011000011101111
11011011000001111000011110110101110101101001000100110110100101100010000000101000010010111010101
1100001001010010011000010011100010010111100010111100110000000110101111111111001111010001111111
1000110110010111110101001001101101111100001111010111011011110100011000001011010100011010011111
0001011111010011111011000110011111010001010100100010100110011001100100011001100110011110110000
1011000111001110010001011110111011010000000100001010101010111011010010001000000011010011110100
01100000000000001000110011011101110100100000011010111000000011101000010011100100101100011101111
10101010011111111001111110010000110001000111000001011111100011011011011100100111100011101111100
0010111001100100101110111001010111001001101011000100101101000011000110000000000110001010111010
111101110100011100000000010100001010100000000111001111111111101110110001011100011010101001000110
100111101010100000000101011100001111001010101000010010111111111100000000110111 1110

Tableau Q (1019 éléments)

11100011011000111010000011001110110000111001111111110000010101111000011111110001011111100001111
11010111111001011011111111110011101101101001111000101011000101001101101100000110100100110101010
0111100001001111101011110010010001111101101011110100110111000100111011000000111011001101110001
1011101000101010110100110010011100001011110010000111110110110001000010011010111001000010110100
0000000000001010011110010001110010111110010000101110010111100110011111100111101101010000010101
0111110111011001110011011110111101001100100011000110101000101011001000101000110111100000000001
0111100101101011010110001110111100111101100010110011001100011101101010101001011110100011000010
0101101000111010010111100100011111110001110000111111100101011111100111000111100011100101110000
10010000010110011111111100001000101000101101100010100010101000100010011100111010010011001010000
0010011100101011001101001101010100101001110110100000010000100101000100101110111011011011000010
000111000101011101111011001110110010011101110110000100011101100011101011100 1111

Tableau R (1013 éléments)

1001100101000001101111100011010101110101000110100101001100110001101100011010111000001100001100
0111000111101000010100100110100111100000000101001100011011010011000001001001000010111101001101
1101001010110100111110101111011100000000110101111111101111100010111000000110011010100001011010
1110000101110101100101101011101010100111101001110110111000001111101111010101000111100000111111
1010001101101100111011011111111000100000011101101001010001010011100101100110001110000000101111
01100100101000001001110100001000001100100110010111010110110011001000111101000000001000011110001
00111011101010110001100011101100011001011001001101000100010011000101110011101111100111101110101
11001011100011010000001011100110010111000000000001001101010110111001111101110010011000011111011
10011110001110001101110010101010001001011011000110001110011010100001000011110000001100101101010
0000001001100000000111011111110111000111101111111000111010100010100100110001111011010011010011010
100100110110110110001010110101111000001111000011010101110001000110100 1111

Une fois qu'un jeu d'éléments des tableaux P, Q et R a été utilisé pour chiffrer un bit du message:

– l'élément du tableau P est permuté avec l'élément du multiplexeur déterminé par les éléments des tableaux Q et R;

– l'élément du tableau Q est permuté avec l'élément du multiplexeur déterminé par les éléments des tableaux R et P;

– l'élément du tableau R est permuté avec l'élément du multiplexeur déterminé par les éléments des tableaux P et Q.

Il convient de noter que l'ordre des opérations est important.

L'exemple suivant qui utilise les sept premiers éléments des tableaux P, Q et R montre en détail le déroulement de la procédure.

Les sept premiers éléments des tableaux initiaux P, Q et R sont:

P	Q	R
1	1	1
1	1	0
1	1	0
0	0	1
1	0	1
0	0	0
1	1	0

Les éléments initiaux du multiplexeur et la table de vérité sont:

P	Q	R	Table de vérité	
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
0	1	1	1	1

Les premiers éléments des tableaux P, Q et R sont 1, 1 et 1 respectivement. Le premier élément du tableau P est permuté en fonction des premiers éléments des tableaux Q et R, qui sont 1 et 1. Au moyen de la "table de vérité", le premier élément du tableau P est permuté avec l'élément de la colonne P du multiplexeur qui correspond à (1, 1) de la "table de vérité", qui est 0 dans ce cas.

Le premier élément de Q est permuté avec l'élément de la colonne Q du multiplexeur correspondant à (1, 1), qui est 1 dans ce cas.

De même, le premier élément de R est permuté avec l'élément de la colonne R du multiplexeur correspondant à (1, 1), qui est 1 dans ce cas.

Il en résulte que les premiers éléments des tableaux P, Q et R, c'est-à-dire 1, 1 et 1, deviennent 0, 1 et 1 respectivement. Ces éléments seront utilisés la prochaine fois que le premier élément d'un tableau sera utilisé.

Après les permutations susmentionnées, les éléments du multiplexeur sont:

P	Q	R
1	1	1
1	1	1
1	1	1
1	1	1

Les nouveaux éléments du multiplexeur sont ensuite utilisés avec les deuxièmes éléments des tableaux P, Q et R exactement de la même manière. Il en résulte que les deuxièmes éléments des tableaux P, Q et R, c'est-à-dire 1, 1 et 0, deviennent 1, 1 et 1; le multiplexeur devient:

P	Q	R
1	1	1
1	1	1
1	1	1
1	1	0

Si l'on applique la procédure susmentionnée aux cinq premiers bits du message à l'aide des cinq premiers éléments des tableaux P, Q et R, on obtient les résultats suivants:

	Tableaux initiaux			Tableaux après chiffrement des cinq bits du message		
Elément	P	Q	R	P	Q	R
1	1	1	1	0	1	1
2	1	1	0	1	1	1
3	1	1	0	1	1	0
4	0	0	1	1	1	1
5	1	0	1	0	1	1
6	0	0	0	0	0	0
7	1	1	0	1	1	0
:	:	:	:	:	:	:
:	:	:	:	:	:	:

Les valeurs du multiplexeur servant à modifier les tableaux après le chiffrement de chacun des cinq premiers bits du message sont:

Après le bit numéro:			1			2			3			4			5		
(valeurs initiales)																	
P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R	P	Q	R
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	0	0

La procédure susmentionnée est appliquée pour chacun des bits du message. Après avoir utilisé l'élément numéro 1009, le tableau R revient à l'élément numéro 1 (début du tableau) et poursuit la procédure avec les nouveaux éléments du tableau. De même, les tableaux Q et P, après avoir utilisé les bits numéros 1015 et 1017 respectivement, recommencent la procédure avec leurs nouveaux premiers éléments. La procédure se poursuit jusqu'à ce que le message entier ait été chiffré.

Le message chiffré est ensuite envoyé à Y. A l'aide de la même clé SSx et de l'algorithme HFX40, Y crée les mêmes tableaux et le même multiplexeur de modification, puis soustrait du message chiffré la somme (modulo 2) des éléments des tableaux pour récupérer le message de télécopie compressé original et modifie les tableaux à l'aide du multiplexeur.

Annexe E

Procédures pour l'utilisation du système de hachage HFX40-I visant à assurer l'intégrité des messages et la sécurité de transmission des documents par télécopie

E.1 Domaine d'application

La présente annexe décrit l'algorithme de hachage HFX40-I en indiquant son utilisation, les calculs à effectuer et les informations à échanger entre les télécopieurs afin d'assurer l'intégrité d'un message de télécopie, cette fonction pouvant être sélectionnée ou préprogrammée en remplacement du chiffrement du message.

Pour assurer l'intégrité des messages, on a recours à une fonction de hachage pour transformer des messages arbitrairement longs en valeurs de longueur fixe. La fonction de hachage produite par l'algorithme HFX40-I transforme le message de télécopie compressé en une suite de chiffres décimaux.

On considère qu'une fonction de hachage est efficace au niveau cryptographique s'il est difficile de découvrir un message qui corresponde à une valeur de hachage donnée ou deux messages qui correspondent à la même valeur de hachage. Pour éviter qu'un tiers puisse y parvenir, l'algorithme HFX40-I a recours à des primitives déterminées à partir d'une clé secrète. La valeur de hachage du message ainsi obtenue est également chiffrée deux fois pour éviter qu'un tiers puisse inverser la fonction de hachage afin de découvrir les primitives initiales.

L'algorithme de hachage HFX40-I repose sur l'utilisation de 19 nombres premiers système, qui sont également employés avec le système de gestion de clés HKM décrit à l'Annexe C et le système de chiffrement des messages décrit à l'Annexe E. La présente annexe ne traite toutefois pas du système de gestion de clés ni du système de chiffrement des messages.

Les procédures qui permettent de sécuriser l'échange de clés secrètes sont décrites en détail à l'Annexe C.

Des exemples de calculs sont donnés aux E.3 et E.4, et peuvent servir à vérifier l'implémentation de l'algorithme.

Le système HFX40-I est visé par des droits de propriété intellectuelle; toutefois, le détenteur de ces droits est convenu d'observer le code de pratique du TSB. Des renseignements complémentaires peuvent être obtenus auprès du TSB.

E.2 Utilisation du système de hachage HFX40-I

Pour envoyer un message dont l'intégrité est protégée, l'utilisateur en X établit une communication avec Y et un enregistrement mutuel est effectué (voir l'Annexe C). Après l'authentification mutuelle de X et de Y (voir l'Annexe C), X crée la clé SSx qui est envoyée en mode sécurisé à Y, à l'aide de la procédure de transfert sécurisé d'une clé secrète (procSTKxy) décrite à l'Annexe C.

X se sert également de la clé SSx pour créer les primitives nécessaires à la fonction de hachage HFX40-I. A l'aide des primitives, la fonction HFX40-I applique des calculs de congruences au message de télécopie compressé, octet par octet. Les arguments de modulos utilisés proviennent d'un ensemble de 19 nombres premiers système mémorisés dans le télécopieur. L'ordre dans lequel les 19 nombres premiers sont utilisés est déterminé par SSx. Cet ensemble de nombres premiers est identique à celui qui est utilisé par le système de gestion de clés HKM (Annexe C) et l'algorithme de chiffrement (Annexe D).

Une série de calculs de congruences est appliquée à un octet du message, le résultat obtenu étant utilisé pour les calculs portant sur l'octet suivant du message. Ce processus est appliqué jusqu'à ce que le message entier ait été traité. La valeur de hachage PH ainsi obtenue est chiffrée deux fois au moyen de la clé de session. La première opération de chiffrement

est une fonction unidirectionnelle qui embrouille les 24 chiffres de PH pour créer la valeur SH. La seconde opération de chiffrement se sert d'une variante unidirectionnelle de la fonction de chiffrement HKM pour créer la valeur ESH. X envoie le message de télécopie compressé et la valeur ESH à Y.

Y se sert de la procédure procSTKxy pour déterminer la clé SSx et de l'algorithme HFX40-I pour calculer la valeur PH du message de télécopie compressé reçu. Y effectue les deux mêmes opérations de chiffrement de la valeur PH que celles qui ont été effectuées par X pour créer ESH. Si cette valeur ESH correspond à la valeur ESH reçue de X, l'intégrité du message est confirmée et l'expéditeur est authentifié. Toute divergence entre la valeur ESH calculée et la valeur ESH reçue est le signe d'une perte d'intégrité.

Y émet le message IMy pour confirmer ou réfuter, auprès de X, l'intégrité du message reçu. IMy est un nombre aléatoire de 12 chiffres choisis parmi les chiffres 2 à 9, dans lequel un chiffre sélectionné de manière aléatoire est remplacé par 1 en cas de confirmation d'intégrité ou par 0 en cas de réfutation. Y procède à l'envoi sécurisé du message IMy à X à l'aide de la clé SSx et de la procédure procSTKyx.

Exemples de réponses:

IMy = 257795**1**99982 Intégrité confirmée.
IMy = 3**1**7736845378 Intégrité confirmée.
IMy = 738543680892 Intégrité réfutée.
IMy = 4577452**0**4639 Intégrité réfutée.

E.3 Système de hachage HFX40-I à utiliser avec des télécopieurs

E.3.1 Introduction

Le sous-paragraphe décrit le système de hachage HFX40-I en indiquant les nombres qui doivent être mémorisés, ainsi que les règles à suivre pour les calculs effectués à l'aide de ces nombres afin de créer les valeurs PH, SH et ESH. Il sera plus facile d'expliquer les règles précitées à l'aide d'exemples numériques. Ces exemples numériques utilisent également des valeurs de test qui permettent de vérifier les implémentations du système HFX40-I.

E.3.2 Informations mémorisées

Tous les terminaux sont dotés du même ensemble de 19 nombres premiers de modulation du système (également utilisé aux Annexes C et D).

32603 32507 32183 32003 31847 31607 31583 31547 31259
31139 30803 30539 30467 30347 30323 30203 29879 29759 29663

E.3.3 Reclassement des nombres premiers de modulation du système

À l'aide de la clé SSx, X détermine trois valeurs primitives de phase initiales, P(0) à P(2), et trois valeurs primitives de base initiales, B(0) à B(2), qui serviront avec les trois premiers nombres premiers 32603, 32507 et 32183 à produire les 19 valeurs d'une séquence PRS (modulo 19).

La clé SSx est divisée en six groupes de trois chiffres qui se chevauchent. Le premier groupe est constitué des trois premiers chiffres de la clé, le deuxième groupe des deuxième, troisième et quatrième chiffres et ainsi de suite. Une opération OU exclusif est appliquée au premier et au deuxième groupe de trois chiffres, puis au résultat obtenu et au troisième groupe et ainsi de suite. On ajoute 2 à chacun des groupes obtenus pour éviter des résultats nuls. Les valeurs obtenues servent ensuite de valeurs primitives de phase et de base initiales P(0) à P(2) et B(0) à B(2). Le Tableau E.1 montre un exemple avec SSx = 568702123345.

Tableau E.1/T.36 – Création des valeurs P(0) à P(2) et B(0) à B(2) pour reclasser les nombres premiers de modulation du système

SSx = 568702123345
Les six groupes sont: 568, 870, 021, 123, 334 et 345

Groupe			Résultat	Addition de 2	Valeur de phase/de base
568			568	570	P(0)
870	XOR	568	350	352	P(1)
021	XOR	350	331	333	P(2)
123	XOR	331	304	306	B(0)
334	XOR	304	126	128	B(1)
345	XOR	126	295	297	B(2)

Les valeurs primitives de phase et de base servent ensuite avec les trois premiers nombres premiers système 32603, 32507 et 32183, dans l'algorithme HKM, à produire les 19 premières valeurs d'une séquence PRS (modulo 19). Un exemple de l'utilisation de l'algorithme HKM avec ces valeurs de test figure au E.4.

La séquence PRS (modulo 19) produite est:

10, 14, 0, 12, 2, 14, 9, 6, 10, 1, 2, 9, 17, 6, 14, 5, 3, 15, 5

Les 19 nombres premiers sont transposés à l'aide de la séquence PRS, ce qui permet de déterminer l'ordre dans lequel ils doivent être utilisés dans l'algorithme de hachage HFX40-I. La première valeur de la PRS détermine quel nombre premier "numéroté" est permuté avec le nombre premier de la première position, la deuxième valeur détermine quel nombre premier "numéroté" est permuté avec le nombre premier de la deuxième position, et ainsi de suite. Le Tableau E.2 indique les 19 étapes de transposition effectuées en fonction de la séquence PRS.

Tableau E.2/T.36 – Transposition à l'aide de la séquence PRS (modulo 19) des nombres premiers n° 0 à 18 utilisés dans les calculs de congruences

Etape	PRS	Nombres premiers n° 0 à 18																		
		0	1	2	3	4	5	6	7	8	9	**10**	11	12	13	14	15	16	17	18
1	10	**10**	1	2	3	4	5	6	7	8	9	**0**	11	12	13	14	15	16	17	18
2	14	10	**14**	2	3	4	5	6	7	8	9	0	11	12	13	**1**	15	16	17	18
3	0	**2**	14	**10**	3	4	5	6	7	8	9	0	11	12	13	1	15	16	17	18
4	12	2	14	10	**12**	4	5	6	7	8	9	0	11	**3**	13	1	15	16	17	18
5	2	2	14	**4**	12	**10**	5	6	7	8	9	0	11	3	13	1	15	16	17	18
6	14	2	14	4	12	10	**1**	6	7	8	9	0	11	3	13	**5**	15	16	17	18
7	9	2	14	4	12	10	1	**9**	7	8	**6**	0	11	3	13	5	15	16	17	18
8	6	2	14	4	12	10	1	**7**	**9**	8	6	0	11	3	13	5	15	16	17	18
9	10	2	14	4	12	10	1	7	9	**0**	6	**8**	11	3	13	5	15	16	17	18
10	1	2	**6**	4	12	10	1	7	9	0	**14**	8	11	3	13	5	15	16	17	18
11	2	2	6	**8**	12	10	1	7	9	0	14	**4**	11	3	13	5	15	16	17	18
12	9	2	6	8	12	10	1	7	9	0	**11**	4	**14**	3	13	5	15	16	17	18
13	17	2	6	8	12	10	1	7	9	0	11	4	14	**17**	13	5	15	16	**3**	18
14	6	2	6	8	12	10	1	**13**	9	0	11	4	14	17	**7**	5	15	16	3	18
15	14	2	6	8	12	10	1	13	9	0	11	4	14	17	7	**5**	15	16	3	18
16	5	2	6	8	12	10	**15**	13	9	0	11	4	14	17	7	5	**1**	16	3	18
17	3	2	6	8	**16**	10	15	13	9	0	11	4	14	17	7	5	1	**12**	3	18
18	15	2	6	8	16	10	15	13	9	0	11	4	14	17	7	5	**3**	12	**1**	18
19	5	2	6	8	16	10	**18**	13	9	0	11	4	14	17	7	5	3	12	1	**15**

Dans le processus susmentionné, la première étape a consisté à permuter le premier nombre premier avec le nombre premier n° 10, la deuxième étape à permuter le deuxième nombre premier avec le nombre premier n° 14 et la dernière étape à permuter le dix-neuvième nombre premier avec le nombre premier n° 5.

L'ordre final des nombres premiers numérotés utilisés dans les calculs de congruences est:

2, 6, 8, 16, 10, 18, 13, 9, 0, 11, 4, 14, 17, 7, 5, 3, 12, 1, 15

ce qui donne:

32183, 31583, 31259, 29879, 30803, 29663, 30347, 31139, 32603

30539, 31847, 30323, 29759, 31547, 31607, 32003, 30467, 32507, 30203

On utilise les trois premiers nombres avec l'algorithme HFX40-I pour produire la valeur PH, ainsi que dans les deux opérations de chiffrement les valeurs SH et ESH.

E.3.4 Calcul des primitives à utiliser avec HFX40-I

Huit primitives de phase P(0) à P(7) sont créées à partir de la clé SSx de la manière suivante.

La clé SSx, 568702123345, est subdivisée en huit groupes de quatre chiffres qui se chevauchent. Le premier groupe est constitué des quatre premiers chiffres de la clé, le deuxième groupe des troisième, quatrième, cinquième et sixième chiffres et ainsi de suite. Une opération OU exclusif est appliquée au premier et au deuxième groupe de quatre chiffres, puis au résultat obtenu et au troisième groupe, et ainsi de suite. On ajoute 2 à chacune des valeurs obtenues pour éviter des résultats nuls et les valeurs obtenues servent de valeurs initiales P(0) à P(7).

Le Tableau E.3 montre un exemple avec SSx = 568702123345.

Tableau E.3/T.36 – Calcul des primitives à utiliser avec HFX40-I

SSx = 568702123345 Les huit groupes sont: 5687, 8702, 7021, 0212, 2123, 1233, 2334 et 3345					
Groupe			**Résultat**	**Addition de 2**	**Valeur de phase/de base**
5687				5689	P(0)
8702	XOR	5687	14281	14283	P(1)
7021	XOR	14281	11428	11430	P(2)
212	XOR	11428	11376	11378	P(3)
2123	XOR	11376	9275	9277	P(4)
1233	XOR	9275	8426	8428	P(5)
2334	XOR	8426	10740	10742	P(6)
3345	XOR	10740	9445	9447	P(7)

E.3.5 Calcul de la valeur PH

On utilise à tour de rôle les valeurs de phase P(0) à P(7) en commençant par P(1). Lorsque la valeur initiale de P(1) a été utilisée dans les calculs portant sur le premier octet du message compressé, elle est remplacée par une nouvelle valeur obtenue à l'aide des calculs décrits dans les paragraphes suivants. C'est cette valeur qui sera appliquée lors de la prochaine utilisation de P(1) dans les calculs. La même procédure est appliquée avec P(2), P(3), etc. La procédure entière, qui continue jusqu'à la fin du message, est expliquée en détail ci-après.

Soit P(n) la valeur de phase actuellement utilisée. On modifie P(n) pour créer P′ (n) en ajoutant à P(n) la valeur décimale en code ASCII de l'octet du message actuellement utilisé, b, et la valeur Q (modulo M), q, obtenue à partir de l'octet précédent. Pour le premier octet, q = 0.

$$P'(n) = P(n) + b + q$$

On détermine Q en multipliant P′(n) par (b + 1).

$$Q = P'(n) * (b + 1)$$

On applique comme argument de modulo à Q l'un des 19 nombres premiers utilisés à tour de rôle à partir du nombre premier (1) pour créer les nouvelles valeurs P(n) et q. Par exemple, pour le premier octet du message, Q (modulo M) crée la valeur q pour le deuxième octet et P(n) crée cette valeur pour le neuvième octet. L'ordre des nombres premiers est celui qui est déterminé au E.3.3. Le nombre premier (1) est utilisé pour le premier octet du message, le nombre premier (2) pour le deuxième octet, et ainsi de suite.

Le Tableau E.4 indique les calculs qui permettent de produire la valeur de hachage pour un message de télécopie compressé de 29 octets.

La valeur PH est créée par concaténation des trois chiffres de poids faible des huit valeurs de phase finales, dans l'ordre P(0), P(1), P(2), etc. à P(7), ce qui donne un nombre de 24 chiffres. Dans l'exemple du Tableau E.4:

PH = 171 666 427 631 042 698 579 505

E.3.6 Premier chiffrement (embrouillage) de la valeur PH pour créer la valeur SH

La clé SSx sert à déterminer trois primitives de phase P(0) à P(2) et trois primitives de base B(0) à B(2). Elle est divisée en six groupes de trois chiffres qui se chevauchent. Les premiers chiffres 1-3 forment P(0), les chiffres 3-5 forment P(1), les chiffres 4-6 forment P(2), les chiffres 7-9 forment B(0), les chiffres 9-11 forment B(1) et les chiffres 10-12 forment B(2).

Une opération OU exclusif est appliquée au premier et au deuxième groupe de trois chiffres, puis au résultat obtenu et au troisième groupe, ainsi de suite. On ajoute 2 à chacun des groupes de trois chiffres obtenus pour éviter des résultats nuls. On combine les nouvelles valeurs avec les six premiers groupes de trois chiffres provenant de la valeur PH pour obtenir les nouvelles valeurs P(0) à P(2) et B(0) à B(2).

Le Tableau E.5 montre un exemple de calcul effectué à l'aide des mêmes valeurs que précédemment.

Tableau E.4/T.36 – Calcul de la valeur de hachage pour un message de télécopie compressé de 29 octets

b	y	P(n)	q	P′(n)	Q	M	Q(mod M)	Résultat du hachage
103	1	14283	0	14386	1496144	31583	11743	
121	2	11430	11743	23294	2841868	31259	28558	
2	3	11378	28558	39938	119814	29879	298	
0	4	9277	298	9575	9575	30803	9575	
34	5	8428	9575	18037	631295	29663	8372	
79	6	10742	8372	19193	1535440	30347	18090	
92	7	9447	18090	27629	2569497	31139	16099	
92	0	5689	16099	21880	2034840	32603	13454	
33	1	11743	13454	25230	857820	30539	2728	
33	2	28558	2728	31319	1064846	31847	13895	
33	3	298	13895	14226	483684	30323	28839	
10	4	9575	28839	38424	422664	29759	6038	
4	5	8372	6038	14414	72070	31547	8976	
238	6	18090	8976	27304	6525656	31607	14614	
161	7	16099	14614	30874	5001588	32003	9120	
141	0	13454	9120	22715	3225530	30467	26495	
24	1	2728	26495	29247	731175	32507	16021	
2	2	13895	16021	29918	89754	30203	29348	
3	3	28839	29348	58190	232760	32183	7479	
62	4	6038	7479	13579	855477	31583	2736	
149	5	8976	2736	11861	1779150	31259	28646	
66	6	14614	28646	43326	2902842	29879	4579	**579 [valeur de P(6)]**
11	7	9120	4579	13710	164520	30803	10505	**505 [valeur de P(7)]**
93	0	26495	10505	37093	3486742	29663	16171	**171 [valeur de P(0)]**
39	1	16021	16171	32231	1289240	30347	14666	**666 [valeur de P(1)]**
133	2	29348	14666	44147	5915698	31139	30427	**427 [valeur de P(2)]**
19	3	7479	30427	37925	758500	32603	8631	**631 [valeur de P(3)]**
124	4	2736	8631	11491	1436375	30539	1042	**042 [valeur de P(4)]**
92	5	28646	1042	29780	2769540	31847	30698	**698 [valeur de P(5)]**

Tableau E.5/T.36 – Calcul des valeurs P(0) à P(2) et B(0) à B(2) pour la première opération de chiffrement de PH

Clé de session = 568702123345

Valeurs P et B initiales			Addition de 2			Groupes PH		Nouvelles valeurs P et B	
568			568	570	+	171	=	741	P(0)
870	XOR 568	=	350	352	+	666	=	1018	P(1)
021	XOR 350	=	331	333	+	427	=	760	P(2)
123	XOR 331	=	304	306	+	631	=	937	B(0)
334	XOR 304	=	126	128	+	042	=	170	B(1)
345	XOR 126	=	295	297	+	698	=	995	B(2)

Les valeurs primitives de phase et de base P(0) à P(2) et B(0) à B(2) susmentionnées, ainsi que les trois premiers nombres premiers de modulation du système 32183, 31583 et 31259 reclassés au E.3.3, sont utilisés dans l'algorithme HKM de la même manière que celle qui est décrite au E.4 pour créer la séquence PRS (modulo 24) suivante, qui comprend 24 valeurs:

4, 5, 4, 16, 2, 6, 16, 12, 24, 21, 6, 5, 11, 4, 8, 21, 22, 5, 24, 9, 3, 16, 19, 8

Cette séquence sert à transposer l'ensemble des 24 chiffres de la valeur PH et à créer la valeur SH. La première valeur de la séquence PRS détermine la position du chiffre de PH qui est permuté avec le premier chiffre de PH, la deuxième valeur détermine la position du chiffre qui est permuté avec le deuxième chiffre et ainsi de suite. En ce qui concerne la séquence PRS susmentionnée, la première étape consiste à permuter la position 1 avec la position 4, la deuxième étape à permuter les positions 2 et 5 et ainsi de suite jusqu'à l'étape finale où la position 24 est permutée avec la position 8. Le Tableau E.6 montre les 24 étapes de transposition effectuées en fonction de la séquence PRS.

Tableau E.6/T.36 – Transposition de la valeur PH pour créer la valeur SH

Etape/position	PRS	PH	171	666	427	631	042	698	579	505
1	4		**6**71	**1**66	427	631	042	698	579	505
2	5		6**6**1	1**7**6	427	631	042	698	579	505
3	4		66**1**	**1**76	427	631	042	698	579	505
4	16		661	**6**76	427	631	042	**1**98	579	505
5	2		6**7**1	6**6**6	427	631	042	198	579	505
6	6		671	66**6**	427	631	042	198	579	505
7	16		671	666	**1**27	631	042	**4**98	579	505
8	12		671	666	1**1**7	63**2**	042	498	579	505
9	24		671	666	11**5**	632	042	498	579	50**7**
10	21		671	666	115	**9**32	042	498	57**6**	507
11	6		671	66**3**	115	9**6**2	042	498	576	507
12	5		671	6**2**3	115	96**6**	042	498	576	507
13	11		671	623	115	9**0**6	**6**42	498	576	507
14	4		671	**4**23	115	906	6**6**2	498	576	507
15	8		671	423	1**2**5	906	66**1**	498	576	507
16	21		671	423	125	906	661	**6**98	57**4**	507
17	22		671	423	125	906	661	6**5**8	574	**9**07
18	5		671	4**8**3	125	906	661	65**2**	574	907
19	24		671	483	125	906	661	652	**7**74	90**5**
20	9		671	483	12**7**	906	661	652	7**5**4	905
21	3		67**4**	483	127	906	661	652	751	905
22	16		674	483	127	906	661	**9**52	751	**6**05
23	19		674	483	127	906	661	952	**0**51	67**5**
24	8		674	483	1**5**7	906	661	952	051	67**2**
		SH = 674 483 157 906 661 952 051 672								

E.3.7 Chiffrement de la valeur SH pour créer la valeur ESH

Trois primitives de phase P(0) à P(2) et trois primitives de base B(0) à B(2) sont déterminées à partir de la clé SSx, de la même manière que celle qui est décrite au E.3.6, sauf que les valeurs résultant de l'opération OU exclusif et de l'addition de 2 sont combinées avec les six premiers groupes de trois chiffres de SH, ce qui donne les nouvelles valeurs qui servent de valeurs initiales pour P(0) à P(2) et B(0) à B(2).

Le Tableau E.7 montre les résultats des calculs effectués à l'aide de la valeur SH obtenue au E.3.6.

Tableau E.7/T.36 – Calcul des primitives servant à chiffrer la valeur EH et à créer la valeur ESH

SH = 674 483 157 906 661 952 051 672											
Valeurs initiales de P et B				**Addition de 2**			**Groupes SH**		**Nouvelles valeurs de P et B**	
568				568	570	+	674	=	1244	P(0)
870	XOR	568	=	350	352	+	483	=	835	P(1)
021	XOR	350	=	331	333	+	157	=	490	P(2)
123	XOR	331	=	304	306	+	906	=	1212	B(0)
334	XOR	304	=	126	128	+	661	=	789	B(1)
345	XOR	126	=	295	297	+	952	=	1249	B(2)

Les valeurs primitives de phase et de base P(0) à P(2) et B(0) à B(2) susmentionnées, ainsi que les trois premiers nombres premiers 32183, 31583 et 31259 (résultant du reclassement effectué au E.3.3), sont utilisés dans l'algorithme HKM de la même manière que celle qui est décrite dans l'exemple du E.4 pour créer les 24 valeurs d'une séquence PRS (modulo 10), qui sont ajoutées (modulo 10) à la valeur de SH de manière à créer la valeur ESH.

SH: 674 483 157 906 661 952 051 672

PRS (modulo 10): 402 025 183 343 270 975 304 836

ESH: 076 408 230 249 831 827 355 408

E.4 Utilisation de l'algorithme HKM pour créer une séquence pseudo-aléatoire

E.4.1 Introduction

Le présent sous-paragraphe donne un exemple qui permet, à l'aide de valeurs de test, de vérifier l'application de l'algorithme HKM en vue de produire une séquence pseudo-aléatoire PRS à l'aide de calculs de congruences.

A cet effet, on utilise des valeurs primitives de phase et de base qui sont déterminées de différentes manières à partir d'un nombre secret (notamment une clé de session) ou à partir de la combinaison d'un nombre secret et d'une autre donnée d'identification, ainsi que des arguments de modulos choisis dans l'ensemble universel de nombres premiers utilisés pour les calculs de congruences, afin de créer une séquence PRS à laquelle un autre calcul de congruences peut être appliqué pour former une séquence PRS particulière.

E.4.1.1 Calculs effectués à l'aide de l'algorithme HKM pour créer une séquence PRS

Les valeurs de test utilisées dans cet exemple sont celles qui sont utilisées au E.3.3 pour former la séquence PRS de 19 valeurs (modulo 19) qui permet de reclasser les nombres premiers utilisés dans les calculs de congruences.

Les trois primitives de phase P(0) à P(2) et les trois primitives de base B(0) à B(2) sont:

P(0) = 570 B(0) = 306
P(1) = 352 B(1) = 128
P(2) = 333 B(2) = 297

Les trois nombres premiers utilisés dans les calculs de congruences sont: 32603, 32507 et 32183.

Prenons le premier "jeu" de valeurs constitué de la valeur de phase P(0), de la valeur de base B(0) et du nombre premier 32603:

On multiplie P(0) par B(0):

570 * 306 = 174420

174420 (modulo le premier nombre premier) = 174420 (modulo 32603) = 11405.

On utilise ensuite 11405 comme nouvelle valeur de phase et on multiplie ce nombre par la valeur de base B(0):

11405 * 306 = 3489930

3489930 (modulo le premier nombre premier) = 3489930 (modulo 32603) = 1409.

Ce processus est appliqué 19 fois en tout (ce qui correspond au nombre de nombres premiers à reclasser). On applique également le processus entier aux deux "jeux" restants de valeurs de phase et de base et de nombres premiers pour créer des séquences analogues.

On additionne les résultats du premier calcul appliqué à chacun des trois "jeux" de valeurs de phase et de base et de nombres premiers, ce qui donne la première valeur de la colonne Total. On additionne les résultats de chacun des autres calculs pour obtenir les autres valeurs de la colonne Total. On applique l'argument de modulo 19 à la séquence obtenue dans cet exemple pour former une séquence PRS (modulo 19). Dans ce type d'opération, on peut appliquer d'autres nombres appropriés à la séquence pour créer des séquences PRS particulières.

L'ensemble des calculs figure au Tableau E.8.

Tableau E.8/T.36 – Calculs effectués à l'aide de HKM pour créer une séquence PRS de 19 valeurs (modulo 19)

Nombre premier B(n) P(n)	32603 306 570	32507 128 352	32183 297 333	Total	**PRS (modulo 19)**
	11405	12549	2352	26306	**10**
	1409	13429	22701	37539	**14**
	7315	28548	15950	51813	**0**
	21386	13360	6249	40995	**12**
	23516	19716	21522	64754	**2**
	23236	20609	19800	63645	**14**
	2762	4885	23294	30941	**9**
	30097	7647	31156	68900	**6**
	15636	3606	16811	36053	**10**
	24578	6470	4502	35550	**1**
	22178	15485	17591	55254	**2**
	5044	31660	10881	47585	**9**
	11123	21612	13357	46092	**17**
	12926	3241	8520	24687	**6**
	10393	24764	20166	55323	**14**
	17767	16613	3264	37644	**5**
	24604	13509	3918	42031	**3**
	30134	6281	5058	41473	**15**
	26958	23800	21808	72566	**5**

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
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	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
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
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 pour données et communication entre systèmes ouverts
Série Z	Langages de programmation